

ЗМІСТ

ВСТУП	4
СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ	6
ТЕМИ ЛЕКЦІЙНИХ ЗАНЯТЬ. СИСТЕМА ОЦІНЮВАННЯ ЗНАТЬ	7
ЛЕКЦІЙНЕ ЗАНЯТТЯ №1 Відкриті операційні системи: історія, концепції, дистрибутиви. Міжнародний досвід впровадження	8
ЛЕКЦІЙНЕ ЗАНЯТТЯ №2 Ядро операційної системи. Процеси. Сценарії запуску	17
ЛЕКЦІЙНЕ ЗАНЯТТЯ №3 Концепції файлової системи. Ієрархія логічних файлових систем GNU/Linux	29
ЛЕКЦІЙНЕ ЗАНЯТТЯ №4 Завантаження операційної системи GNU/Linux. Робота з командним інтерпретатором	38
ЛЕКЦІЙНЕ ЗАНЯТТЯ №5 Поняття користувача і групи у багатокористувацькій операційній системі	47
ЛЕКЦІЙНЕ ЗАНЯТТЯ №6 Графічний інтерфейс операційної системи. Графічні оточення KDE і GNOME	52
ЛЕКЦІЙНЕ ЗАНЯТТЯ №7 Прикладне програмне забезпечення для роботи у UNIX-подібних операційних системах	58
СПИСОК ПИТАНЬ, ВИНЕСЕНИХ НА САМОСТІЙНУ РОБОТУ	72
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	74

ВСТУП

У сьогоднішніх карантинних реаліях, актуальним для суспільства стає вміння користуватися інформаційними технологіями. Важко назвати іншу сферу людської діяльності, яка б розвивалася так стрімко і породжувала б таке розмаїття проблем, як інформатизація та комп'ютеризація суспільства. Інформаційні технології інтенсивно розвиваються, росте число функціональних проявів комп'ютерних систем як у промислових масштабах, так і на побутовому рівні. Незважаючи на всю технічну розмаїтість і складність апаратних і архітектурно-модульних складових сучасної електронно-обчислювальної техніки, вона доступна і для невідготованого користувача. Це можливо завдяки інтуїтивно зрозумілому набору високорівневих абстракцій, якими оперує операційна система в процесі своєї роботи. Користувачу відомо, що для роботи з комп'ютером потрібна операційна система. Операційна система є набором узгоджено працюючих програм, які забезпечують керування ресурсами комп'ютера, запуск прикладних програм, їхню взаємодію з периферійними пристроями, а також забезпечують діалог з користувачем.

Відкриті операційні системи – це окремий, особливий, сегмент операційних систем. По-перше, переважна більшість дистрибутивів таких систем є безкоштовними. По-друге, даний клас системного забезпечення розповсюджується з відкритим вихідним кодом, подібна практика досить поширена, саме завдяки їй з часом з'являється все більше різних варіацій GNU/Linux-дистрибутивів. По-третє, світове IT-співтовариство активно підтримує їхній розвиток і технологічне удосконалення.

Використання в професійній діяльності відкритих операційних систем – це стяг демократичного поступу, свободи, відкритого суспільства і права ділитись своїми доробками з іншими. Розвиток і підтримка відкритих рішень є одним із найважливіших показників демократичного розвитку країни. Очевидно, що використовуючи відкриті операційні системи, зокрема дистрибутиви GNU/Linux в державних навчальних закладах України, можна значно зекономити державні кошти, що витрачаються на ліцензування системного і прикладного програмного забезпечення. Протягом довгого часу, питання щодо використання відкритих операційних систем в процесі вивчення інформаційних технологій просто не була предметом обговорення або серйозного аналізу. Очевидно, і ні для кого не секрет, що частка комерційного програмного забезпечення в українських навчальних закладах – значна, а суспільна думка, в переважачій своїй більшості, поки схильна вважати «легальність» системного і програмного забезпечення питанням другорядним, не важливим. Серед переваг, щодо використання відкритих операційних систем можна виділити такі: економія бюджетних коштів, використання ліцензійних операційних систем із достатньо широким набором прикладного і інструментального програмного забезпечення, ефективність роботи на застарілих комп'ютерах, можливість запуску операційної системи без попередньої інсталяції.

Лекційний курс навчальної дисципліни «Відкриті операційні системи», включає матеріали для вивчення основ відкритих операційних систем, зокрема такі теми:

- Відкриті операційні системи: історія, концепції, дистрибутиви. Міжнародний досвід впровадження.
- Ядро операційної системи. Процеси. Сценарії запуску.
- Концепції файлової системи. Ієрархія логічних файлових систем GNU/Linux.
- Завантаження операційної системи GNU/Linux. Робота з командним інтерпретатором.
- Поняття користувача і групи у багатокористувацькій операційній системі.
- Графічний інтерфейс операційної системи. Графічні оточення KDE і GNOME.
- Прикладне програмне забезпечення для роботи у UNIX-подібних операційних системах.

Даний курс лекцій, сприятиме ознайомленню студентства з відкритими операційними системами, їхніми архітектурними аспектами функціонування, популярними «вільними» щодо правових та ліцензійних аспектів програмами та системами, корисними при викладанні шкільного курсу інформатики. Дозволить оволодіти знаннями і вміннями, щодо:

- програмних і архітектурних аспектів роботи відкритих операційних систем;
- роботи ядра операційної системи та її модулів;
- організації планування процесів;
- роботи з різними видами файлових систем;
- етапів завантаження операційної системи;
- роботи з командою оболонкою bash;
- роботи з графічним оточенням робочого столу, на прикладі GNOME і KDE;
- роботи з файловим менеджером Midnight Commander;
- роботи з екранно-орієнтованим текстовим редактором vi.

СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин											
	денна форма						заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с. р.		л	п	лаб	інд	с. р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Змістовий модуль 1. Архітектура відкритих операційних систем.												
Тема 1. Відкриті операційні системи: історія, концепції, дистрибутиви. Міжнародний досвід впровадження.	13	2		2		9	13	1				12
Тема 2. Ядро операційної системи. Процеси. Сценарії запуску.	13	2				11	13	1				12
Тема 3. Концепції файлової системи. Ієрархія логічних файлових систем GNU/Linux.	13	2		2		9	13			1		12
Разом за змістовим модулем 1	39	6		4		29	39	2		1		36
Змістовий модуль 2. Основи адміністрування відкритих операційних систем.												
Тема 4. Завантаження операційної системи GNU/Linux. Робота з командним інтерпретатором.	13	2		4		7	12	1		1		10
Тема 5. Поняття користувача і групи у багатокористувацькій операційній системі.	13	2				11	14					14
Разом за змістовим модулем 2	26	4		4		18	26	1		1		24
Змістовий модуль 3. Аспекти прикладного застосування відкритих операційних систем.												
Тема 6. Графічний інтерфейс операційної системи. Графічне оточення KDE і GNOME.	13	2		2		9	12	1		2		9
Тема 7. Прикладне програмне забезпечення для роботи у UNIX-подібних операційних системах.	12	2		6		4	13			2		11
Разом за змістовим модулем 3	25	4		8		13	25	1		4		20
Усього годин	90	14		16		60	90	4		6		80

ТЕМИ ЛЕКЦІЙНИХ ЗАНЯТЬ

№ з/п	Назва теми	Години
1	Відкриті операційні системи: історія, концепції, дистрибутиви. Міжнародний досвід впровадження.	2
2	Ядро операційної системи. Процеси. Сценарії запуску.	2
3	Концепції файлової системи. Ієрархія логічних файлових систем GNU/Linux.	2
4	Завантаження операційної системи GNU/Linux. Робота з командним інтерпретатором.	2
5	Поняття користувача і групи у багатокористувацькій операційній системі.	2
6	Графічний інтерфейс операційної системи. Графічні оточення KDE і GNOME.	2
7	Прикладне програмне забезпечення для роботи у UNIX-подібних операційних системах.	2
РАЗОМ		14

СИСТЕМА ОЦІНЮВАННЯ ЗНАНЬ

Поточне тестування та самостійна робота							Сума
Змістовий модуль 1			Змістовий модуль 2		Змістовий модуль 3		
T1	T2	T3	T4	T5	T6	T7	10
12	11	12	15	11	12	17	
							100

ЛЕКЦІЙНЕ ЗАНЯТТЯ №1

Тема: Відкриті операційні системи: історія, концепції, дистрибутиви. Міжнародний досвід впровадження.

Мета: Формування знань і вмінь, щодо використання відкритих операційних систем в професійній діяльності.

ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1. Основні визначення

1.2. Класифікація, функції та характеристики операційної системи

1.3. Генеалогічне дерево UNIX-систем

1.4. Дистрибутив операційної системи GNU/Linux

1.5. Основні дистрибутиви GNU/Linux

1.6. Міжнародний досвід впровадження операційної системи GNU/Linux

1.1. Основні визначення

Ліцензія – це документ, що дає право на виконання деяких дій.

Авторське право – це підрозділ цивільного права, що регулює правовідносини, пов'язані зі створенням і використанням, виданням, виконанням, показом здобутків науки, літератури або мистецтва. Програмне забезпечення також охороняється авторським правом.

Вільне програмне забезпечення – це програмне забезпечення у відношенні до якого користувач має право: запускати, вивчати, поширювати й поліпшувати програму. Програмний продукт і його вихідний код за замовчуванням охороняється авторським правом, що дає авторові повну владу над поширенням і зміною програми, навіть у випадку, коли вихідний код загальнодоступний для огляду.

Відкрите програмне забезпечення (програмне забезпечення з відкритим вихідним кодом, англ. open source software) – це спосіб розробки програмного забезпечення, при якому вихідний код створюваних програм відкритий, тобто загальнодоступний для перегляду й зміни. Це дозволяє всім бажаючим використати вже створений код для своїх потреб.

UNIX – перша мобільна операційна система для міні-комп'ютерів, розроблена у 1969 році Браяном Керніганом і Деннісом Рітчі на мові програмування C.

GNU General Public License (Відкрита ліцензійна угода GNU) – це ліцензія на вільне програмне забезпечення, створена в рамках проекту GNU у 1988 році. Ціль GNU GPL – надати користувачеві права копіювати, модифікувати й поширювати (у тому числі на комерційній основі) програми, а також гарантувати, що й користувачі всіх похідних програм одержать перераховані вище права. Версії – GPL v1, GPL v2, GPL v3.

GNU/Linux – загальна назва UNIX-подібних операційних систем на основі однойменного ядра (Linux). Це один із найкращих прикладів розробки проекту з відкритим кодом (open source) за використання вільного програмного забезпечення.

Фонд вільного програмного забезпечення (*Free Software Foundation*) – некомерційна організація, заснована в жовтні 1985 року Річардом Столменом для підтримки руху вільного програмного забезпечення й особливо проекту GNU.

Безкоштовне програмне забезпечення або freeware (free – безкоштовний і software – програмне забезпечення) – це програмне забезпечення, ліцензійна угода якого не вимагає яких-небудь виплат власнику.

Ліцензія MIT (MIT License) – група ліцензій, розроблених Масачусетським технологічним інститутом для поширення вільного програмного забезпечення.

Ліцензія BSD – програмна ліцензія університету Берклі, вперше застосована для поширення UNIX-подібних операційних систем BSD.

GNU Lesser General Public License – це ліцензія на вільне програмне забезпечення, що схвалена фондом вільного програмного забезпечення й розроблена як компроміс між GNU General Public License і дозволеними ліцензіями, такими як BSD License і MIT License.

Комерційне програмне забезпечення (commercial software) – це програмне забезпечення, створене комерційною організацією з метою одержання прибутку від його використання іншими, наприклад шляхом продажу екземплярів.

Операційна система – це комплекс програмних засобів і даних, які забезпечують управління роботою апаратної та програмної складових обчислювальної системи, координують їх взаємодію, забезпечують виконання функцій посередника між користувачем і комп'ютером.

1.2. Класифікація, функції та характеристики операційної системи

У залежності від функціонального призначення, операційну систему можна класифікувати за різними критеріями:

- 1) Однозадачні та багатозадачні операційні системи. Однозадачні операційні системи можуть виконувати наступну програму тільки після завершення поточної (активної) задачі. Багатозадачні операційні системи, на відміну від однозадачних, надають можливість одночасного виконання кількох програм, розподіляючи між ними ресурси обчислювальної системи: процесорний час, пам'ять, периферійні пристрої.
- 2) Однокористувацькі та багатокористувацькі.
- 3) З підтримкою роботи в мережі (мережеві операційні системи) та без підтримки роботи в мережі.
- 4) Операційна система для виконання функцій сервера мережі або робочої станції (клієнта мережі).

Операційна система є основним системним програмним комплексом, який виконує наступні функції:

- тестування працездатності обчислювальної системи і її налаштування при початковому завантаженні;
- забезпечення синхронної і ефективної взаємодії всіх апаратних і програмних компонентів обчислювальної системи в процесі її функціонування;
- забезпечення ефективної взаємодії користувача з обчислювальною

системою.

Операційні системи, класифікують за такими характеристиками:

- Число процесів, які одночасно можуть виконуватися під керуванням даної системи – однозадачні та багатозадачні. Поняття багатозадачності означає підтримку паралельного виконання кількох задач, які існують в межах однієї обчислювальної системи в один момент часу.
- Кількість підтримуваних процесорів (однопроцесорні, багатопроцесорні). Багатопроцесорні операційні системи підтримують режим розподілених ресурсів декількох процесорів для розв'язання однієї задачі.
- Розрядність коду: 8-ми, 16-ти, 32-ох та 64-ох розрядні операційні системи.
- Тип інтерфейсу: командні (з текстовим інтерфейсом) і об'єктно-орієнтовані (з графічним інтерфейсом).
- Тип доступу користувача до комп'ютера: з пакетною обробкою, з розділенням часу і реальним часом. Наприклад, в режимі пакетної обробки з виконуваних програм формується пакет (набір) завдань, які вводяться в комп'ютер і виконуються в порядку черговості з можливим врахуванням пріоритетів.
- Тип використовуваних ресурсів: мережеві та локальні. Локальна операційна система забезпечує роботу тільки одного користувача, в той час як мережева – дозволяє одночасно працювати багатьом незалежним користувачам в мережі, використовуючи спільні ресурси.

Об'єкти ядра операційної системи:

- Процеси.
- Файли.
- Події.
- Потoki.
- Семафори.
- Канали.

1.3. Генеалогічне дерево UNIX-систем

Брайан Керніган, Кен Томпсон і Денніс Рітчі – творці операційної системи UNIX. У 1969 році Кен Томпсон, прагнучи реалізувати ідеї, які були покладені в основу MULTICS, але на більш скромному апаратному забезпеченні (DEC PDP-7), написав першу версію нової операційної системи, а Брайан Керніган придумав для неї назву – UNICS (UNIplicated Information and Computing System) – на противагу MULTICS (MULTIplicated Information and Computing Service). Пізніше ця назва скоротилася до UNIX. Перша система UNIX була розроблена в 1969 році у підрозділі Bell Labs компанії AT&T. З тих часів була створена велика кількість різних UNIX-систем. Юридично лише деякі з них мають повне право називатися «UNIX» інші ж хоча й використовують подібні концепції й технології, поєднуються терміном «UNIX-подібні» (англ. UNIX-like).

У 1978 році розпочинає свою історію операційна система BSD UNIX, створена в університеті Берклі. Її перша версія була заснована на шостій редакції

UNIX. У 1979 році випущена нова версія, названа 3BSD, заснована на сьомій редакції UNIX. На початку 1980-х років компанія AT&T, усвідомила цінність UNIX і відразу розпочала просування її комерційної версії. Ця версія, яка надійшла у продаж у 1982 році, отримала назву – UNIX System III.

Деякі ознаки притаманні UNIX-системі:

- використання простих текстових файлів для налаштування й керування системою;
- широке застосування утиліт, які прописуються у командному рядку;
- взаємодія з користувачем за допомогою віртуального пристрою – терміналу;
- надання фізичних і віртуальних пристроїв;
- використання конвєсерів з декількох програм, кожна з яких виконує одне завдання.

У часи становлення, UNIX використовується в основному на серверах, а також як вбудована система для різного устаткування. На ринку операційних систем для робочих станцій і домашнього використання UNIX поступився іншим операційним системам в першу чергу Microsoft Windows. Хоча існуючі програмні рішення для UNIX-систем дозволяли реалізувати повноцінну робочу станцію як для офісного, так і для домашнього використання.

1.4. Дистрибутив операційної системи GNU/Linux

Дистрибутив операційної системи – це форма поширення системного програмного забезпечення. Наявність дистрибутивів викликана тим, що форма програмного забезпечення, використовувана для його поширення, майже ніколи не збігається з формою програмного забезпечення працюючої системи. Різновиди GNU/Linux називають версіями або дистрибутивними пакетами. До складу дистрибутивного пакету входять: ядро, інсталяційна програма орієнтована на дану версію GNU/Linux, набір утиліт (спеціальних інструментів), програм користувача.

Слово «Linux» позначає тільки назву ядра. Тому, коли мова йде про операційну систему, вірніше було б говорити «операційна система, заснована на ядрі Linux». Ядро операційної системи GNU/Linux розробляється під загальним керівництвом Лінуса Торвальдса й поширюється вільно (на основі ліцензії GPL), як і величезна кількість іншого програмного забезпечення, утиліт і прикладних програм. Одним з наслідків вільного поширення програмного забезпечення для GNU/Linux є те, що велика кількість різних фірм і компаній, а також просто незалежних груп розроблювачів стали випускати так звані дистрибутиви GNU/Linux.

У будь-якій операційній системі, можна виділити чотири її основні частини: ядро, файлову систему, інтерпретатор команд користувача й утиліти. Ядро – це основна, визначальна частина операційної системи, яка керує апаратними засобами й виконанням програм. Файлова система – це структура для зберігання файлів на запам'ятовуючих пристроях. Інтерпретатор команд або оболонка – це програма, яка організує взаємодію користувача з комп'ютером. І нарешті, утиліти – це просто окремі програми, які загалом, кажучи, нічим принципово не відрізняються від

інших програм, що запускаються користувачем, хіба тільки своїм основним призначенням – вони виконують службові функції. Компілятори – це програми, які перекладають (перетворюють) мову, зрозумілу людям у комп'ютерну. Робочі столи – це додатки, які забезпечують графічну взаємодію з операційною системою.

1.5. Основні дистрибутиви GNU/Linux

Red Hat (червоний капелюх) – американська ІТ-компанія (з 2018 року належить корпорації IBM), що базується в місті Ралі, столиці штату Північна Кароліна у США. Компанія випускає рішення на базі вільної операційної системи GNU/Linux: Red Hat Enterprise GNU/Linux (комерційна версія) і Fedora Core (вільна версія), та інші програмні продукти і послуги на базі відкритого вихідного коду (зокрема середовище компіляції і виконання застосунків GNU/Linux під операційну систему Microsoft Windows – Cygwin). Серед послуг, які надає компанія: безперервна технічна підтримка, навчання системних адміністраторів і розробників, прийом іспитів RHCE (Red Hat Certified Engineer), Red Hat Certified Technician (RHCT), RHCSS (Red Hat Certified Security Specialist) і RHCA (Red Hat Certified Architect). Компанія почала працювати у 1993 році і нині налічує понад 5000 співробітників і 27 підрозділів у всьому світі, будучи одним з найбільших постачальників рішень на базі GNU/Linux та компанією №1 за внеском у розвиток ядра Linux.

Debian – це операційна система, основний дистрибутив якої складається виключно з вільного програмного забезпечення. Надається можливість встановлення також пакетів програм, які не відповідають означенню вільного програмного забезпечення, і їх винесено в окремий архів non-free. Найпоширеніша версія цієї операційної системи – це Debian GNU/Linux. Це багатозільова операційна система, може використовуватись на настільних комп'ютерах, лептопах, серверах та вбудованих системах. Поточний випуск Debian містить понад 50000 пакетів із програмним забезпеченням. Debian відомий відданістю філософії UNIX та вільному програмному забезпеченню. Перевагою Debian є також велика кількість підтримуваних комп'ютерних архітектур – він може працювати на одинадцяти різних видах процесорів, серед яких окрім типових Intel/AMD (які найчастіше використовуються у персональних комп'ютерах) є також архітектури ARM та MIPS, які використовують переважно у вбудованих системах, а також архітектура s390x, яка використовується мейнфреймами серії IBM System Z. На основі операційної системи Debian, створено чимало відомих дистрибутивів, серед яких: Ubuntu, Raspbian, Dream GNU/Linux, Damn Small GNU/Linux, Xandros, Knoppix, Linspire, sidux, Kanotix та багато інших.

Fedora – це дистрибутив GNU/Linux, який розробляється спільнотою Fedora Project і спонсорується компанією Red Hat. Fedora прагне бути повноцінною, універсальною операційною системою, до складу якої входить виключно вільне та відкрите програмне забезпечення. Fedora є відгалуженням від дистрибутиву Red Hat Linux і має намір стати його заміною для домашніх та офісних комп'ютерів. Проект служить для тестування нових технологій, які надалі включаються в продукти Red

Nat та інших виробників. Підтримкою користувачів Fedora на добровільних засадах займається велика Linux-спільнота. Нові випуски Fedora виходять кожні 3-6 місяців.

OpenSUSE – один з дистрибутивів GNU/Linux. Спочатку мав назву SuSE й розроблявся у Німеччині однойменною компанією, яка згодом була придбана американською корпорацією Novell, яка стала спонсором openSUSE та власником кількох похідних від нього комерційних дистрибутивів. openSUSE базується на дистрибутиві Slackware, але відрізняється від останнього зручністю, орієнтацією на підтримку якнайширшого спектру обладнання і увагою до графічного оформлення робочих середовищ. Базується на менеджері пакетів RPM, має хорошу систему адміністрування та керування пакетами YaST власної розробки. Цикл випуску нових версій – від 8 місяців до 1 року.

Ubuntu – операційна система для робочих станцій, ноутбуків і серверів є одним з найпопулярніших дистрибутивів GNU/Linux. Серед основних цілей Ubuntu – надання сучасного і водночас стабільного програмного забезпечення для пересічного користувача із сильним акцентом на простоту встановлення і користування. Ubuntu надає користувачу набір програм загального призначення: багатівіконне настільне середовище, засоби для перегляду Інтернету, організації електронної пошти, офісні програми з можливістю читати і записувати файли у форматі Microsoft Office, редактор зображень, програвач компакт-дисків тощо. Спеціалізоване програмне забезпечення, потрібне досвідченішим користувачам, можна отримати із відповідних репозиторіїв. Серверний варіант системи включає також засоби, потрібні для організації сервера баз даних, веб-сервера, сервера електронної пошти тощо.

Arch Linux – це мінімалістичний, гнучкий дистрибутив GNU/Linux, оптимізований для архітектур i686 та x86-64. Використовує останні стабільні версії програм і доповнюється AUR-репозиторієм. Мінімалістичний дизайн означає відсутність непотрібних для базової роботи програм, включаючи графічний інтерфейс і пакети програм, а гнучкість визначається наявністю добре документованих конфігураційних файлів і можливістю зробити з системи як сервер так і робочу станцію. Початковим інтерфейсом Arch Linux є bash-термінал, саме тому цей дистрибутив вважається дистрибутивом для досвідчених користувачів. Під враженням від операційної системи CRUX, ще одного мінімалістичного дистрибутиву, Джудд Вінет започаткував Arch Linux у березні 2002 року. При цьому Arch Linux є дистрибутивом «from scratch» і не заснований ні на якому іншому дистрибутиві Linux. На відміну від CRUX, Arch GNU/Linux поширюється у вигляді бінарних пакетів.

Slackware – один із найпопулярніших та найстаріший дистрибутивів GNU/Linux, який все ще підтримується (розробник Патрік Волкердінг). Його принципи – це включення лише стабільних випусків програмного забезпечення, які підтримують простоту і стабільність системи, а також відсутність специфічних для дистрибутиву засобів конфігурації. Чимало наразі популярних дистрибутивів були створені саме на основі Slackware Linux. Шанувальникам цього дистрибутиву приписують такий вислів: «Якщо ви знаєте Slackware, ви знаєте Linux. Якщо ви знаєте Red Hat, то все, що ви знаєте – це Red Hat».

FreeBSD – вільна UNIX-подібна операційна система, нащадок AT&T UNIX по лінії BSD, створеної в університеті Берклі. FreeBSD працює на PC-сумісних системах сімейства Intel x86, а також на DEC Alpha, Sun UltraSPARC, IA-64, AMD64, PowerPC, NEC PC-98, ARM, MIPS. FreeBSD розробляється як цілісна операційна система. Вихідний код ядра, драйвери пристроїв, базові програми користувача і командні оболонки розміщені в одному дереві системи керування версіями FreeBSD. Це відрізняє FreeBSD від GNU/Linux – іншої вільної UNIX-подібної операційної системи – у якій ядро розробляється однією групою розробників, а набір програм користувача – іншими (наприклад, проєкт GNU), а численні групи збирають це все в єдине ціле й випускають у вигляді різних дистрибутивів GNU/Linux. FreeBSD добре зарекомендувала себе як система для побудови Інтранет-серверів. Вона надає досить надійні мережні служби й ефективне керування пам'яттю. FreeBSD широко представлена в списку веб-серверів з найбільшим часом безперервної роботи.

Linux Mint – безкоштовний дистрибутив GNU/Linux, заснований на Ubuntu і Debian, ставить за мету надати користувачеві сучасну, елегантну і зручну операційну систему, яка одночасно є потужною і простою у використанні. Linux Mint надає повну підтримку різноманітних форматів мультимедіа, включає в себе деякі пропрієтарні програми і великий набір програм з відкритим вихідним кодом. Засновник проєкту – Клемент Лефевр.

Minix – вільна UNIX-подібна мікроядерна операційна система, яка поширюється на умовах ліцензії BSD. Ендрю Таненбаум створив першу версію Minix у 1987 році. Серед студентів, які вивчали теорію операційних систем по Minix, найбільшої популярності досяг Лінус Торвальдс, розробник ядра Linux. 1 листопада 2008 року Ендрю Таненбаум отримав грант в 2.5 мільйона євро від «Європейської дослідницької ради» для роботи над проєктом Minix. Грант дозволив продовжити роботу з розвитку операційної системи Minix протягом не менше п'яти років.

OpenSolaris – це операційна система з відкритим вихідним кодом, створена корпорацією Sun Microsystems на базі Solaris. Вихідні тексти Solaris були відкриті 14 червня 2005 року. На момент покупки Sun Microsystems компанією Oracle, Sun відкрила вихідні тексти ядра операційної системи, мережних компонентів, системних бібліотек і базових програм для архітектур SPARC і x86. Ці компоненти можна взяти за основу для побудови дистрибутивів сторонніми фірмами, що на даний момент і зроблено як за участю Sun Microsystems, так і без неї. Процес розробки OpenSolaris ведеться на добровільній і безоплатній основі спільнотою розробників OpenSolaris. У майбутньому планується, що розробка нових версій Solaris буде здійснюватись головним чином в проєкті OpenSolaris зусиллями спільноти розробників-ентузіастів. Sun залишає за собою право не додавати в комерційну версію Solaris деякі нові можливості, функції і утиліти, які можуть бути присутніми в OpenSolaris.

PC-BSD – це операційна система, заснована на FreeBSD. Операційна система PC-BSD орієнтована на використання у домашніх комп'ютерах і робочих станціях

та покликана конкурувати з іншими настільними операційними системами, такими як Microsoft Windows, Mac OS та різними дистрибутивами GNU/Linux.

Xenix – UNIX-подібна операційна система, ліцензована Microsoft в AT&T наприкінці 1970-х років. Перші версії працювали на процесорах Intel 80286, потім була введена підтримка Intel 80386. Santa Cruz Operation придбала ексклюзивні права на *Xenix* і з деякого моменту стала поширювати систему як SCO UNIX.

A/UX (від англ. *Apple UNIX* – «UNIX на комп'ютерах Apple») – UNIX-система, створена корпорацією Apple для комп'ютерів Macintosh із платформою m68k. Була заснована на UNIX System V версії 2.2. Відповідала стандартам POSIX і System V Interface Definition. Перша робоча версія вийшла в 1988 році.

1.6. Міжнародний досвід впровадження операційної системи GNU/Linux

Актуальною особливістю операційної системи GNU/Linux є відсутність якого-небудь єдиного власника прав на цю операційну систему і неможливість патентування її з метою комерційної вигоди. Отже, організація, яка використовує GNU/Linux, не ставить себе в залежність від постачальника, що стосується як розробки додаткових можливостей операційної системи і виправлення можливих помилок, так і підтримки прикладних задач, реалізованих у середовищі GNU/Linux. Більше того, деякі країни розглядають GNU/Linux у якості потенційно перспективної області бізнесу для місцевих компаній, що сприяє економічному росту. Тому, аналізуючи офіційні публікації останніх років, треба зазначити досить впевнену тенденцію щодо впровадження операційної системи GNU/Linux та вільного програмного забезпечення (що розповсюджується за ліцензіями – GPL, LGPL, BSD, Apache, MIT) в бюджетних організаціях різних країн. Для прикладу можна навести факти стосовно ефективності масового застосування відкритого програмного забезпечення як у Європі, так і у світі в цілому.

Міністерство освіти Франції та компанія Red Hat у 2007 році здійснили процес переведення 2.5 тисяч серверів, установлених в 30 підрозділах міністерства по всій країні, на використання операційних систем Red Hat Enterprise Linux. У продовження загальнодержавної політики впровадження відкритого програмного забезпечення в бюджетних установах, національна жандармерія Франції провела заміну комерційного програмного забезпечення робочих станцій на вільне, що дозволило заощадити з 2004 року понад 50 мільйонів євро. Адміністрація міста Боблінген (Німеччина) у 2010 році повністю відмовилась від всіх існуючих ліцензій компанії Microsoft, у дванадцяти департаментах міської адміністрації на всіх робочих місцях було встановлено операційну систему Ubuntu, а всі додатки Microsoft Office XP були замінені на відповідні OpenOffice. Федеральний інститут геофізичних досліджень і природних ресурсів Німеччини (BGR) в 2007 році перейшов на Sun Microsystems Solaris.

Міністерство юстиції Бельгії, на підставі рекомендацій уряду країни про розширення сфери застосування вільного програмного забезпечення у державному секторі, прийняло рішення про переведення своїх департаментів на дистрибутив SUSE Linux і на офісний пакет OpenOffice. Міська рада Амстердама (Нідерланди) прийняла рішення щодо повного переходу всіх робочих місць – понад 10 тисяч

комп'ютерів на вільне програмне забезпечення і GNU/Linux. Варто відзначити, що впровадження вільного програмного забезпечення в Нідерландах є державною політикою. Уряд Венесуели ще в листопаді 2008 року закупив для загальноосвітніх закладів країни мільйон ноутбуків із наперед встановленим Ubuntu Linux. Протягом декількох останніх років ряд японських міністерств використовують відкриті рішення і на сьогодні близько 60% комп'ютерів державних установ працює під управлінням GNU/Linux.

У школах Туреччини на всіх комп'ютерах встановлений GNU/Linux замість Microsoft Windows. Турецький університет у місті Адіаман перевів всі свої адміністративні й лабораторні комп'ютери на Pardus Linux. Фахівці університету не бояться проблем, які можуть виникнути при переході на GNU/Linux, а весь позитивний досвід і написані додатки планується поширювати в інших турецьких університетах. Впровадження операційної системи GNU/Linux здійснюється в бюджетних організаціях таких країн як Італія, Китай, Корея, Норвегія, Іспанія, Шотландія, Росія, Австрія та багатьох інших.

Світовий досвід показує, що програмне забезпечення з відкритим вихідним кодом може займати до 50% від загального програмного забезпечення, що використовується при роботі в бюджетних організаціях. Його використання у вирішенні практичних задач різних галузей як розбудови ІТ-інфраструктури так і забезпечення функціонування прикладних програмних засобів, наприклад, організації електронного документообігу є перспективним з огляду на суттєву економію коштів на ліцензування та підтримку робочих місць у сфері діяльності державних органів влади, охорони здоров'я та освіти.

КОНТРОЛЬНІ ПИТАННЯ

- 1) Які функції виконує операційна система?
- 2) Як класифікують операційні системи?
- 3) Які є типи операційних систем?
- 4) Що входить до складу операційної системи?
- 5) Які є вимоги до обладнання при встановленні операційної системи?
- 6) Що включає підсистема ядра операційної системи?
- 7) Що включає підсистема введенням-виведенням операційної системи?
- 8) Що включає підсистема керування пам'яттю у операційній системі?
- 9) В чому сутність організації безпеки операційної системи?
- 10) Які були етапи зародження операційних систем?
- 11) Які операційні системи популярні на даний час?
- 12) Що таке хмарна операційна система?
- 13) Перелічіть етапи розвитку UNIX-систем.
- 14) Перелічіть основні дистрибутиви операційної системи GNU/Linux.
- 15) Яким є міжнародний досвід впровадження операційної системи GNU/Linux?

ЛЕКЦІЙНЕ ЗАНЯТТЯ №2

Тема: Ядро операційної системи. Процеси. Сценарії запуску.

Мета: Формування знань і вмінь, щодо роботи з ядром операційної системи та її процесами.

ТЕОРЕТИЧНІ ВІДОМОСТІ

- 2.1. Ядро операційної системи
- 2.2. Підходи до проектування ядра
- 2.3. Нумерація версій ядра операційної системи GNU/Linux
- 2.4. Kernel panic
- 2.5. Підтримка ядра в розвитку та обслуговуванні
- 2.6. Процес і ідентифікатор процесу
- 2.7. Взаємодіючі процеси
- 2.8. Категорії засобів обміну інформацією
- 2.9. Керування процесами
- 2.10. Життєвий цикл процесу
- 2.11. Список UNIX-процесів
- 2.12. Сценарії запуску в GNU/Linux

2.1. Ядро операційної системи

Ядро (Kernel) – це базовий компонент операційної системи, який реалізує інтерфейс між прикладними процесами та обладнанням комп'ютера. Завантажується в оперативну пам'ять комп'ютера і безпосередньо взаємодіє з апаратною, забезпечуючи керування апаратними засобами (при цьому використовуються драйвери – модулі ядра), підтримку одночасної роботи багатьох користувачів (багатокористувацький режим), підтримку паралельного виконання багатьох процесів в системі (багатозадачність). Зазвичай ядро робить ці об'єкти доступними для прикладних процесів через механізми міжпроцесної взаємодії і системних викликів.

Ядро Linux, розповсюджується під ліцензією GNU General Public License і розробляється та підтримується людьми з усього світу, що дозволяє йому стати одним із найвидатніших прикладів відкритого програмного забезпечення. Ядро Linux підтримує багатозадачність, віртуальну пам'ять, динамічні бібліотеки, відкладене завантаження, продуктивну систему управління пам'яттю і багато мережних протоколів. На сьогоднішній день Linux – монолітне ядро з підтримкою завантажуваних модулів. Драйвери пристроїв і розширення ядра зазвичай запускаються на «кільці 0» (етап завантаження операційної системи) з повним доступом до устаткування. На відміну від звичайних монолітних ядер, драйвери пристроїв легко збираються у вигляді модулів і завантажуються або вивантажуються під час роботи системи.

В ядрі Linux реалізована ціла низка важливих архітектурних елементів. І на найзагальнішому і на детальніших рівнях ядро можна розділити на багато різних підсистем. З іншого боку, Linux можна розглядати як монолітне ціле, оскільки всі

базові сервіси зібрані в ядрі системи. Такий підхід відрізняється від архітектури з мікроядром, коли ядро надає тільки найзагальніші сервіси, такі як обмін інформацією, ввід/вивід, управління пам'яттю і процесами, а конкретніші сервіси реалізуються в модулях, які підключаються до рівня мікроядра. З часом ядро Linux стало ефективнішим з погляду використання пам'яті і процесорних ресурсів і набуло виняткової стабільності. Важливий аспект Linux, враховуючи розмір і складність цієї системи – це її переносність. Linux можна відкомпілювати для величезної кількості різних процесорів, пристроїв і платформ, що мають різні архітектурні обмеження і потреби.

Ядро складається зі статичної частини й динамічної частини (модулів). Крім того, у ядрі розташовуються численні таблиці, розмір яких задається параметрами ядра. Фактично налаштування ядра зводиться до наступних завдань:

1. Визначення (або перевизначення) параметрів ядра.
2. Примусове додавання зазначених модулів у ядро.
3. Примусова заборона завантаження зазначених модулів.

2.2. Підходи до проектування ядра

Монолітне ядро – надає багатий набір абстракцій устаткування. Всі частини монолітного ядра працюють в одному адресному просторі. Монолітні ядра мають довгу історію розвитку та вдосконалення і в цей же час є найбільш архітектурно зрілими і придатними до експлуатації. Разом з тим, монолітність ядер ускладнює їх налаштування, розуміння коду, додавання нових функцій і можливостей, щодо видалення «мертвого» – непотрібного та успадкованого від попередніх версій коду. «Розбухання» коду монолітних ядер також підвищує вимоги до обсягу оперативної пам'яті, потрібного для функціонування ядра операційної системи. Це робить монолітну архітектуру мало придатною для експлуатації в системах, які обмежені обсягом пам'яті, наприклад у виробничих мікроконтролерах.

Переваги використання монолітних ядер – це швидкість роботи, спрощена розробка модулів, різноманіття надаваних можливостей і функцій, підтримка великої кількості різноманітного устаткування. Монолітне ядро більш продуктивніше, ніж мікроядро, оскільки складається не з множини розрізнених процесів, «що спілкуються» між собою через механізм надсилання повідомлень, а працює в одному адресному просторі, крім того немає потреби здійснювати перемикання між процесами.

Недоліками використання монолітних ядер є те, що все ядро працює в одному адресному просторі, збій в одному з компонентів може порушити працездатність всієї системи. У цьому випадку компоненти операційної системи є не самостійними модулями, а складовими частинами однієї великої програми. Тому така організація роботи операційної системи якраз називається монолітним ядром. Монолітне ядро є набором процедур, які працюють у привілейованому режимі.

Модульне ядро – це сучасна, удосконалена модифікація архітектури монолітного ядра. На відміну від «класичних» монолітних ядер, які нині вважаються застарілими, модульні ядра, зазвичай, не вимагають повної перекомпіляції ядра при зміні складу апаратного забезпечення комп'ютера. Замість

цього модульні ядра надають той або інший механізм підвантаження модулів ядра, які підтримують те, або інше апаратне забезпечення (наприклад, драйвери). При цьому, додавання модулів у ядро може бути як динамічним – без перезавантаження операційної системи так і статичним з перезавантаженням операційної системи та переконфігуруванням системи. Всі модулі працюють в одному адресному просторі ядра і можуть користуватись всіма функціями, що надаються ним. Тому модульні ядра продовжують залишатися монолітними.

Модульне ядро зручніше для створення і розгортання, ніж традиційне монолітне ядро, яке не підтримує динамічне перезавантаження модулів. Виявлення, локалізація, налаштування й усунення помилок при тестуванні також полегшуються при роботі з модульним ядром. Модульні ядра надають особливий програмний інтерфейс (API) для зв'язування модулів з ядром та забезпечення динамічного підвантаження і вивантаження модулів. У свою чергу, не будь-яка програма може бути модулем ядра: на модуль ядра накладаються певні обмеження в частині використовуваних функцій (наприклад, вони не можуть користуватися функціями стандартних бібліотек мови програмування C/C++ і повинні використовувати спеціальні аналоги, які є функціями API-ядра). Крім того, модулі ядра зобов'язані експортувати певні функції, потрібні ядру для правильного підключення й розпізнавання модуля, для його коректної ініціалізації при завантаженні й коректному завершенні при вивантаженні.

Мікроядро – це мінімальна реалізація функцій ядра операційної системи. Класичні мікроядра, реалізують лише дуже обмежений набір низькорівневих примітивів або системних викликів, які представляють собою базові сервіси операційної системи.

До них відносяться:

- Управління адресним простором оперативної пам'яті.
- Управління адресним простором віртуальної пам'яті.
- Управління процесами і потоками.
- Засоби міжпроцесної взаємодії.

Всі інші сервіси операційної системи, які в класичних монолітних ядрах реалізуються безпосередньо їм, в мікроядерній архітектурі реалізуються в адресному просторі користувача. Прикладами таких сервісів є файлова система і драйвери. Така конструкція дозволяє збільшити загальну швидкодію і час реакції системи. Невелике ядро може, поміститись в кеші процесора. Приклади операційних систем на основі мікроядра – QNX, L4, Minix3, GNU/Hurd.

Екзоядро – це тип ядра, яке не абстрагує апаратні засоби у «теоретичну» модель, замість цього воно розподіляє всі апаратні ресурси, включаючи процесорний час, пам'ять та блоки диска, між усіма програмами операційної системи. Програма, яка запущена на екзоядрі використовує бібліотеки операційної системи для підвищення швидкодії.

Наноядро – це архітектура ядра операційної системи, у рамках якої вкрай спрощене й мінімалістичне ядро виконує лише одне завдання – обробку апаратних переривань, які генеруються пристроями комп'ютера. Після обробки переривання від обладнання, наноядро надсилає інформацію про результати обробки,

відповідному програмному забезпеченню. Прикладом операційної системи на основі наноядра є KeuKOS, перша версія якої вийшла у 1983 році.

Гібридне ядро – це модифіковане мікроядро, яке використовується для прискорення запуску несуттєвих частин в просторі ядра. Гібридне ядро є компромісом між монолітним і мікроядром. Всі розглянуті підходи до побудови операційних систем мають свої переваги й недоліки. У більшості випадків сучасні операційні системи використовують різні комбінації цих підходів. Так, наприклад, ядро Linux є монолітною системою з елементами мікроядерної архітектури. При компіляції ядра можна виконувати динамічне завантаження й вивантаження багатьох його компонентів – так званих модулів. У момент завантаження модуля його код завантажується на рівні системи й зв'язується з іншою частиною ядра. Всередині модуля можуть використовуватися будь-які експортовані ядром функції.

2.3. Нумерація версій ядра операційної системи GNU/Linux

GNU/Linux – складна операційна система і процедура її завантаження не зводиться до простого натискання кнопки на системному блоці. Щоб система працювала коректно, потрібно виконувати операції запуску й вимкнення за всіма правилами, які визначені у її ядрі. Номер версії ядра Linux включає чотири числа. Допустимо, що номер версії ядра складається, максимально, з 4 букв: A.B.C.D (наприклад, 5.3 або 2.6.12.3).

- Число A позначає версію ядра. Воно змінюється найменш часто й тільки тоді, коли вносяться значні зміни в код і концепцію ядра (наприклад, в 1994 рік – версія 1.0, 1996 рік – версія 2.0).
- Число B позначає старшу версію релізу ядра. Парні числа позначають стабільні ревізії, тобто ті, які призначені для масового використання, не парні числа позначають ревізії для розробників – призначені для тестування нових технологій і драйверів.
- Число C позначає молодшу версію релізу ядра. Змінюється, коли в ядро включають елементи пов'язані з інформаційною безпекою, виправленням помилок і додаванням драйверів.
- Число D вперше з'явилося після випадку, коли в коді ядра версії 2.6.8 була виявлена груба помилка, пов'язана з NFS. Тоді випустили версію 2.6.8.1 з єдиним виправленням у NFS. З ядра версії 2.6.11 ця нумерація була адаптована в якості нової офіційної політики.

2.4. Kernel panic

Kernel panic («Паніка ядра») – це повідомлення про критичну помилку ядра операційної системи, після якої вона вже не може функціонувати. Цей термін широко використовується в середовищі операційних систем сімейства UNIX. Назва походить від повідомлення про помилку – «Kernel panic: ...» і функції ядра panic(). У Linux виникненню «паніки ядра» найчастіше передують стан під назвою oops. Повідомлення Kernel panic було введено ще в ранніх версіях UNIX. У більшості випадків причиною для Kernel panic є критична апаратна помилка (відмова пам'яті або іншого важливого комп'ютерного пристрою) або помилка в самому ядрі

операційної системи, наприклад спроба звертання до помилкової або забороненої адреси в пам'яті. Також причиною для Kernel panic можуть бути помилки в драйверах периферійних пристроїв або помилки у файловій системі.

У випадку виникнення Kernel panic відбувається зупинка роботи операційної системи з виведенням повідомлення про помилку на екрані монітора, після чого система очікує вимкнення комп'ютера або перезавантаження. У сучасних операційних системах, таких як GNU/Linux, FreeBSD або Solaris, існує можливість змінити стандартне поведіння функції panic() і здійснити перезавантаження комп'ютера автоматично. В GNU/Linux таке налаштування здійснюється за допомогою procfs. Для виявлення причини, що призвела до помилки Kernel panic потрібно дослідити файл System.map.

2.5. Підтримка ядра в розвитку та обслуговуванні

Якщо Linux включає тисячі різних пакетів, від найпоширеніших, таких як утиліти GNU, X.Org, графічні середовища GNOME і KDE, до специфічних для певного дистрибутиву. Команди розробників кожного проекту, відрізняються чисельністю, підходом до роботи, використовуваними інструментами і методами планування робіт. При цьому ядро Linux займає особливе місце серед всіх інших застосувань, від нього залежить можливість роботи операційної системи на різних апаратних платформах і ступінь підтримки різних пристроїв.

Linux Foundation проводить дослідження, що охоплює трирічний період розробки ядра і фокусується на таких характеристиках, як частота релізів та розмір початкових текстів ядра. Статистика демонструє, що в середньому, кожен новий реліз ядра виходить раз на 2 – 3 місяці. У першу чергу це пов'язано з вибраною у 2005 році моделлю розробки, спрямованою на зведення до мінімуму тривалості проміжків між розробкою нових функцій, появою підтримки нових пристроїв і включенням їх в ядро. При цьому число патчів, які вносяться до ядра – зростає. Чотири всесвітньо відомі компанії: Red Hat, Novell, IBM та Intel – внесли понад 32% зроблених за час дослідження змін, а сумарний відсоток змін в ядро, внесених їхніми працівниками становить 70%.

2.6. Процес і ідентифікатор процесу

Процес – це абстракція, використовувана в UNIX та GNU/Linux для опису виконуваних програми. Процес є системним об'єктом, за допомогою якого можна контролювати звернення програми до оперативної пам'яті, центрального процесору й ресурсів вводу-виводу. Процес складається з адресного простору і набору структур даних, що є у ядрі. Адресний простір – це сукупність сторінок оперативної пам'яті, які були виділені ядром для виконання процесу. В адресний простір завантажується код процесу й використовувані їм бібліотеки функцій, а також змінні, вміст стека і різна допоміжна інформація, необхідна ядру під час роботи процесу. Оскільки GNU/Linux є системою в якій підтримується концепція віртуальної пам'яті, то сторінки адресного простору процесів у конкретний момент часу, можуть перебувати або в фізичній пам'яті, або в розділі підкачування на диску. До найбільш важливої інформації про кожний процес, входять:

- таблиця розподілу пам'яті;
- поточний статус (неактивний, припинений, виконується, «зомбі»);
- пріоритет;
- інформація про використовувані ресурси;
- інформація про файли й мережеві порти;
- інформація про власника процесу.

Деякі з цих атрибутів можуть спільно використовуватися декількома процесами, які утворюють «поточкову групу», що є Linux-аналогом багатопоточного процесу в традиційній системі UNIX. Хоча ці процеси можуть спільно використовувати адресний простір, на практиці лише деякі з них цікаві для системних адміністраторів. Багато характеристик процесу безпосередньо впливають на його виконання, зокрема, має значення, скільки часу виділяється йому центральним процесором, до яких файлів він має доступ та інше. Далі розглянемо найцікавіші з погляду системного адміністратора, характеристики процесів (вони аналогічні для всіх версій UNIX і GNU/Linux).

Ідентифікатор процесу (PID). Ядро призначає кожному процесу унікальний ідентифікатор (Process ID, PID). Більшість команд і системних викликів, які працюють з процесами, вимагають задання конкретного ідентифікатора, щоб був зрозумілий контекст операції. Процес отримує ідентифікатор під час свого створення.

Ідентифікатор батьківського процесу (PPID). В GNU/Linux немає системного виклику, який створював би новий процес для виконання конкретної програми. Існуючий процес повинен клонувати сам себе, щоб породити новий. В операції клонування вихідний процес називають батьківським, а його клон – дочірнім. Крім власного ідентифікатора, кожний дочірній процес має атрибут PPID (Parent Process ID) – це інформація про батьківський процес.

Ідентифікатор користувача (UID) та ефективний ідентифікатор користувача (EUID). UID (User ID) – це ідентифікатор користувача, що створив даний процес. EUID (Effective User ID) – це «ефективний» або поточний ідентифікатор процесу користувача, призначений для того, щоб визначити, до яких ресурсів і файлів у процесу є право доступу в цей момент. У більшості процесів, значення UID і EUID однакові. Виключення становлять програми у яких встановлений біт зміни ідентифікатора користувача. Два ідентифікатора потрібні, щоб розмежувати поняття персоніфікації й прав доступу. До того ж програми з встановленим бітом зміни не завжди виконуються з розширеними привілеями. Значення EUID можна встановлювати й скидати, щоб надавати процесу додаткові повноваження або відбирати їх.

2.7. Взаємодіючі процеси

Для досягнення ядром поставленої мети різні процеси можуть виконуватись псевдопаралельно на одній обчислювальній системі або паралельно на різних обчислювальних системах, взаємодіючи між собою. Така організація роботи, сприяє:

- Підвищенню швидкості роботи операційної системи. Поки один процес

очікує настання деякої події (наприклад, закінчення операції вводу-виводу), інші можуть виконувати завдання загального призначення. У багатопроцесорних обчислювальних системах програма розбивається на окремі частини, кожна з яких виконується на своєму процесорі.

- Спільному використанню даних. Різні процеси можуть працювати з спільною базою даних або файлами, разом змінюючи їхній вміст.
- Модульній конструкції системи. Типовим прикладом може служити мікроядерний спосіб побудови операційної системи, коли різні її частини є окремими процесами, взаємодіючи при цьому шляхом передачі повідомлень через мікроядро.

Процеси не можуть взаємодіяти, не спілкуючись, тобто не обмінюючись інформацією. «Спілкування» процесів приводить до зміни їхнього поведіння залежно від отриманої інформації. Якщо діяльність процесів залишається незмінною при будь-якій прийнятій ними інформації, то це означає, що вони насправді в «спілкуванні» не мають потреби. Процеси, які впливають на поведінку один одного шляхом обміну інформацією, прийнято називати кооперативними або взаємодіючими процесами.

2.8. Категорії засобів обміну інформацією

Процеси можуть взаємодіяти один з одним, тільки обмінюючись інформацією. За обсягом переданої інформації та ступенем можливого впливу на поведінку іншого процесу, всі засоби такого обміну можна розділити на три категорії:

- Сигнальні. Передається мінімальний обсяг інформації – один біт. Використовуються, зазвичай, для повідомлення процесу про настання якої-небудь події. Ступінь впливу на поведінку процесу, що отримав таку інформацію, мінімальний. Надалі, все залежить від того, чи знає він, що означає отриманий сигнал, чи треба на нього реагувати і яким чином. Неправильна реакція на сигнал або його ігнорування може привести до трагічних наслідків.
- Канальні. «Спілкування» процесів відбувається через лінії зв'язку, надані операційною системою і нагадує спілкування людей по телефону, за допомогою записок, листів або оголошень.
- Розділена пам'ять. Два або більше процесів можуть спільно використовувати деяку область адресного простору. Поділом пам'яті займається операційна система. «Спілкування» процесів нагадує спільне проживання студентів в одній кімнаті гуртожитку. Використання розділеної пам'яті для передачі (отримання) інформації здійснюється за допомогою засобів звичайних мов програмування.

2.9. Керування процесами

Команда ps. насамперед потрібно визначати, які процеси запущені в системі. Для цього в GNU/Linux (як і у всіх UNIX-системах) є команда *ps*. Якщо вказану команду запустити без параметрів, то вона виведе на екран список процесів,

запущених у поточній сесії. Якщо ж необхідно побачити список всіх процесів, запущених у системі, слід використати параметр -ax.

Існує три рівноправних формати задання цієї команди:

ps [-опції]

ps [опції]

Залежно від заданих опцій, можуть бути відображені наступні поля:

- USER – ім'я власника процесу;
- PID – ідентифікатор процесу в системі;
- PPID – ідентифікатор батьківського процесу;
- %CPU – частка використання (утилізації) часу центрального процесора (у відсотках);
- %MEM – частка використання (утилізації) реальної пам'яті (у відсотках);
- VSZ – віртуальний розмір процесу (у кілобайтах);
- RSS – розмір резидентного набору (кількість 1К-сторінок у пам'яті);
- STIME – час старту процесу;
- TTY – вказівка на термінал, з якого запущений процес;
- S або STAT – статус процесу;
- PRI – пріоритет;
- NI – значення нісе;
- TIME – скільки процесорного часу центрального зайняв даний процес.

Значення, виведені в більшості цих полів зрозумілі без додаткових пояснень.

В полі STAT, як вже говорилося раніше, може стояти наступне значення:

- R – призупинений процес, який очікує тільки моменту, коли планувальник завдань виділить йому черговий квант часу;
- S – процес «спить»;
- D – процес перебуває в стані підкачки на диску;
- T – зупинений процес;
- Z – процес-зомбі.

Ось кілька прикладів застосування команди ps:

5) Для того, щоб побачити всі процеси запущені в системі – [user]\$ ps -ax.

6) Для того, щоб побачити всі процеси «предок-нащадок» в системі – [user]\$ ps -ef.

7) Для того, щоб побачити, скільки відсотків процесорного часу займають процеси – [user]\$ ps -u.

8) Щоб довідатися пріоритет процесу – [user]\$ ps -l.

Команда top. Команда ps дозволяє зробити як би «миттєвий знімок» процесів, запущених у системі. На відміну від ps команда top відображає стан процесів та їхню активність в реальному режимі часу. Після запуску команди top у верхній частині вікна відображається час, що пройшов з моменту запуску системи, число користувачів у системі, число запущених процесів і число процесів, які перебувають у різних станах, дані про використання центрального процесора, оперативної пам'яті і файла підкачки. А далі відображається таблиця, що

характеризує окремі процеси. Число рядків, відображуваних у цій таблиці, визначається розміром вікна. Графи таблиці позначені так само, як поля команди ps. Табличні дані оновлюються кожні 5 секунд. Список процесів може бути відсортований за використанням часом центрального процесора (за замовчуванням), обсягом пам'яті, за PID, за часом виконання. Перемикати режими відображення можна за допомогою наступних команд:

- Shift+N – сортування за PID процесу;
- Shift+A – сортування за тривалістю існування процесу;
- Shift+P – сортування за використанням центрального процесора;
- Shift+M – сортування за використанням оперативної пам'яті;
- Shift+T – сортування за часом виконання.

Крім команд, що визначають режим сортування, команда top сприймає ще ряд команд, які дозволяють керувати процесами в інтерактивному режимі. За допомогою команди <K> можна завершити деякий процес, а за допомогою команди <R> можна перевизначити значення nice для деякого процесу. Отже, ці дві команди аналогічні командам kill і renice, які розглянемо далі.

Пріоритети, значення nice і команда renice. Пріоритет для кожного процесу встановлюється в той момент, коли він породжується. Пріоритет процесу визначається так званим «значенням nice», що лежить у межах від +20 (найменший пріоритет) до -20 (найвищий пріоритет). Значення пріоритету процесу можна змінювати командою nice. Формат запуску команди nice, наступний: nice [-adnice] command, де adnice – значення (від -20 до +19) пріоритетності процесу, command – ім'я процесу. Команда renice, служить для зміни значення nice для вже виконуваних процесів. Користувачі системи можуть змінювати значення пріоритету тільки для тих процесів, власниками яких вони є.

Сигнали й команда kill. Сигнали – це засіб, за допомогою якого процесам можна передати повідомлення про деякі події в системі. Самі процеси теж можуть генерувати сигнали, за допомогою яких вони передають певні повідомлення ядру операційної системи й іншим процесам. За допомогою сигналів можна здійснювати такі акції керування процесами, як призупинення процесу, запуск призупиненого процесу, завершення роботи процесу.

Усього в GNU/Linux є 63 різні сигнали, їхній перелік можна переглянути запустивши команду [user]\$ kill -l. Сигнал прийнято позначати номером або символічним іменем. Команда kill дозволяє надіслати заданому процесу будь-який сигнал. Для надсилання сигналу процесу можна скористатися командою kill у наступному форматі: [user]\$ kill [-сигн] PID.

2.10. Життєвий цикл процесу

Для створення нового процесу існуючий процес, зазвичай, клонує сам себе за допомогою системного виклику fork. У результаті формується копія вихідного процесу, який має лише деякі відмінності. Зокрема, новому процесу надається власний ідентифікатор і ресурси. Системний виклик fork має унікальну властивість – він повертає два різних значення. У дочірньому процесі це буде 0, а в

батьківському – ідентифікатор процесу-нащадка. Оскільки в іншому процесі ідентичні, вони повинні перевіряти до кого надійшов запит.

Після завершення системного виклику `fork`, дочірній процес запускає нову програму за допомогою одного із системних викликів сімейства `exec`. Всі виклики цього сімейства виконують приблизно однакові дії: вони заміщують сегмент коду процесу та встановлюють сегменти даних і стека у початковий стан. Коли система завантажується, ядро операційної системи автоматично запускає кілька процесів. Найважливіший з них – це `init`, ідентифікатор якого завжди дорівнює 1. Процес `init` відповідає за виконання сценаріїв запуску системи. Для завершення деякої програми процес `init` викликає функцію `exit()`. Проте, перш ніж процес остаточно зникне, його видалення має бути узгоджене з батьківським процесом за допомогою системного виклику `wait`.

2.11. Список UNIX-процесів

В наступній таблиці подано список UNIX-процесів, які можуть бути присутні в різних її модифікаціях, наприклад в GNU/Linux.

Процес	Опис
<code>init</code>	UNIX програма, яка породжує всі інші процеси.
<code>dhcpd</code>	Динамічно визначає конфігурацію TCP/IP.
<code>fingerd</code>	Забезпечує мережний інтерфейс для протоколу <code>finger</code> .
<code>ftpd</code>	Служба FTP.
<code>httpd</code>	Процес веб-сервера.
<code>inetd</code>	Відслідковує мережеві запити.
<code>lpd</code>	Протокол керування друком.
<code>ntpd</code>	Синхронізація часу в мережі.
<code>sshd</code>	Прослуховує запити – <code>secure shell</code> (<code>ssh</code>).
<code>syslogd</code>	Процес журналювання системних повідомлень.

2.12. Сценарії запуску в GNU/Linux

В наступній таблиці подано список сценаріїв запуску в операційній системі GNU/Linux.

Служба	Призначення
<code>acpid</code>	Здійснює підтримку системи ACPI – реакція системи на різні сигнали керування живлення.
<code>amd</code>	Керує роботою процесу <code>automount</code> – автоматичне монтування локальних і мережевих файлових систем.

arpwatch	Стежить за відповідністю між IP-адресою і фізичною адресою мережевої карти.
atd	Керує роботою процесу at (планувальник завдань).
autofs	Автоматичне монтування файлових систем.
bluetooth	Підтримка системи bluetooth.
crond	Виконання регулярних завдань за розкладом.
dhcpcd	Керує динамічним вибором конфігурації хоста (Dynamic Host Control Protocol).
gated	Керує процесом маршрутизації.
halt	Керує процесом вимкнення й перезавантаження системи.
httpd	Керує Web-сервером Apache (служба HTTP).
identd	Керує сервером ідентифікації користувачів.
ip6tables	Керує брандмауером пакетів IPv6
iptables	Утиліта фільтрації пакетів.
killall	Допоміжний сценарій, який використовують amd, crond, inet, nfs, init для знищення процесів.
kudzu	Керування перевіркою й конфігурацією апаратного забезпечення комп'ютера.
ldap	Протокол керування доступом до каталога.
lircd	Підтримка віддаленого керування за допомогою пристроїв, які використовують інфрачервоний порт.
mailman	Підтримка поштових програм.
mysqld	Підтримка системи MySQL.
named	Керує запуском і зупинкою Domain Name Service.
netfs	Керує монтуванням файлових систем NFS.
network	Керує запуском і зупинкою системних мережевих служб.
nfs	Керує роботою Network File System (мережева файлова система).
ntpd	Процес Network Time Protocol (NTP) – синхронізація часу.
portmap	Керує роботою служби Remote Procedure Call.
postgresql	Керує роботою бази даних PostgreSQL.
privoxy	Проксі-сервер.

radvd	Скрипти маршрутизації пакетів IPv6.
sendmail	Керує поштовою службою.
smartd	Система підтримки моніторингу дискових пристроїв S.M.A.R.T.
smb	Керує роботою системи Samba – файлова організація взаємодії UNIX і Windows.
squid	Керує роботою служби кешування об'єктів httpd.
sshd	Керує сервером OpenSSH.
syslog	Служба System Logging (ведення системних журналів).
wine	Вільний емулятор Windows.
xfs	Сервер шрифтів системи X11.
yum	Система автоматичного відновлення пакетів.
zebra	Система маршрутизації пакетів.

КОНТРОЛЬНІ ПИТАННЯ

- 1) Перелічіть складові частини ядра операційної системи.
- 2) Чим характерне мікроядро?
- 3) Як відбувається керування процесами у операційній системі?
- 4) Чим характерне наноядро?
- 5) Які бувають системні виклики у операційній системі?
- 6) Чим характерне екзоядро?
- 7) Яка історія розвитку різних типів ядер?
- 8) Що таке ядро операційної системи?
- 9) Як відбувається нумерація версій ядра операційної системи GNU/Linux?
- 10) Що таке «Kernel panic»?
- 11) Чим характерне гібридне ядро?
- 12) Що таке процес і ідентифікатор процесу GNU/Linux?
- 13) Що таке взаємодіючі процеси?
- 14) Чим характерне монолітне ядро?
- 15) Перелічіть список UNIX-процесів?

ЛЕКЦІЙНЕ ЗАНЯТТЯ №3

Тема: Концепції файлової системи. Ієрархія логічних файлових систем GNU/Linux.

Мета: Формування знань і вмінь, щодо роботи з файловою системою в GNU/Linux.

ТЕОРЕТИЧНІ ВІДОМОСТІ

- 3.1. Типи файлів в UNIX
- 3.2. Файлова система
- 3.3. Структура файлової системи на диску
- 3.4. Список основних файлових систем
- 3.5. Файлові системи GNU/Linux
- 3.6. Ієрархія логічних файлових систем
- 3.7. Монтування файлових систем
- 3.8. Цілісність файлової системи
- 3.9. Журналювання файлів
- 3.10. Поточний і домашній каталог
- 3.11. Жорсткі і символічні посилання

3.1. Типи файлів в UNIX

Файл – це іменована область даних на носії інформації, робота з файлами реалізується засобами операційних систем. Багато операційних систем привінують до файлів і обробляють подібним чином та інші ресурси: області даних, порти або принтери, потоки даних, мережеві ресурси, сокети. В операційній системі GNU/Linux всі дані, які зберігаються на носіях, обов'язково перебувають всередині певного файла – інакше вони просто недоступні ні для операційної системи ні для користувачів. Більше того, багато пристроїв, підключених до комп'ютера (жорсткий диск, принтер, сканер) GNU/Linux відображає (формує) у вигляді файлів.

В операційній системі UNIX існують 6 типів файлів, що різняться за функціональним призначенням і діях операційної системи при виконанні тих або інших операцій над ними:

- Звичайний файл (regular file).
- Каталог (directory).
- Спеціальний файл пристрою (special device file).
- FIFO або іменованний канал (named pipe).
- Зв'язок (link).
- Сокет.

Звичайний файл є найбільш загальним типом файлів, який містить дані в деякому форматі. Для операційної системи такі файли є простою послідовністю байтів. Вся інтерпретація вмісту файла виконується прикладною програмою, яка обробляє файл. До таких файлів відносяться, наприклад текстові файли.

Каталог. За допомогою каталогів формується логічне дерево файлової системи. Каталог – це файл, який містить імена файлів, що перебувають у ньому, а

також покажчики на додаткову інформацію – метадані, які дозволяють операційній системі здійснювати операції над ними. Каталоги визначають положення файла в дереві файлової системи, оскільки сам файл не містить інформації про своє місцезнаходження.

Спеціальний файл пристрою забезпечує доступ до фізичного пристрою. В операційній системі UNIX розрізняють символні (character) і блокові (block) файли пристроїв. Доступ до пристроїв здійснюється шляхом відкриття, читання й запису в спеціальний файл пристрою.

FIFO або іменованний канал – це файл, який використовується для зв'язку між процесами. FIFO вперше з'явилися в System V UNIX.

Зв'язок. Як вже говорилось, каталог містить імена файлів і покажчики на їх метадані. В той же час самі метадані не містять ні імені файла, ні покажчика на це ім'я. Така архітектура дозволяє одному файлу мати кілька імен у файловій системі.

Сокети використовуються для взаємодії між процесами. Інтерфейс socket використовується, наприклад для доступу до мережі TCP/IP.

3.2. Файлова система

Файлова система – це частина операційної системи, призначення якої полягає в тому, щоб організувати ефективну роботу з даними, що зберігаються в зовнішній пам'яті, і забезпечити користувачеві зручний інтерфейс при роботі з такими даними. Організувати зберігання інформації на магнітному диску непросто. Файлова система – спосіб зберігання й організації доступу до даних на інформаційному носії або його розділі. Класична файлова система має ієрархічну структуру у якій файл однозначно визначається повним шляхом до нього.

Файлова система потрібна для забезпечення:

- 1) зручного інтерфейсу для користувача при роботі з даними, що зберігаються на диску;
- 2) спільного використання файлів кількома користувачами і процесами.

У широкому розумінні поняття «файлова система» включає:

- 1) сукупність всіх файлів на диску;
- 2) набір структур даних, використовуваних для керування файлами: каталоги файлів, дескриптори файлів, таблиці розподілу вільного і зайнятого простору на диску;
- 3) комплекс системних програмних засобів, які реалізують функції управління файлами, зокрема: створення, видалення, читання, запис, пошук та інше.

Основні функції файлової системи:

1. Ідентифікація файлів. Зв'язування імені файла з виділеним йому адресним простором зовнішньої пам'яті.
2. Розподіл зовнішньої пам'яті між файлами. Для роботи з конкретним файлом користувачеві не потрібно мати інформацію про місце розташування цього файла на зовнішньому носії інформації.
3. Забезпечення надійності. Збереження даних та інформації.
4. Забезпечення захисту від несанкціонованого доступу.

5. Забезпечення спільного доступу до файлів, так щоб користувачеві не доводилося докладати спеціальних зусиль по забезпеченню синхронізації доступу.
6. Забезпечення високої продуктивності.

Для більшості користувачів файлова система – найбільш видима частина операційної системи. Вона надає механізм для зберігання й доступу як до даних, так і до програм, всім користувачам системи. З погляду користувача, файл – це одиниця зовнішньої пам'яті, тобто дані, записані на диск. Програміст сприймає файл у вигляді набору однорідних записів. Запис – це найменший елемент даних, що може бути оброблений як єдине ціле прикладною програмою при обміні із зовнішнім пристроєм. Причому в більшості операційних систем розмір запису дорівнює одному байту. Далі записи поєднуються в блоки.

3.3. Структура файлової системи на диску

Кластер – логічна одиниця зберігання даних у таблиці розміщення файлів, яка поєднує групу секторів.

Кеш – проміжний буфер зі швидким доступом, що містить копію тієї інформації, який зберігається в пам'яті з повільним доступом. Доступ до даних у кеші здійснюється швидше, ніж вибірка вихідних даних з повільної пам'яті.

Сектор диска – мінімальна адресована одиниця зберігання інформації на дискових запам'ятовуючих пристроях. Є частиною доріжки диска.

На фізичному рівні робота з накопичувачами відбувається у всіх файлових системах однаково, на рівні команд:

- 1) підвести зчитуючий/записуючий елемент до вказаного місця (сектора);
- 2) зчитати дані з вказаного місця (сектора);
- 3) записати дані у вказане місце (сектор).

На початку розділу є суперблок, який містить загальний опис файлової системи:

- тип файлової системи;
- розмір файлової системи в блоках;
- розмір масиву індексних вузлів;
- розмір логічного блоку.

Описані структури даних створюються на диску в результаті його форматування. Їхня наявність дозволяє звертатися до даних на диску як до файлової системи, а не як до звичайної послідовності блоків. У файлових системах сучасних операційних систем для підвищення стабільності підтримується кілька копій суперблоку.

3.4. Список основних файлових систем

- AdvFS – Advanced File System, створена в Digital Equipment Corporation для операційної системи Digital UNIX/Tru64.
- AFS – Acer Fast Filesystem, використовується в SCO OpenServer.
- FFS (Amiga) – Fast File System, використовується в операційних системах сімейства AmigaOS.
- BFS – Be File System, використовується в операційній системі BeOS.

- Btrfs – файлова система, яка створена Oracle Corporation у 2007 році.
- EFS – Encrypted filesystem – розширення для NTFS.
- Ext – розширена файлова система, розроблена для операційних систем на ядрі Linux.
- ext2 – друге розширення файлової системи, розроблена для операційних систем на ядрі Linux.
- ext3 – журнальована версія ext2.
- ext4 – журнальована файлова система, яка використовується в операційних системах на ядрі Linux.
- FAT – використовується в дисковій операційній системі DOS та операційній системі Microsoft Windows.
- FFS – Fast File System, використовується в операційних системах сімейства BSD.
- Fossil – архівна файлова система від Plan 9 Bell Labs.
- HPFS – High Performance Filesystem, використовується в операційній системі OS/2.
- ISO 9660 – використовується для монтування CD і DVD дисків.
- JFS – журнальована файлова система, створена корпорацією IBM.
- Minix – використовується в операційній системі Minix.
- Qnx4fs – файлова система, яка використовується в операційній системі QNX.
- ReiserFS – файлова система для операційної системи GNU/Linux.
- Reiser4 – файлова система, здатна до журналювання – «реінкарнація» ReiserFS.
- STL – Standard language file system, файлова система створена в корпорації IBM.
- UFS – UNIX Filesystem, стандартна файлова система, яка використовується в операційних системах сімейства BSD.
- UFS2 – UnixFilesystem, оновлена редакція файлової системи UFS.
- XFS – використовується в операційних системах IRIX, GNU/Linux і BSD.
- ZFS – Zettabyte filesystem, відкрита файлова система від Sun Microsystems.

3.5. Файлові системи GNU/Linux

ext2 – файлова система, яка розроблена спеціально під операційну систему GNU/Linux. Вона досить швидка для того, що б бути еталоном у тестах швидкодії файлових систем, але не є такою і це її головний недолік. Всі типи файлів *ext2* мають символні імена. Обмеження на просте ім'я – це 255 символів. В імені файла не повинні бути присутні символи NUL і «/». Обмеження на символ NUL, пов'язане з поданням рядків мовою C, а на символ «/» з тим, що він використовується як розділовий символ між каталогами.

Основними атрибутами файла в *ext2* є:

- тип і права доступу до файла;
- власник, група;
- інформація про дозволені операції доступу до файла;

- час створення, останнього доступу та зміни;
- поточний розмір файла;
- тип файла;
- інші.

Атрибути файлів зберігаються не в каталогах, як це зроблено в ряді простих файлових систем, а в спеціальних таблицях. У результаті каталог має дуже просту структуру, яка складається всього із двох частин: номера індексного дескриптора й імені файла. Весь адресний простір розділу жорсткого диска розділено на блоки фіксованого розміру. Розмір блока вказується при створенні файлової системи на розділі жорсткого диска. Кожен блок має порядковий номер. З метою зменшення фрагментації й числа переміщень голівки жорсткого диска при читанні, блоки поєднуються в групи блоків. Кожна група блоків має однакову будову. Суперблок – це основний елемент файлової системи ext2, він містить загальну інформацію про:

- загальне число блоків та індексних дескрипторів;
- число вільних блоків та індексних дескрипторів;
- розмір блоку файлової системи;
- число блоків та індексних дескрипторів у групі;
- розмір індексного дескриптора;
- ідентифікатор файлової системи.

Від цілісності суперблоку, напряму залежить працездатність файлової системи. Операційна система створює кілька резервних копій суперблоку для можливості його відновлення у випадку пошкодження.

ext3 – це журнальована файлова система, яка використовується в операційних системах на ядрі Linux. Заснована на файловій системі ext2. Основна відмінність від ext2fs полягає в тому, що ext3 журнальована, тобто в ній передбачено запис деяких даних, які дозволяють відновити файлову систему при збоях у роботі комп'ютера. Файлова система ext3 підтримує файли розміром до 4 ТБ.

Передбачено два режими журналювання:

- *writeback*: у файл журналювання записуються тільки метадані файлової системи, тобто інформація про її зміну;
- *journal*: повне журналювання як метаданих файлової системи, так і даних користувача.

ext4 – четверта версія розширеної файлової системи, журнальована, використовується в операційних системах з ядром Linux. Головна особливість ext4 – це збільшення максимального обсягу одного розділу жорсткого диска до 1 екзабайта. Крім того, в ext4 реалізовано механізм просторового запису файлів (нова інформація додається в кінець попереднього блоку), що зменшує фрагментацію і збільшує продуктивність.

Особливості файлової системи ext4:

- максимальний розмір файла – 16 ТБ;
- максимальний розмір розділу на жорсткому диску 1 екзабайт.

ReiserFS – журнальована файлова система, розроблена спеціально для операційної системи GNU/Linux компанією Namesys. Слід відмітити, що файлова система ReiserFS працює не лише на GNU/Linux, але й на багатьох інших

платформах. ReiserFS – стандартна структура для організації роботи з файлами в ArchLinux, Slackware, SuSE, Xandros, Yoper, Linspire і Kurumin Linux.

Unix Filesystem2 – це файлова система, яка розроблена для сімейства операційних систем BSD (FreeBSD, OpenBSD, NetBSD). Також, підтримка UFS2 реалізована в ядрі Linux і операційній системі Solaris.

3.6. Ієрархія логічних файлових систем

В наступній таблиці наведено структуру каталогів операційної системи GNU/Linux.

Каталог	Призначення в ієрархії файлової системи
/	Кореневий каталог містить всю ієрархію файлової системи.
/bin	Містить найважливіші бінарні файли. Включає базові команди (ls, cp, cd), які можуть використовуватися всіма користувачами і які є необхідними для роботи системи.
/boot	Містить файли, необхідні для початкового завантажника GNU/Linux.
/dev	Містить файли системних пристроїв (dev від англ. DEVICES). Деякі файли, що знаходяться в /dev є обов'язковими, наприклад /dev/null, /dev/zero і /dev/tty.
/etc	Містить всі конфігураційні файли операційної системи.
/home	Містить всі файли та каталоги користувачів системи.
/lib	Містить бібліотеки, життєво необхідні для роботи операційної системи.
/mnt	Містить точки монтування для тимчасово монтованих файлових систем, таких як /mnt/cdrom, /mnt/floppy, /mnt/removable.
/opt	Містить не надто важливі для роботи операційної системи програмні пакети, наприклад Adobe Acrobat Reader.
/root	Домашній каталог суперкористувача root.
/sbin	Містить важливі системні бінарні файли, необхідні для запуску системи. Більшість цих файлів може запускати тільки root.
/tmp	Каталог призначений для збереження тимчасових файлів, що можуть створюватися окремими програмами.
/usr	Основний каталогом для збереження додатків.
/usr/bin	Містить значну кількість системних бінарних файлів. Будь-яка бінарна програма, яка не є необхідною для обслуговування системи і не призначена для системного адміністрування, повинна знаходитися в цьому каталозі.
/usr/lib	Містить всі бібліотеки, необхідні для запуску програм, що

	знаходяться в /usr/bin і /usr/sbin.
/usr/local	Місце, куди користувач може встановлювати програми, які компілюються ним з вихідних кодів.
usr/share	Містить всі апаратно-незалежні дані тільки в режимі читання, необхідні для додатків з /usr. Серед всього іншого в ньому можна знайти інформацію про часові пояси і регіональні стандарти.
/usr/share/doc	Містить документацію користувача.
/usr/share/man	Містить сторінки інтерактивної документації користувача.
/var	Місце для розміщення даних, які можуть змінюватися з часом, наприклад поштовий сервер, сервер друку та інше.

3.7. Монтування файлових систем

Так само як файл повинен бути відкритий перед використанням, так і файлова система, що зберігається на розділі диска, повинна бути змонтована, щоб стати доступною процесам системи.

Функція mount (монтувати) зв'язує файлову систему із заданого розділу на диску з існуючою ієрархією файлових систем, а функція umount (демонтувати) виключає файлову систему з ієрархії. Функція mount, таким чином, надає користувачам можливість звертатися до даних у дисковому розділі як до файлової системи, а не як до послідовності дискових блоків. Процедура монтування, відбувається наступним чином: користувач повідомляє операційній системі ім'я пристрою й місце у файловій структурі, куди потрібно приєднати файлову систему (крапка монтування).

Наприклад, в операційній системі GNU/Linux функція mount має наступний вигляд:

```
mount (special pathname, directory pathname, options);
```

де special pathname – ім'я спеціального файла пристрою, що відповідає дисковому розділу зі змонтованою файловою системою, directory pathname – каталог в існуючій ієрархії, куди буде монтуватися файлова система, а options вказує, чи варто монтувати файлову систему «тільки для читання». Далі операційна система повинна переконатися, що пристрій містить дійсну файлову систему із суперблоком, списком індексів і кореневим індексом.

3.8. Цілісність файлової системи

Важливий аспект надійної роботи файлової системи – контроль її цілісності. У результаті виконання файлових операцій, блоки диска можуть зчитуватись, модифікуватись і потім записуватись на диск. Причому багато файлових операцій зачіпають відразу кілька об'єктів файлової системи. Наприклад, копіювання файла припускає виділення йому блоків диска, формування індексного вузла, зміну вмісту каталога та інше. Протягом короткого періоду часу, між цими кроками, інформація у файловій системі виявляється неузгодженою. І якщо внаслідок непередбаченої зупинки системи на диску будуть збережені зміни тільки для частини цих об'єктів,

файлова система на диску може бути пошкоджена. У результаті можуть виникнути порушення логіки роботи з даними, наприклад з'явитися «загублені» блоки диска, які не належать жодному файлу й у той же час позначені як зайняті або навпаки, блоки позначені як вільні, але в той же час зайняті.

3.9. Журналювання файлів

Журнальовані файли в операційній системі GNU/Linux це звичайні текстові файли, більшість з них недоступні для простих користувачів, навіть для читання. Слідкувати за журнальними файлами – це завдання адміністратора системи. Основний конфігураційний файл служби syslog знаходиться у файлі /etc/syslog.conf. Його вміст складається з двох стовпців: у першому містяться шаблони в яких описано, які типи повідомлень і від яких програм буде обробляти syslog, у другому стовпці вказується дія яку буде виконувати syslog при отриманні повідомлення. Журналювання реалізовано в таких файлових системах як NTFS, Ext3fs, ReiserFS і деяких інших.

3.10. Поточний і домашній каталог

У GNU/Linux кожна виконувана програма працює в певному каталозі файлової системи. Такий каталог називається поточним каталогом. Можна сказати, що програма під час роботи перебуває саме в цьому каталозі, це їхнє робоче місце. Поточний каталог є у будь-якої програми, у тому числі й у командної оболонки (shell) користувача. Щоб дізнатись свій поточний каталог, потрібно використати команду: [user]\$ pwd (результат виконання – /home/user). Команда pwd (print working directory) повертає повний шлях поточного каталога командної оболонки.

У кожного користувача GNU/Linux обов'язково є власний каталог, який і стає поточним відразу після реєстрації в системі. Домашній каталог (home directory) – це каталог, призначений для зберігання власних даних користувача. Повний шлях до домашнього каталога зберігається в змінній оточення HOME. Користувач є власником цього каталога і всіх створених у ньому файлів. Як власник файлу, користувач може встановлювати права на файли. Ці права розподіляються між трьома категоріями користувачів:

- 1) власником файлу;
- 2) всіма користувачами, що є членами групи, асоційованою з файлом, але не є власниками;
- 3) та всіма іншими, куди входять всі інші користувачі, які не є ні власниками, ні членами групи власника.

Існують три різновиди прав доступу:

- 1) права на читання (Read, r);
- 2) права на запис (Write, w);
- 3) права на виконання (execute, x).

Команда chmod дозволяє змінювати права доступу – chmod a+r stuff, надає право всім користувачам читати файл stuff.

Оскільки кожний користувач має у своєму розпорядженні власний каталог, вирішується питання розмежування файлів між різними користувачами. Звичайно,

доступ інших користувачів до чужого домашнього каталога обмежений: найбільш типова ситуація, коли користувачі можуть читати вміст файлів один одного, але не мають права їх змінювати або видаляти.

3.11. Жорсткі і символічні посилання

Кожний файл це область даних на жорсткому диску комп'ютера або на іншому носії інформації. У файловій системі вміст файла зв'язується з його ім'ям за допомогою жорстких посилань. Створення файла за допомогою деякої програми, означає, що буде створено жорстке посилання і відкрита нова область даних на диску. Причому кількість посилань на ту саму область даних (файл) не обмежена, тобто у файла може бути кілька імен. Користувач GNU/Linux може надати файлу ще одне ім'я (створити жорстке посилання на файл) за допомогою команди `ln` (від англ. `link` – з'єднувати, зв'язувати). Перший параметр – це ім'я файла, на який потрібно створити посилання, другий – його повне ім'я.

У жорстких посилань є два істотних обмеження:

- Жорстке посилання може вказувати тільки на файл, але не на каталог, тому що інакше у файловій системі можуть виникнути цикли – нескінченні шляхи.
- Жорстке посилання не може вказувати на файл в іншій файловій системі. Наприклад, неможливо створити жорстке посилання на файл, який розташований на дискеті.

Щоб уникнути цих обмежень, були розроблені символічні посилання. Символьне посилання – це просто файл, у якому прописане ім'я іншого файла. Символьне посилання, як і жорсткі, надають можливість звертатись до файла, не зважаючи на його поточне ім'я. Крім того, символічні посилання можуть вказувати й на каталог, чого не дозволяють жорсткі посилання. Символьні посилання називаються так тому, що вони містять символи – шлях до файла або каталога.

КОНТРОЛЬНІ ПИТАННЯ

- 1) Перелічіть сучасні файлові системи.
- 2) Що таке ієрархічна файлова система.
- 3) Як класифікують файлові системи?
- 4) Що таке плоска файлова система?
- 5) Що таке кластерна файлова система?
- 6) Символьні посилання у GNU/Linux.
- 7) Що таке мережева файлова система?
- 8) Що таке розподілена файлова система?
- 9) Які є типи файлів в UNIX?
- 10) Яка структура файлової системи на диску?
- 11) Які є основні файлові системи у UNIX?
- 12) Перелічіть ключові файлові системи GNU/Linux.
- 13) Як відбувається монтування файлових систем?
- 14) Що таке поточний і домашній каталог?
- 15) Жорсткі посилання у GNU/Linux.

ЛЕКЦІЙНЕ ЗАНЯТТЯ №4

Тема: Завантаження операційної системи GNU/Linux. Робота з командним інтерпретатором.

Мета: Формування знань і вмінь, щодо роботи зі завантажником та командним інтерпретатором GNU/Linux.

ТЕОРЕТИЧНІ ВІДОМОСТІ

- 4.1. Етапи завантаження GNU/Linux
- 4.2. Початкове завантаження
- 4.3. Досистемне завантаження
- 4.4. Запуск системних служб
- 4.5. Файли, що впливають на процес завантаження
- 4.6. Вхід у систему
- 4.7. Популярні завантажувачі операційних систем
- 4.8. Утиліти початкового завантаження: LILO і GRUB
- 4.9. Користувацьке середовище GNU/Linux
- 4.10. Інтерфейс командного рядка
- 4.11. Командна оболонка
- 4.12. Командні оболонки операційної системи UNIX і GNU/Linux
- 4.13. Користувацькі команди

4.1. Етапи завантаження GNU/Linux

Завантажувач операційної системи:

- підготовлює апаратне забезпечення комп'ютера до старту ядра операційної системи;
- забезпечує засоби для ефективного діалогу між користувачем і комп'ютером (наприклад, завантажник дозволяє вибрати ядро операційної системи для завантаження);
- завантажує ядро операційної системи в оперативну пам'ять;
- формує параметри, які направляються до ядра операційної системи;
- керує процесом остаточного завантаження операційної системи.

4.2. Початкове завантаження

Під початковим завантаженням розуміється запуск операційної системи при ввімкненні живлення. Оскільки звичайні засоби операційної системи на даному етапі ще недоступні, система повинна в буквальному значенні «опрацювати сама себе». В ході цього процесу ядро операційної системи завантажується й активізується, потім виконується ряд ініціалізаційних завдань, після цього система готова до роботи.

Початкове завантаження – це період особливої вразливості операційної системи. Помилки в конфігураційних файлах, збої в роботі або відсутність потрібного устаткування, пошкодження файлової системи може перешкодити комп'ютеру почати звичну роботу. Налаштування режимів завантаження часто є одним з перших завдань, яке доводиться вирішувати адміністраторові в новій

системі. Типова процедура початкового завантаження операційної системи GNU/Linux складається із шести етапів:

- завантаження й ініціалізація ядра;
- виявлення й конфігурування пристроїв;
- створення процесів ядра;
- дії оператора (тільки при ручному завантаженні);
- виконання сценаріїв запуску;
- робота в багатокористувацькому режимі.

Коли вмикається живлення, запускається на виконання завантажувальний код, який зберігається в постійно запам'ятовуючому пристрої. Він повинен запустити ядро. Ядро опитує стан апаратних пристроїв, а потім запускає процес `init`, ідентифікатор якого завжди дорівнює 1. Перш ніж на екрані з'явиться реєстраційне запрошення, повинен відбутися цілий ряд подій. Файлові системи повинні бути перевірені й змонтовані, а системні процеси – запущені. Відповідні процедури реалізуються за допомогою сценаріїв інтерпретатора команд, які послідовно запускаються процесом `init`. Ці сценарії часто називають `rc`-файлами, оскільки вони мають префікс `rc` (розшифровується як `run command` – команда запуску). Традиційно, процес `init` визначає 7 рівнів виконання, на кожному з яких повинен виконуватися конкретний набір системних служб:

- Рівень 0 – система повністю припинила роботу.
- Рівень 1 або `S`, визначає (реалізує) однокористувацький режим.
- Рівні 2-5, визначає (реалізує) багатокористувацький режим.
- Рівень 6, визначає (реалізує) етап перезавантаження системи.

Рівні 0 і 6 характерні тим, що система не може в них залишатися, а переходить на них у процесі завершення роботи або перезавантаження. У багатокористувацькому режимі система найчастіше перебуває на рівні 2 або 3. Рівень 5 опрацьовується реєстраційними процесами `X Window`.

Ініціалізація ядра. Першим етапом початкового завантаження є читання записів з ядра Linux. Файл ядра розташований в каталозі `/boot`. В GNU/Linux, завантаження ядра здійснюється у два етапи. Спочатку з жорсткого диска зчитується невелика програма початкового завантаження, яка далі виконує завантаження ядра. Ядро виконує тести, які дозволяють визначити, скільки пам'яті є в розпорядженні системи. Частина внутрішніх структур ядра має фіксований розмір, тому ядро резервує певний обсяг фізичної пам'яті для самого себе. Ця пам'ять недоступна процесам користувача. Ядро видає на консоль повідомлення про загальний обсяг фізичної пам'яті, доступної користувачеві, а далі:

- задається ім'я комп'ютера;
- визначається часовий пояс;
- здійснюється перевірка дисків за допомогою команди `fsck`;
- монтуються файлові системи;
- видаляються старі файли з каталога `/tmp`;
- відбувається конфігурування мережевих інтерфейсів;
- відбувається запуск системних процесів і мережевих служб.

4.3. Досистемне завантаження

Програма login, яка реєструє користувачів у системі, запускається тільки тоді, коли сама система вже приведена в повну готовність і працює у штатному режимі. Саме завантаження – процес поступовий, поведінка комп'ютера на різних етапах завантаження визначається різними людьми – від розробників апаратної складової до системного адміністратора. Пропоновані до системи вимоги, можливості змінювати її залежно від апаратної складової, необхідність вирішувати різні завдання за допомогою того самого комп'ютера теж роблять процес завантаження поступовим: спочатку визначається профіль майбутньої системи, а потім цей профіль реалізується.

Початковий етап взагалі не залежить від того, яка операційна система встановлена на комп'ютері, для деяких етапів у кожній операційній системі пропонуються свої рішення. Цю стадію (початкову) назвемо досистемним завантаженням. Досистемне завантаження проходить у три етапи.

1. Завантажувач із постійно запам'ятовуючого пристрою визначає, з яких пристроїв можна вантажитись. Він завантажує з обраного пристрою первинний завантажувач і передає йому керування.
2. Первинний завантажувач визначає (а найчастіше – знає), де перебуває вторинний завантажувач – велика й досить інтелектуальна програма. Йому це зробити простіше, ніж програмі з постійного запам'ятовуючого пристрою: по-перше, тому що для кожного пристрою первинний завантажувач свій, а по-друге, тому що його можна легко змінювати.
3. Вторинний завантажувач знає, де перебуває ядро операційної системи (можливо, не одне) і пропонує користувачеві варіанти завантаження на вибір.

4.4. Запуск системних служб

Повноцінне завантаження Linux-системи – не тільки виконання програми login на віртуальній консолі (системі є чим зайнятися й крім ідентифікації користувачів). Потрібно виконати системні утиліти і програми, це ще називається запуском підсистем і служб. Зазвичай, системна служба організована так: під час початкового завантаження запускається у фоновому режимі програма, яка протягом роботи системи перебуває в таблиці процесів, однак здебільшого не діє, очікуючи, коли її про що-небудь попросять. Для того щоб попросити цю програму про послугу, яку вона надає, використовуються утиліти, які взаємодіють з нею згідно спеціального протоколу.

У ранніх версіях операційної системи UNIX все, що потрібно було запускати при старті системи, прописувалось в inittab. Було досить зручно в одному файлі вказувати, які саме процеси повинні працювати і в якому порядку їх запускати. Поведінка процесу при запуску розрахована на використання у файлі inittab методу «wait»: класичний процес запускається інтерактивно, перевіряючи правильність конфігураційних файлів та інші умови роботи. Далі, inittab чекає поки процес почне виконуватись і переходить до наступного рядка файла inittab.

4.5. Файли, що впливають на процес завантаження

Всі найважливіші загальносистемні конфігураційні файли розташовані в каталозі /etc і його підкаталогах. Ось короткий список цих файлів:

- /etc/lilo.conf (grub.conf) – файл, що визначає конфігурацію завантажувача lilo (grub);
- /etc/modules.conf (або /etc/conf.modules) – файл, що визначає конфігурацію модулів ядра;
- /etc/fstab – файл профілів, встановлює змінну \$PATH та інші найважливіші змінні;
- /etc/bashrc – глобальний файл конфігурації командної оболонки bash;
- /etc/issue – містить повідомлення, які виводяться на термінал перед входом у систему.

Перераховані вище конфігураційні файли впливають на процес завантаження системи й процес входу в систему будь-якого користувача.

Послідовність подій при повній реєстрації користувача в системі, виглядає так.

1. Користувач вводить реєстраційне ім'я, відповідно до запрошенням програми login процесу getty.
2. Процес getty виконує програму login, використовуючи як аргумент зазначене ім'я.
3. Програма login запитує пароль і звіряє ім'я й пароль із записаними у файлі /etc/passwd.
4. Програма login виводить на екран з файла /etc/issue «повідомлення дня».
5. Програма login запускає інтерпретатор shell, наприклад bash.
6. Інтерпретатор shell виконує відповідні файли запуску, після чого виводить на екран запрошення.

Про файли запуску треба сказати кілька слів додатково. У домашньому каталозі користувача перебувають кілька особистих файлів конфігурації. Якщо таких файлів у домашньому каталозі нема, то після входу в систему будуть прочитані глобальні файли, які містять значення «за замовчуванням».

4.6. Вхід у систему

Після виконання програми login, система видасть запит на введення логіна і паролю. Очевидно, що у відповідь потрібно вводити пароль того користувача, ім'я якого було введено раніше. При першому завантаженні системи необхідно ввести той пароль, який був заданий для користувача root у процесі інсталяції і натиснути Enter. Після введення пароля, командна оболонка виведе рядок запрошення:

```
[root@localhost /root]#
```

Поява запрошення означає, що система готова до роботи в текстовому режимі. Вид запрошення може різнитись залежно від виду командної оболонки.

4.7. Популярні завантажувачі операційних систем

- NTLDR – завантажувач ядра операційної системи Windows NT, 2000, XP;
- Windows Boot Manager (bootmgr.exe, winload.exe) – завантажувач ядра операційної системи Windows Vista;

- LILO (GNU/Linux LOader) – завантажувач ядра операційної системи GNU/Linux;
- GRUB (Grand Unified Bootloader) – завантажувач ядра операційної системи GNU/Linux;
- OS/2 BootManager – завантажувач ядра операційної системи OS/2;
- SILO (SPARC Improved bootLOader) – завантажувач ядра операційної системи Linux і Solaris;
- Loadlin – завантажувач ядра операційної системи GNU/Linux з під DOS або Windows;
- BOOTP – завантажувач ядра операційної системи по мережі;
- Yaboot – завантажувач ядра операційної системи GNU/Linux на PowerPC;
- BootX – завантажувач ядра операційної системи Mac OS X.

4.8. Утиліти початкового завантаження: LILO і GRUB

LILO – один із стандартних з завантажувачів ядра операційної системи GNU/Linux та FreeBSD. LILO не залежить від версії файлової системи. Підтримує від одного до шістнадцяти різних завантажувальних записів. LILO може бути встановлений в якості основної програми для завантаження в MBR або завантажувальний сектор. LILO широко використовується на практиці, однак, все частіше замінюється на завантажник GRUB. Як було сказано раніше, LILO не залежить від файлової системи і тому може завантажувати операційну систему з жорсткого диска або з дискети. Він підтримує два види завантаження – це завантаження Linux-ядра з підтримкою процесу *initrd* і передача прав запуску іншому завантажувачу операційної системи. LILO може бути прописаний як в головному завантажувальному секторі MBR так і в завантажувальному секторі розділу жорсткого диска.

GRUB – це завантажувач операційної системи GNU/Linux. GRUB є еталонною реалізацією завантажувача, відповідно до специфікації Multiboot, оскільки може завантажити будь-яку сумісну з ним операційну систему. Підтримує такі операційні системи як *BSD, Debian, Solaris, надає функціонал для передачі управління іншому завантажувачу, а це дозволяє запускати Windows, MS-DOS, OS/2. GRUB – найпопулярніший завантажувач у світі GNU/Linux. Основні можливості GRUB:

- завантаження GNU/Linux, OpenSolaris, *BSD інших Multiboot-сумісних операційних систем;
- захист паролем пунктів меню;
- підтримка BOOTP і TFTP для мережевого завантаження;
- інтерактивний командний рядок завантаження;
- підтримка файлових систем: NTFS, ISO, UFS, UFS2, FFS, FAT16, FAT32, Minix, ext2/ext3/ext4, ReiserFS, JFS і XFS.
- має модульну структуру.

4.9. Користувачьке середовище GNU/Linux

Командний рядок – це порожній рядок в якій прописуються команди з клавіатури. Немає значків, немає меню, немає кнопок. Насправді, командний рядок

не абсолютно порожній. Кілька символів, які називаються запрошенням системи, відображаються на початку рядка. Після цих символів мерехтить курсор. Хто не працювали в DOS або з командним рядком Windows, тому інтерфейс командного рядка може здатись зовсім дивним.

Мерехтливий текстовий курсор сигналізує, що командна оболонка очікує введення користувацьких команд. Оболонка – це комп'ютерна програма, яка дозволяє користувачу взаємодіяти з ядром операційної системи GNU/Linux, що виконує команди. Оболонка надає командний рядок у який вводяться команди, далі інтерпретує та обробляє їх і надсилає результат виконання ядру операційної системи. В GNU/Linux доступно кілька командних оболонок, за замовчуванням в багатьох дистрибутивах – це Bash shell (bash). Оболонка напряму обмінюється даними з ядром Linux, оскільки вся «комп'ютерна магія» зосереджена в ньому. Саме ядро надсилає дані на периферійні пристрої, записує файли на жорсткий диск, відправляє дані через мережу й виконує всі інші завдання нижнього рівня, здійснюючи таким чином роботу комп'ютера.

Багато користувачів GNU/Linux можуть виконувати всі необхідні завдання з робочого стола, не використовуючи командний рядок. Однак робота виконана безпосередньо в оболонці, майже завжди виконується швидше. Крім того, командний рядок надає більшу функціональність. Деякі завдання, які складні або неможливі для виконання з робочого стола в оболонці реалізуються досить просто. Навіть для паралельних функцій, команди мають набагато більше параметрів і надають більшу функціональність, а ніж об'єкти робочого стола. Системним адміністраторам і програмістам доволі часто доводиться працювати з командною оболонкою GNU/Linux.

4.10. Інтерфейс командного рядка

Будь-яку прикладну програму розробляють для широкого кола користувачів. Програма повинна мати прості та зручні засоби взаємодії з користувачем. Ці засоби називають – інтерфейсом користувача. Через інтерфейс, користувач керує роботою програми, отримує від неї повідомлення і відповідає на запити. Нині інтерфейс користувача має деякі стандартні засоби, основними елементами яких є меню та вікна діалогу. У графічній оболонці Windows та її додатках ці елементи доповнені, панеллю інструментів і смугами прокручування. Поняття меню інтерфейсу користувача за своєю суттю відповідає загальноприйнятому поняттю меню, наприклад, меню кафе. Меню користувача – це список деяких об'єктів з яких здійснюється конкретний вибір. Об'єкти меню називають пунктами меню або – командами.

Сучасні операційні системи забезпечують інтерфейс користувача у вигляді командного рядка або графічного інтерфейсу. Оболонка користувача, реалізується як окремий модуль, який надає можливість використовувати як текстовий так і графічний режими роботи операційної системи.

Алгоритм роботи інтерфейсного модуля досить простий:

- 1) більшу частину часу він очікує введення команди користувача;
- 2) інтерпретує команду, яка надійшла;

- 3) формує відповідний системний виклик або запускає визначену програму;
- 4) при потребі виводить на екран повідомлення про процес виконання;
- 5) після закінчення виконання команди переходить у режим очікування нової команди.

4.11. Командна оболонка

Говорити, що «користувач працює з операційною системою», фактично правильно бо взаємодію з користувачем організують спеціальні програми. Існує два види таких програм – оболонка (shell) для роботи в текстовому режимі (інтерфейс командного рядка) і графічний інтерфейс користувача GUI.

Деякі програми в GNU/Linux можуть бути запущені як через командну оболонку так і через графічний інтерфейс користувача. Запуск програм з оболонки еквівалентний подвійному клацанню мишкою по іконці програми в GUI. Передача аргументів програми в текстовому режимі аналогічна графічному. Однак, деякі програми не пристосовані для запуску через GUI і мають бути виконані, тільки з командного рядка.

В перших UNIX-системах, функції командного рядка, виконувала програма sh (скороченням від shell – оболонка). З часом було розроблено кілька її варіантів, зокрема, Bourne shell – розширена версія sh. Командна оболонка відповідає за роботу зі змінними оточення і виконує деякі перетворення (підстановки) аргументів. Головна ж властивість оболонки, яка робить її потужним інструментом користувача – це те, що вона включає в себе просту мову програмування.

4.12. Командні оболонки операційної системи UNIX і GNU/Linux

Інтерфейс командного рядка (Command line interface, CLI) – різновид текстового інтерфейсу між людиною і комп'ютером, в якому інструкції комп'ютеру даються шляхом введення з клавіатури команд. Він відомий також під назвою консоль. Командна оболонка – командний інтерпретатор, що використовується в операційних системах сімейства UNIX, в якому користувач може вводити команди окремо або запускати скрипти.

bash (*Bourne again shell*) – вдосконалена й модернізована варіація командної оболонки Bourne shell, розповсюджується безкоштовно по ліцензії GNU GPL. Одна з найпопулярніших різновидів командної оболонки UNIX. Особливо популярна в середовищі GNU/Linux, де в основному використовується як командна оболонка за замовчуванням. *bash* це командний процесор, що працює в інтерактивному режимі текстового вікна, може читати команди з файла, який називається скриптом (або сценарієм). Як і більшість оболонок, підтримує автодоповнення назв файлів і каталогів (за допомогою клавіші [Tab]), підстановку виведення результату команд, змінні, контроль за порядком виконання, оператори розгалуження і циклу. Ключові слова, синтаксис та інші особливості мови програмування були запозичені з sh.

Almquist shell (*ash*) – одна з самих «маленьких» командних оболонок, що працює в операційній системі UNIX і GNU/Linux.

Korn shell (*ksh*) – командна оболонка операційної системи UNIX. Повністю сумісний з Bourne shell і містить у собі можливості C shell.

Z shell – одна з сучасних командних оболонок операційної системи UNIX. Може використовуватись як інтерактивна оболонка або як потужний скриптовий інтерпретатор. Zsh є розширенням bourne shell з великою кількістю вдосконалень.

Наведемо, ще деякі командні оболонки, які використовуються в операційних системах сімейства UNIX.

- sh – оригінальна версія Bourne shell.
- csh – оболонка, яка використовується в операційних системах сімейства BSD.
- tcsh – покращена версія командної оболонки csh.
- ksh – «клон» облонки Bourne shell.
- Debian almqvist shell (dash) – полегшена командна оболонка.

4.13. Користувацькі команди

В наступній таблиці, подано ключові команди, що використовуються в оболонках операційної системи GNU/Linux.

Команда	Функціональна дія
startx	Запуск X Window.
arch	Назва апаратної платформи.
ar	Робота з архівами.
chown	Зміна власника.
at	Запуск команди в певний час.
login	Вхід у систему.
riogin	Вхід у віддалену систему.
chgrp, newgrp	Зміна користувацької групи.
date	Зміна й вивід поточної дати.
df	Обсяг вільного місця на жорсткому диску.
xwd, xv	Захоплення зображення на екрані.
hostname	Задання імені системи.
inetd	Ввімкнення Інтернет-сервера.
finger	Вивід інформації про користувача.
cd	Зміна поточного каталога.
mkdir	Створення каталога.
pwd	Вивід імені поточного каталога.
tail	Вивід кінця файлу.
chroot	Зміна кореневого каталога.
gcc	Компіляція C-програм.
g++	Компіляція C++-програм.
mount	Монтування файлової системи.

cat	Об'єднання файлів.
clear	Очищення екрана.
free	Обсяг вільної пам'яті на жорсткому диску.
xset	Встановлення параметрів X-сервера.
passwd	Зміна пароля.
env	Встановлення значень змінної оточення.
who	Інформація про користувача.
su, sudo	Підвищення прав до суперкористувача root.
mv	Переміщення файлів.
we	Підрахунок кількості слів у тексті.
find	Пошук файлів.
kill, killall, xkill	Завершення роботи процесу.
ps, top	Вивід списку процесів.
fdisk	Створення розділів на жорсткому диску.
emacs, vi, vim	Редагування текстового файла.
chmod	Зміна режиму доступу.
host	Вивід інформації про систему.
ls	Вивід списку файлів у каталозі.
sort	Сортування файлів.
rm	Видалення файла.
rmdir	Видалення каталога.

КОНТРОЛЬНІ ПИТАННЯ

- 1) Що таке завантажник операційної системи?
- 2) Перелічіть основні користувацькі команди GNU/Linux.
- 3) Чим визначається функціональність завантажника операційної системи?
- 4) Які завантажники використовуються в операційній системі GNU/Linux?
- 5) Що таке інтерфейс командного рядка?
- 6) Які завантажники є широко використовуваними?
- 7) Що таке LILO?
- 8) Чим характерне початкове завантаження?
- 9) Що таке GRUB?
- 10) Які етапи завантаження GNU/Linux?
- 11) Що таке досистемне завантаження?
- 12) Як відбувається запуск системних служб?
- 13) Які файли впливають на процес завантаження GNU/Linux?
- 14) Чим характеризується користувацьке середовище GNU/Linux?
- 15) Перелічіть командні оболонки операційної системи UNIX і GNU/Linux.

ЛЕКЦІЙНЕ ЗАНЯТТЯ №5

Тема: *Поняття користувача і групи у багатокористувацькій операційній системі.*

Мета: *Формування знань і вмінь, щодо роботи з користувачами і їхніми групами.*

ТЕОРЕТИЧНІ ВІДОМОСТІ

- 5.1. Права доступу у файловій системі
- 5.2. Ідентифікатор користувача і групи
- 5.3. Суперкористувач root і користувач nobody
- 5.4. Файл /etc/passwd
- 5.5. Файл /etc/group
- 5.6. Файл /etc/shadow
- 5.7. Системні реєстраційні імена

5.1. Права доступу у файловій системі

Працюючи з GNU/Linux, користувач помічає, що деякі файли й каталоги недоступні йому ні для читання, ні для запису, ні для використання. GNU/Linux є багатокористувацькою системою з розмежованими обліковими записами. Коли створюється обліковий запис – користувач автоматично зараховується до певної групи. Наприклад, в університеті може бути група студентів і група викладачів, з своїм рівнем доступу до певних ресурсів. Однак, користувач може бути членом кількох груп, наприклад викладач математики може бути членом групи викладачів і групи студентів, з якими він працює.

У багатокористувацьких операційних системах необхідно забезпечувати захист об'єктів (файлів, процесів), які належать одному користувачеві, від всіх інших. Операційна система UNIX пропонує базові засоби захисту й спільного використання файлів на основі відслідковування користувача й групи, що володіють файлом, трьох рівнів доступу (для користувача-власника, для користувачів групи-власника і для всіх інших користувачів) і трьох базових прав доступу до файлів (на читання, на запис і на виконання). Базові засоби захисту процесів засновані на відстеженні приналежності процесів користувачам.

Для відслідковування власників процесів і файлів використовуються числові ідентифікатори. Ідентифікатор користувача й групи – ціле число у діапазоні від 0 до 65535. Значення ідентифікатора користувача й групи – не просто числа, які ідентифікують користувача – вони визначають власників файлів і процесів. Серед користувачів системи виділяється один користувач – системний адміністратор або суперкористувач, який володіє всією повнотою прав на використання й конфігурування системи. Це користувач з ідентифікатором 0 і реєстраційним ім'ям root.

При поданні інформації людині зручніше використовувати замість відповідних ідентифікаторів символні імена – реєстраційне ім'я користувача й ім'я групи. Відповідність ідентифікаторів і символних імен, а також інша інформація

про користувачів і групи в операційній системі, зберігається у файлах `/etc/passwd`, `/etc/group` і `/etc/shadow`.

5.2. Ідентифікатор користувача і групи

Говорячи про права доступу користувача до файлів, варто зауважити, що в дійсності маніпулює файлами не сам користувач, а запущений їм процес (наприклад, утиліта `rm` або `cat`). В GNU/Linux політика доступу реалізується через властивості, а саме через ідентифікатор користувача – UID. Кожний процес системи, обов'язково належить якомусь користувачу і має відповідний ідентифікатор користувача (UID) – обов'язкова властивість будь-якого процесу в GNU/Linux. Коли програма `login` запускає стартовий командний інтерпретатор, вона приписує йому UID, отриманий у результаті діалогу з користувачем.

Як сказано раніше, користувач GNU/Linux може бути членом як однієї так і декількох груп. Історично так склалось, що одна з груп (GID) є для користувача основною. І коли говорять про GID-користувача, мають на увазі саме ідентифікатор групи. Інформація про відповідність імен груп їхнім ідентифікаторам, прописана у файлі `/etc/group`. Часто процедуру створення користувача проектують так, що ім'я групи за замовчуванням збігається з вхідним ім'ям користувача, а GID користувача – з його UID. Однак це зовсім не обов'язково: не завжди потрібно заводити для користувача окрему групу, а якщо заводити, то не завжди вдається зробити так, щоб бажаний ідентифікатор групи збігався з бажаним ідентифікатором користувача.

5.3. Суперкористувач root і користувач nobody

Суперкористувач або `root` – це спеціальний акаунт в UNIX-подібних операційних системах, власник якого має право на виконання всіх без винятку операцій. Така схема була заснована для полегшення процесу адміністрування операційної системи. Суперкористувач `root` може ставати іншим користувачем за допомогою операції `setuid`, але не навпаки. Командами `su` та `sudo` дозволяють (після введення вірного пароля) отримати права доступу суперкористувача `root`. Це робиться винятково з метою забезпечити максимальну інформаційну безпеку комп'ютерної системи.

В багатьох UNIX-подібних операційних системах, `nobody` (англ. `nobody` – ніхто) – це ім'я користувача, який не є власником жодного файла, який не є членом жодної привілейованої групи і не має ніяких повноважень, окрім стандартних для звичайних користувачів. Ідентифікатор користувача `nobody` в операційній системі – це число 100 (у суперкористувача `root` – 0).

5.4. Файл /etc/passwd

Кожний рядок у файлі `/etc/passwd` – це сім розділених двокрапкою полів. Приклад запису:

```
Dynamo:x:169:10:Student:/home/Dynamo:/bin/sh
```

Поля файла `/etc/passwd` та їхнє призначення.

Поле	Призначення
Ім'я користувача	Містить символічне ім'я користувача, використовуване при реєстрації в системі. Реєстраційне ім'я повинне складатися з алфавітно-цифрових символів, без пропусків, з максимальною довжиною, яка обумовлена операційною системою. Найбільш часто використовується довжина у вісім символів.
Пароль	Поле зберігає зашифрований пароль. Допускається порожнє поле. При використанні системи тінювого зберігання паролів у цьому полі перебуває тільки мітка пароля (x), а зашифрований пароль зберігається в іншому місці. Деякі системи враховують реєстр та передбачають використання як мінімум одного не алфавітно-цифрового символу.
Ідентифікатор користувача	Поле зберігає числовий ідентифікатор користувача, який пов'язаний з його реєстраційним ім'ям.
Ідентифікатор групи	Містить числовий ідентифікатор групи.
Коментар	Містить коментар – будь-який алфавітно-цифровий рядок.
Початковий каталог	Визначає початковий каталог користувача. Коли користувач розпочинає сеанс роботи, система направляє його саме в цей каталог.
Початкова команда оболонка	Визначає командне середовище користувача.

5.5. Файл /etc/group

Цей файл співвідносить числові ідентифікатори груп із символічними іменами. Кожний рядок файла /etc/group містить чотири поля. Приклад запису:

bin::2:root,bin

Поля файла /etc/group та їхнє призначення.

Поле	Призначення
Ім'я групи	Містить унікальне символічне ім'я групи.
Пароль групи	Алфавітно-цифровий пароль групи.
Ідентифікатор групи	Містить числовий ідентифікатор групи.

Список користувачів	Містить список реєстраційних імен користувачів даної групи. Імена в цьому списку розділяються комами. Користувачі можуть належати до декількох груп і за потреби, перемикаються між ними за допомогою команди newgrp.
---------------------	---

5.6. Файл /etc/shadow

Цей файл використовується в системах з тінювим зберіганням паролів, де вони винесені з доступного всім користувачам для читання файла /etc/passwd в цілях підвищення безпеки системи. Тут, крім власне зашифрованих паролів, зберігаються додаткові обмеження, пов'язані з реєстраційним ім'ям і паролем користувача. Доступ до цього файла має тільки суперкористувач root (команди passwd і login). Файл містить вісім полів, розділених двокрапкою, для кожного облікового запису в системі.

Поля файла /etc/shadow та їхнє призначення.

Номер поля	Призначення
1	Ім'я користувача.
2	Зашифрований пароль (алгоритм DES або MD5).
3	Кількість днів між 01.01.1970 (UNIX-день) і днем останньої зміни пароля.
4	Мінімальна кількість днів між змінами пароля.
5	Термін дії пароля користувача.
6	За скільки днів система буде розпочинати попереджати користувача про потребу змінити пароль.
7	Скільки днів користувач може не працювати в системі, перш ніж його реєстраційне ім'я буде заблоковано.
8	Дата, після якої профіль користувача буде заблокований.

5.7. Системні реєстраційні імена

Кожна версія операційної системи UNIX резервує кілька спеціальних реєстраційних імен для певних системних цілей. Так, в UNIX SVR4 системними вважаються реєстраційні імена, що відповідають ідентифікаторам від 0 до 100. Найбільше часто резервуються реєстраційні імена, представлені в наступній таблиці.

Реєстраційне ім'я	Призначення
root	Реєстраційне ім'я суперкористувача, адміністратора системи, має ідентифікатор 0. Єдине ім'я, обов'язково наявне в будь-якій UNIX-системі. Для виконання більшості програм адміністрування, використовується реєстраційне ім'я root, це забезпечує гарантований доступ до потрібних ресурсів.
daemon	Власник процесів, який реалізує служби користувача.
sys	Власник виконуваних системних команд UNIX.
bin	Власник стандартних утиліт UNIX.
adm	Псевдокористувач, який здійснює журналювання файлової системи.
cron	Псевдокористувач, від імені якого виконуються процеси запуску програм за розкладом.
nobody	Псевдокористувач, який використовується при роботі з мережевою файловою системою NFS.
uucp	Псевдокористувач підсистеми UUCP, передає поштові повідомлення і файли між UNIX-хостами.
lp, lpd	Псевдокористувач, від імені якого виконуються процеси друку.

КОНТРОЛЬНІ ПИТАННЯ

- 1) Охарактеризуйте поняття – інтерфейс користувача.
- 2) Поняття групи у GNU/Linux.
- 3) Охарактеризуйте поняття – інтерфейс командного рядка.
- 4) Що містить файл /etc/passwd?
- 5) Охарактеризуйте поняття – інтерфейс меню.
- 6) Що таке графічний інтерфейс користувача?
- 7) Які бувають стилі інтерфейсу користувача?
- 8) Що містить /etc/group?
- 9) Які є моделі інтерфейсу користувача?
- 10) Перелічіть інструментарій розробника інтерфейсів.
- 11) Поняття користувача у GNU/Linux.
- 12) Що таке ідентифікатор групи?
- 13) Що таке ідентифікатор користувача?
- 14) Чим особливий суперкористувач root?
- 15) Що містить /etc/shadow?

ЛЕКЦІЙНЕ ЗАНЯТТЯ №6

Тема: *Графічний інтерфейс операційної системи. Графічні оточення KDE і GNOME.*

Мета: *Формування знань і вмінь, щодо роботи з графічним оточення KDE і GNOME операційної системи.*

ТЕОРЕТИЧНІ ВІДОМОСТІ

- 6.1. *Графічний інтерфейс користувача (GUI)*
- 6.2. *X Window System*
- 6.3. *Проекти XFree86 і X.Org*
- 6.4. *Графічне середовище KDE*
- 6.5. *Елементи графічного інтерфейсу KDE*
- 6.6. *Графічне середовище GNOME*
- 6.7. *Елементи графічного інтерфейсу GNOME*
- 6.8. *Графічне середовище Xfce*
- 6.9. *Огляд мережевих пристроїв*

6.1. Графічний інтерфейс користувача (GUI)

Інтерфейс користувача – сукупність засобів для обробки та відображення інформації, максимально пристосованих для зручності користувача. У графічних системах інтерфейс користувача, реалізовується багатовіконним режимом, змінами кольору, розміру, видимості вікон, їхнім розташуванням, сортуванням елементів вікон, гнучкими налаштуваннями як самих вікон, так і окремих його елементів. Графічний інтерфейс користувача (GUI, Graphical User Interface) – тип інтерфейсу користувача, який дозволяє користувачам взаємодіяти з електронними пристроями через графічні зображення та візуальні вказівки. Ідея використання графічного інтерфейсу користувача зародилась в середині 70-х років XX століття, коли в дослідницькому центрі Xerox Palo Alto Research Center (PARC) була розроблена концепція візуального інтерфейсу. Передумовою появи графічного інтерфейсу стало зменшення часу реакції комп'ютера на введену команду.

Перша система з графічним інтерфейсом 8010 Star Information System групи PARC з'явилась за чотири місяці до виходу першого персонального комп'ютера корпорації IBM у 1981 році. На перших етапах візуальний інтерфейс використовувався тільки для прикладного програмного забезпечення. Зростання попиту на обчислювальну техніку змусило розробників операційних систем використовувати засоби візуального управління, спочатку на комп'ютерах Atari та Apple Macintosh, а потім і на IBM-сумісних комп'ютерах.

Паралельно з розробкою графічних інтерфейсів для операційної системи, відбувався процес уніфікації використання клавіатури і мишки при роботі з прикладними програмами. Злиття цих двох тенденцій привело до створення інтерфейсу користувача, за допомогою якого, при мінімальних затратах часу і зусиль на перенавчання, можна працювати з будь-якими програмними продуктами.

Основною концепцією сучасних графічних інтерфейсів є подання об'єктів для опрацювання за допомогою операційної системи (файл, каталог, програма) у вигляді візуальних графічних об'єктів, які мають певні властивості. Першу реалізацію об'єктно-орієнтованих концепцій інтерфейсу користувача, втілено в операційній системі IBM – OS/2.

Наступним розширенням ідеології об'єктно-орієнтованого інтерфейсу стала концепція робочого стола, як аналогія середовища, на якому розміщуються об'єкти:

- 1) документи – файли, каталоги з файлами;
- 2) програми – інструменти для роботи з цими документами.

Протягом тривалого часу desktop-ідеологія була складовою частиною різних інтерфейсів користувача, починаючи з Macintosh і закінчуючи Workplace Shell операційної системи OS/2. У сучасних операційних системах Windows, Mac OS, GNU/Linux робочий стіл користувача набув подальшого розвитку та вдосконалення. На сьогодні GUI є стандартною і непорушною складовою операційних систем.

6.2. X Window System

X Window System – це віконна система, яка забезпечує стандартні інструменти та протоколи для побудови графічних інтерфейсів користувача. X Window System – розроблений у Массачусетському технологічному інституті (США) у 1984 році. Сьогоднішня версія протоколу – X11, створена ще у вересні 1987 року. Проект X очолює фонд X.Org Foundation. X Window System часто називають X11 чи X.

Фактично, всі UNIX-подібні операційні системи підтримують X Window System, який забезпечує базові функції графічного середовища: відображення вікон на екрані, переміщення вікон на екрані, взаємодію з мишею і клавіатурою. X Window System – система, яка заснована на абстракціях і обробці потоку подій. Умовно можна виділити три класи подій – ті, які генеруються при зміні стану фізичних пристроїв і ті, які генеруються при зміні стану відображуваних вікон і події підтримки взаємодії між програмами-клієнтами.

X Window System використовує клієнт-серверну модель взаємодії. Сервер приймає запити на вивід графіки (вікон) і відправляє назад ввід користувача (від клавіатури, миші або сенсорного екрана). X-сервер обмінюється повідомленнями з різними клієнтськими програмами. Сервер приймає запити виведення вікон і відправляє назад введення користувача. X-сервер може бути:

- системною програмою, яка контролює виведення зображення на комп'ютері;
- додатком, який відображає графіку;
- виділеним компонентом апаратного забезпечення.

З часом візуальне подання X-інтерфейсів зазнало істотних змін, оскільки X Window System навмисно не визначає, як повинен виглядати інтерфейс користувача програми – ці питання вирішуються на рівні: віконних менеджерів, інструментаріїв елементів інтерфейсу, середовищ робочого стола, на рівні окремих додатків. Що стосується віконного менеджера, то він керує розміщенням і зовнішнім виглядом вікон додатків, може створювати інтерфейс, подібний Microsoft Windows або

Macintosh. Віконний менеджер може бути простим і мінімалістичним. Історично до появи X існувало кілька растрових дисплейних систем – Lisa (Apple, 1983), Macintosh (Apple, 1984), Blit (UNIX, 1984) та Andrew (UNIX, 1982). В наш час, найпопулярніші середовища робочого столу розроблені за використання X є GNOME та KDE.

6.3. Проекти XFree86 і X.Org

XFree86 – реалізація сервера X Window System, яка використовувалась в UNIX-подібних операційних системах та частково в Mac OS. Зараз практично не використовується, оскільки більшість розробників використовують X.Org Server. Проект XFree86 історично виник у 1992 році із сервера X386 для IBM PC-сумісних комп'ютерів. З часом XFree86 перетворився з просто окремо взятого порту X в провідну й найпопулярнішу реалізацію системи і став де-факто керувати розробкою X до середини 2004 року. Причиною занепаду XFree86 стала закрыта модель розробки, розробники не могли отримати доступ на запис в CVS-репозитарій і постачальникам доводилося підтримувати великі набори патчів. В лютому 2004 року проект XFree86 випустив версію 4.4 під ще більше обмеженою ліцензією. Багато проектів на X Window System таку ліцензію не сприйняли і не прийняли. Зокрема, Debian вказав, що додані пункти несумісні з умовами ліцензії GNU GPL. Інші групи (наприклад, *BSD) вважали, що накладення додаткових ліцензійних обмежень суперечить споконвічному духу вільного програмного забезпечення. Через ці ліцензійні проблеми з XFree86 склались передумови для розвитку і всебічної підтримки проекту X.Org.

У травні 1999 року The Open Group заснувала X.Org. X.Org Server (X.Org Foundation Open Source Public Implementation of X11) – вільна реалізація сервера X Window System з відкритим вихідним кодом. Випуски X.Org Server супроводжуються X.Org Foundation, який надає публічний доступ до стандартної реалізації X Window System для спільноти вільного програмного забезпечення. На сьогодні X.Org Server використовується практично у всіх варіантах GNU/Linux і *BSD.

6.4. Графічне середовище KDE

KDE (K Desktop Environment) – вільне графічне середовище і, одночасно, платформа для розробки, заснована на бібліотеці Qt, розповсюджується безкоштовно по ліцензії GNU GPL. Розробка KDE розпочалась у 1996 році студентом Тюбінгенського університету (Німеччина) – Матіасом Еттріхом. На даний час графічне середовище KDE підтримується командою програмістів зі всього світу. KDE складається з трьох базових частин:

- 1) Платформи.
- 2) Набору додаткових програм.
- 3) Робочого столу Plasma.

Крім того, KDE містить велику кількість програм та додатків, що змінюють відношення до операційних систем UNIX, GNU/Linux, *BSD. Інструментарієм розробки інтерфейсу користувача було вибрано бібліотеку Qt (кросплатформовий

інструментарій розробки програмного забезпечення мовою програмування C++), яка розповсюджується відповідно до умов ліцензування GNU GPL. KDE це також спільнота, що складається з програмістів, перекладачів, художників, авторів текстів та користувачів з усього світу. Метою міжнародної технічної команди KDE є створення найкращого програмного забезпечення для персональних комп'ютерів. Слід відмітити, що не існує єдиної групи компаній чи організацій, які б контролювали проєкт KDE, заохочуються будь-які внески до подальшого розвитку. Перший стабільний випуск KDE 1.0 датується 12 липня 1998 року, за період, що пройшов з цього дня, було презентовано понад п'ятдесят релізів.

6.5. Елементи графічного інтерфейсу KDE

Графічне середовище KDE, дотримується принципу сучасного робочого стола, тобто це робочий стіл на якому розміщені значки та панель задач, наявні віртуальні робочі столи і віджети. Значки на робочому столі є віджетами, файлами, каталогами, додатками, пристроями. Крім того, на столі розміщуються:

- Домашня папка. Надає доступ до всіх особистих файлів користувача.
- Головне меню. Надає доступ до програмного забезпечення, встановленого в системі (програми розміщено за відповідними категоріями).
- Системні значки. Надають доступ до системних налаштувань, мережевих ресурсів, тощо.
- Кошик. Надає доступ до всіх раніше видалених файлів.

6.6. Графічне середовище GNOME

GNOME (GNU Network Object Model Environment) – графічне середовище робочого стола для UNIX-подібних операційних систем, складається з вільних і відкритих кодів програм, розповсюджується безкоштовно по ліцензії GNU GPL.

GNOME був розроблений мексиканськими програмістами – Мігелем де Ікаса і Федеріко Мена у 1997 році. На сьогодні GNOME достатньо розвинений проєкт, який здобув масову популярність і використовується у багатьох дистрибутивах GNU/Linux. GNOME є частиною проєкту GNU. GTK+ є інструментарієм розробок GNOME. GTK+ розповсюджується на умовах General Public License, ліцензії на вільне програмне забезпечення, яка дозволяє GPL-несумісному програмному забезпеченню посилатися на нього. GNOME орієнтується на створення вільного графічного середовища. Для ефективної координації проєкту було засновано GNOME Foundation (відкриту організацію). GNOME Foundation допомагає спільноті розробників у фінансових, організаційних та правових питаннях, визначає бачення проєкту та його подальші плани. Основними цілями GNOME є:

- «свобода» для розробників робочого стола;
- вільне поширення робочого стола;
- нарощення підтримки локалізація;
- вільне використання GTK+;
- постійне оновлення релізів та випуск нових.

У підсумку, завдяки своїй «відкритості», GNOME широко використовується такими UNIX-подібними операційними системами як AIX, IRIX, *BSD, HP-UX, Solaris та інших. Робочий стіл GNOME є основним у таких дистрибутивах GNU/Linux як Debian, Fedora, RedHat, OpenSolaris, Linux Mint, FreeBSD та в ранніх і теперішніх версіях Ubuntu. Перший стабільний випуск GNOME 1.0 датується – 1 серпня 1997 року.

6.7. Елементи графічного інтерфейсу GNOME

Можливості графічного середовища GNOME багато в чому схожі з KDE. Інтерфейс користувача дещо відрізняється, але взагалі GNOME і KDE надають приблизно однакову функціональність.

Зазвичай на робочому столі GNOME розміщені наступні піктограми:

- «Домівка». Особистий каталог користувача системи.
- «DVD-привод». Відображає вміст диска, який вставлено до оптичного приводу.
- «Кошик». Зберігає всі файли, які видалені раніше.
- Системні утиліти, файли, каталоги.

Піктограми робочого оточення GNOME графічно репрезентують об'єкти і також є об'єктами. Вони мають вигляд підписаних мініатюрних картинок. Підпис – це назва піктограми. Назву піктограмі дає користувач. Картинку, назву та її розміри можна змінювати. В графічних оболонках дії з об'єктами зводяться до дій з їхніми піктограмами. Одним із видів піктограм є посилання (ярлик). Для кожного об'єкта можна створити декілька посилань і зручно їх розмістити у файлової системі.

Основним елементом графічного інтерфейсу GNOME є вікно, за допомогою якого, можна виконувати такі операції:

- відкрити вікно – запустити на виконання певну програму;
- закрити вікно – припинити виконання програми;
- розгорнути вікно на весь екран або згорнути до попереднього розміру;
- мінімізувати вікно, розгорнути мінімізоване вікно;
- змінити розміри вікна; перенести вікно в інше місце робочого столу; зробити вікно поточним.

6.8. Графічне середовище Xfce

Xfce – «легке» робоче середовище для UNIX-подібних операційних систем. Мета створення – швидкість і мала ресурсоемність, водночас, привабливість та простота у використанні. Xfce втілює традиційну філософію UNIX – модульність та повторне використання. Xfce складається з низки взаємозв'язаних компонентів, які за бажанням, можна використати в інших проектах. Серед таких компонентів:

- віконний менеджер;
- панель для запуску застосунків;
- дисплейний менеджер;
- менеджер керування сесіями користувача та контролю за енергоспоживанням;

- файловий менеджер – Thunar;
- веб-браузер – Midori;
- система налаштування параметрів оточення.

Іншим пріоритетом Xfce є відповідність стандартам, особливо, визначених freedesktop.org. Xfce встановлюється майже на всіх платформах UNIX: GNU/Linux, *BSD, Solaris, Cygwin, Mac OS X, PowerPC, Sparc.

6.9. Огляд мережевих пристроїв

Ядро Linux включає драйвери для різних типів мережевого устаткування. GNU/Linux має ряд стандартних імен для мережевих інтерфейсів, які описані нижче:

- 1) Локальний інтерфейс для loopback.
- 2) Ethernet-інтерфейс eth0, eth1 і т.д.
- 3) Token Ring -інтерфейс tr0, tr1 і т.д.
- 4) SLIP-інтерфейс sl0, sl1 і т.д.
- 5) PPP-інтерфейс ppp0, ppp1 і т.д.

КОНТРОЛЬНІ ПИТАННЯ

- 1) Що таке графічний інтерфейс користувача?
- 2) Що таке X Window System?
- 3) В чому важливість проекту X.Org?
- 4) Як класифікують графічний інтерфейсів користувача?
- 5) В чому важливість проекту XFree86?
- 6) Графічне середовище KDE.
- 7) Які переваги від використання графічного інтерфейсу користувача?
- 8) Які є елементи графічного інтерфейсу KDE?
- 9) Історія розвитку X Window System.
- 10) Графічне середовище GNOME.
- 11) Які недоліки від використання графічного інтерфейсу користувача?
- 12) Які є елементи графічного інтерфейсу GNOME?
- 13) Графічне середовище Xfce.
- 14) Що таке інтерфейс користувача.
- 15) Які основні елементи графічного інтерфейсу?

ЛЕКЦІЙНЕ ЗАНЯТТЯ №7

Тема: Прикладне програмне забезпечення для роботи у UNIX-подібних операційних системах.

Мета: Формування знань і вмінь, щодо роботи з прикладним програмним забезпеченням у UNIX-подібних операційних системах.

ТЕОРЕТИЧНІ ВІДОМОСТІ

- 7.1. *Поняття офісного пакету*
- 7.2. *Вільний офісний пакет OpenOffice.org (Apache OpenOffice)*
- 7.3. *OpenDocument Format і The Document Foundation*
- 7.4. *Текстовий редактор AbiWord і KOffice*
- 7.5. *Консольні текстові редактори*
- 7.6. *Екранно-орієнтований текстовий редактор vi*
- 7.7. *Веб-браузери*
- 7.8. *Графічні редактори*
- 7.9. *Файловий менеджер Dolphin і Nautilus*
- 7.10. *Файловий менеджер Midnight Commander (mc)*
- 7.11. *Програвачі мультимедіа*
- 7.12. *Програми для запису дисків і перегляду документів*
- 7.13. *Поштові клієнти*
- 7.14. *Емулятор Wine*
- 7.15. *Secure Shell – захищена командна оболонка*
- 7.16. *Система портів і пакетів у FreeBSD*

7.1. Поняття офісного пакету

Офісний пакет – сукупність програм з уніфікованим інтерфейсом, які дозволяють виконувати обробку інформації. Типовими компонентами, які входять до офісних пакетів є:

- текстовий процесор;
- редактор електронних таблиць;
- засіб створення презентацій;
- система управління базами даних.

Робоче місце користувача комп'ютера неможливо уявити без офісного пакета. Універсальне, цільове призначення засобів, що входять в його склад, обумовили широке розповсюдження цього програмного забезпечення серед всіх категорій користувачів.

Своє походження офісні пакети розпочинають від програм обробки текстів. Перші повноцінні офісні пакети, що містять текстовий процесор, редактор електронних таблиць і різні утиліти, з'явилися лише у 80-х роках ХХ століття. Піонером у сфері розробки офісних пакетів вважається компанія Lotus, яка в 1982 році випустила першу версію табличного редактора Lotus 1-2-3. У цьому ж році компанія Microsoft випустила першу версію текстового редактора Word для MS-

DOS. Через деякий час до набору офісних програм були додані програми для створення електронних таблиць, створення та проведення презентацій, планувальники-органайзери, а також персональні системи управління базами даних.

Основний етап еволюції офісних пакетів припав на початок 80-х років до середини 90-х років минулого століття. Подальші вдосконалення стосувалися лише покращення інтерфейсу користувача, зручності в роботі, інтеграції з Інтернет та інше. Сучасні офісні продукти пропонують досить багато різних засобів, які часто залишаються не затребуваними середньостатистичним користувачем. На сьогодні, основними офісними пакети є: Microsoft Office, LibreOffice, Gnome Office, KOffice, Apache OpenOffice, Ability Office, iWork, IBM Lotus, AbiWord.

7.2. Вільний офісний пакет OpenOffice.org (Apache OpenOffice)

OpenOffice.org (Apache OpenOffice) – повнофункціональний офісний пакет з багатомовною підтримкою, який працює в сучасних операційних системах, такими як Windows, GNU/Linux, *BSD, SunSolaris. Він є вільним офісним пакетом, який одночасно є і програмним продуктом та об'єднанням добровольців, що розробляють та підтримують його. Історично розробка OpenOffice.org розпочалась з комерційного продукту StarOffice (створеного на початку 90-х років XX століття компанією Star Division). У серпні 1999 року компанія Star Division була поглинута корпорацією Sun Microsystems, яка в свою чергу «перевела» StarOffice в розряд вільних проєктів. Після передачі у 2011 році кодової частини Openoffice.org від Oracle в управління фонду Apache офісний пакет був перейменований на Apache OpenOffice та переліцензований під Apache License.

Популярність відкритого офісного пакета Openoffice.org щороку зростає, хоча й зовсім не такими темпами, як хотілося б його розробникам. Openoffice.org забезпечує комфорт користувачам незалежно від середовища використання за рахунок однакового інтерфейсу. Дозволяє створювати і редагувати всі поширені формати документів, у тому числі, формати документів Microsoft Office. Власний формат документа Openoffice.org – Open Document Format (ODF), схвалений міжнародною організацією зі стандартизації ISO як стандарт ISO/IEC. На стороні Openoffice.org, в той чи інший час, були такі відомі корпорації Oracle, HP, Novell, IBM, які не тільки безпосередньо інвестували у проєкт, але й примушували Microsoft все більше піклуватися про підтримку стандартів.

До складу OpenOffice.org входять такі компоненти: Writer (текстовий процесор і редактор HTML); Calc (електронні таблиці); Draw (графічний редактор); Impress (система презентацій); Math (редактор формул); Base (база даних).

- Writer – інструмент із багатьма можливостями для створення листів, книг, звітів, інформаційних бюлетенів, брошур та інших документів, може експортувати файли в HTML, XHTML, XML, Adobe's Portable Document Format (PDF), Microsoft Word.
- Calc – програма для роботи з електронними таблицями, аналог Microsoft Excel, має засоби аналізу даних та побудови діаграм. Включає понад 300 функцій у тому числі для фінансових, статистичних і математичних

операцій. Надає можливість відкрити і працювати з робочими книгами Microsoft Excel і зберігати їх у форматі Excel.

- Impress – забезпечує всі загальні засоби представлення мультимедіа, такі як спеціальні ефекти, анімація і засоби малювання. Impress підтримує вихідні файли Microsoft PowerPoint.
- Draw – інструмент векторного малювання за допомогою якого можна виконувати завдання від створення простих діаграм або блок-схем до тривимірних художніх робіт.
- Base – система управління базами даних, яка надає всі інструментальні засоби для роботи з базами даних у межах простого інтерфейсу. Дозволяє створювати і редагувати форми, звіти, запити, таблиці, представлення, відношення.
- Math (редактор формул) – редактор формул і рівнянь.

7.3. OpenDocument Format i The Document Foundation

OpenDocument Format (ODF) – відкритий формат файлів для збереження й обміну офісними документами, доступними для редагування текстовими документами, електронними таблицями, малюнками, базами даних, презентаціями. Він є файловим форматом для збереження документів, заснований на XML. OpenDocument Format розроблений індустріальною спільнотою OASIS, яка займається розробкою і просуванням відкритих стандартів.

Стандарт, спільно та публічно, розроблений різними організаціями, доступний всім і може використовуватись без обмежень. Він є альтернативою закритому формату Microsoft Office Open XML з яким пов'язані різні ліцензійні обмеження. У жовтні 2011 року міжнародний консорціум OASIS затвердив фінальний варіант специфікації OpenDocument 1.2 як стандарт OASIS. ODF 1.2 відтепер офіційно став наступником випущеного в 2002 році стандарту ODF 1.0 і рекомендований для широкого впровадження. Специфікація OpenDocument підготовлена незалежним технічним комітетом, за участю експертів з таких корпорацій як Sun Microsystems, Oracle, IBM, Novell, Microsoft і Nokia. Крім комерційних компаній, у розвитку специфікації брали участь розробники з відкритих проєктів OpenOffice.org, LibreOffice, Gnumeric, KOffice, AbiWord, WebODF, IpOD і ODF Toolkit. Всього на підготовку специфікації OpenDocument було витрачено майже п'ять років, сама ж специфікація складається з трьох частин: загальна схема ODF, OpenFormula (формули для електронних таблиць), модель пакування даних в ODF-контейнер.

Document Foundation – некомерційна організація, яка займається розробкою і підтримкою програмного забезпечення з відкритим вихідним кодом. Фонд організований колишніми учасниками проєкту OpenOffice.org для розробки та підтримки LibreOffice. Основною метою фонду є розробка незалежного офісного проєкту з підтримкою відкритого формату ODF. Фонд ставить перед собою завдання – створити в повній мірі вільний офісний пакет як для користувачів, так і для розробників, не примушуючи їх передавати авторські права будь-якій

організації, в тому числі самому фонду. The Document Foundation доступна приватним особам та організаціям, які погоджуються з її основними цінностями і беруть участь в її діяльності.

7.4. Текстовий редактор AbiWord і KOffice

AbiWord – це вільний текстовий редактор, що розповсюджується за ліцензією GNU General Public License. Розробка базується на бібліотеці GTK+. Функціонує на платформах GNU/Linux, Mac OS X, Microsoft Windows, ReactOS, SkyOS, BeOS та інших. Назва AbiWord походить від кореня іспанського слова «Abierto», що значить – відкритий. AbiWord створений компанією SourceGear як перша складова AbiSuite – амбіційного проекту створення повного офісного пакета, який містить тільки відкрите й безкоштовне програмне забезпечення. Після того, як пріоритетним напрямком розвитку SourceGear стало фінансове збагачення, AbiWord став розроблятися цілком і повністю командою добровольців. Нині AbiWord є частиною GNOME Office.

В цілому, AbiWord поступається у функціональності Microsoft Word і Apache OpenOffice Writer, однак перевершує прості текстові редактори, такі як WordPad для Windows або gedit для UNIX. Набір команд редактора може бути частково розширений шляхом підключення плагінів, які доступні на їхньому офіційному сайті.

KOffice – це вільний і відкритий офісний пакет, який працює на базі графічної оболонки KDE. Всі його компоненти опубліковані під вільними і відкритими ліцензіями. KOffice використовує формат OpenDocument. До його складу входять такі програмні складові:

- Words – текстовий процесор.
- Tables – табличний процесор.
- Stage – система для підготовки презентацій.
- Flow – система для побудови діаграм, блок-схем і структур мереж (раніше Kivio).
- Kexi – середовище для управління базами даних.
- Krita – растровий графічний редактор.
- Karbon – векторний графічний редактор.

7.5. Консольні текстові редактори

Текстовий редактор – комп'ютерна програма, який дозволяє виконувати операції набору, редагування та оформлення тексту.

Emacs. Кросплатформний редактор, що широко використовується в UNIX-подібних операційних системах. Один з найбільш популярних серед користувачів UNIX. Emacs може вирішувати широке коло завдань – від написання текстів і компіляції програм, до роботи в Інтернеті. Розповсюджується як у варіанті для текстового режиму, так і у варіанті для графічної оболонки.

ed. Перший стандартний редактор операційної системи UNIX. Розширена його версія, відома як *ex*, стала основою редактора *vi*. Має командно-орієнтований інтерфейс.

bvi. Редактор двійкових файлів, представлених у вигляді шістнадцяткових чисел. При створенні редактора за основу було взято текстовий редактор *vi*.

Elvis. Інтерактивний, повноекранний редактор, сумісний з операційною системою UNIX. Один з перших клонів *vi*. Основними можливостями є підсвічування синтаксису і редагування кількох файлів в різних вікнах.

Nano. Консольний текстовий редактор для UNIX-подібних операційних систем, який базується на бібліотеці *curses*. Був створений, щоб повторити функціональність і зручність інтерфейсу іншого редактора – *Pico*.

7.6. Екранно-орієнтований текстовий редактор vi

vi (*vedit*) – екранно-орієнтований текстовий редактор, який дозволяє відображати цілу сторінку тексту, переміщатися по ньому курсором і безпосередньо бачити внесені зміни. Розповсюджується згідно BSD License. Перша версія була написана Біллом Джоєм у 1976 році. У той час найпоширенішим текстовий редактор був редактор *ed*. На відміну від багатьох звичних текстових редакторів, *vi* має модальний інтерфейс. Це означає, що ті самі клавіші в різних режимах роботи виконують різні дії. У редакторі *vi* є два основних режими: командний режим і режим вставки. За замовчуванням, робота починається в командному режимі.

7.7. Веб-браузери

Mozilla Firefox. Веб-браузер – це програмне забезпечення для комп'ютера або іншого електронного пристрою, яке надає можливість користувачеві взаємодіяти з текстом, малюнками або іншою інформацією на гіпертекстовій веб-сторінці. *Mozilla Firefox* – вільний, безкоштовний згідно ліцензії GPL, веб-браузер з відкритим вихідним кодом, заснований на ядрі *Gecko*. Найперше потрібно відмітити, що програмний код *Mozilla Firefox* був створений з «нуля» в *Mozilla Organization* замість коду *Netscape Communicator 5*. Відразу після виходу, 9 листопада 2004 року, *Mozilla Firefox 1.0* отримав визнання в таких виданнях, як «*Форбс*» і «*The Wall Street Journal*».

Mozilla Firefox швидкий, функціональний, технологічно постійно розширюваний за можливостями браузер. *Firefox* підтримує вкладки, що дозволяє відкрити в одному вікні декілька сторінок. Надає можливість завантажувати посилання у фоновому режимі. Підтримує безкоштовне оновлення через Інтернет. *Mozilla Firefox* підтримує:

- web-стандарти;
- має зручний інтерфейс;
- вбудований захист від спливаючих вікон;
- інтегровану панель пошуку;
- зручний менеджер *cookie* і паролів.

Firefox працює на *Windows*, *GNU/Linux*, **BSD*, *OS/2*, *Mac OS*.

Особливості Firefox: невеликий розмір інсталяційного пакету, підтримка великої кількості розширень, автоматичне оновлення, широкі можливості з налаштування зовнішнього вигляду і загальної роботи програми, наявність інтегрованих в закладки RSS потоків, наявність вбудованої панелі пошуку та веб-редактора сторінок. Переваги Firefox над іншими браузерами: блокування спливаючих вікон, вбудовані інструменти для веб-розробника, безпечно зберігання паролів.

Google Chrome – це безкоштовний веб-переглядач, розроблений компанією Google на основі веб-переглядача з відкритим кодом Chromium та іншого відкритого програмного забезпечення. Його програмна архітектура була створена з чистого аркуша з метою задоволення поточних потреб користувачів. Нова архітектура насамперед визнає той факт, що на сьогодні більшість веб-сайтів є не просто веб-сторінками, але веб-програмами. Заявленими перевагами цієї архітектури є підвищена стабільність, швидкість, безпека, а також простий та зручний інтерфейс користувача.

За даними статистики сайту StatCounter, браузер Chrome є абсолютним лідером за популярністю у світі, а його ринкова частка, серед найпопулярніших веб-браузерів на всіх платформах у лютому 2019 склала 62,41%.

Opera – це веб-переглядач розроблений норвезькою компанією Opera Software. Браузер написаний мовою програмування C++, має високу швидкість роботи і сумісний з основними веб-технологіями. Характерними особливостями Opera довгий час були багатосторінковий інтерфейс (система вкладок у вікні програми) і можливість масштабування документів цілком, разом із графікою; згодом ці функції з'явилися і в інших браузерах. У Opera розширені функціональні можливості використання комп'ютерної миші: окрім стандартних способів навігації передбачені так звані «жести мишею». У систему безпеки браузера входять:

- вбудований захист від фішингу;
- додаткове кодування інформаційного потоку при роботі зі сторінками, які містять запити конфіденційних відомостей;
- можливість видалення cookies.

Браузер Opera працює на операційних систем – Windows, Mac OS, GNU/Linux, *BSD, Solaris, а також використовуються для мобільних платформ на основі Symbian, MeeGo, MeeGo WeTab, Java, Android, Windows Mobile, Apple iOS.

Eirphany – офіційний веб-браузер графічного середовища GNOME. Eirphany заснований на бібліотеці Gecko. Eirphany позиціонується як простий у використанні браузер, позбавлений зайвої функціональності, а саме поштового клієнта, редактора веб-сторінок і тому подібного. Проект Eirphany започаткований у 2002 році Марко Песенті Грітті, який відокремився від групи розробників браузера Galeon з огляду на розбіжності в баченні майбутнього проекту. Eirphany підтримує так звані «розумні» закладки: закладці може передаватися як аргумент деякий текстовий рядок.

7.8. Графічні редактори

Растровий графічний редактор GIMP. Растрові графічні редактори потрібні для створення, зміни, оптимізації та збереження різноманітних типів зображень. Такі програми можуть використовувати для професійної діяльності: художники, фотографи, ілюстратори, веб-майстри, а також всі інші бажаючі, що хочуть покращити або відредагувати фото, малюнок, або інше графічне зображення.

Основними властивостями даних редакторів є створення малюнка за допомогою різних інструментів:

- олівця, пензля, розпилювача, інструментів малювання ліній, прямокутників, еліпсів тощо;
- використання при малюванні різних кольорів;
- робота над фрагментами малюнка (виділення окремих елементів малюнка, їх переміщення, копіювання, обертання тощо);
- масштабування малюнка для перегляду всього зображення або малювання дрібних деталей;
- імпортування до малюнка об'єктів з інших малюнків;
- додавання текстових написів до малюнків; виконання операцій над файлами: відкриття, зберігання, друкування, тощо.

GIMP (The GNU Image Manipulation Program) – вільний растровий графічний редактор із деякою підтримкою векторної графіки, розповсюджується безкоштовно згідно ліцензії GNU GPL. GIMP – це професійний засіб для роботи із зображеннями растрової та частково векторної графіки. Одним з найвідоміших прикладних пакетів, розроблених в рамках проекту GNU. Вільно поширюваний на умовах ліцензії GPL. Назва GIMP є скороченням від GNU Image Manipulation Program. Завдання GIMP – розробка універсального програмного продукту для всіх сучасних платформ: GNU/Linux, *BSD, MAC OS, Microsoft Windows.

Проект було започатковано у 1995 році Спенсером Кімбеллом і Пітером Маттісом як навчальний проект в університеті Берклі (США). У 1997 році GIMP став частиною проекту GNU. GIMP є одним з перших додатків подібного класу, який поширюється з відкритими вихідними кодами. Незважаючи на те, що GIMP є вільним програмним забезпеченням, його можливості практично не поступаються аналогічному продукту Photoshop від Adobe. Останнім часом інтерес до GIMP значно зріс за межами GNU/Linux-співтовариства у зв'язку з появою версії для Windows.

Редактор векторної графіки Inkscape. Inkscape – вільний редактор векторної графіки з можливостями, подібними до можливостей Illustrator, Freehand, CorelDraw. Програма ліцензована за умовами GNU GPL. Inkscape створено на основі набору бібліотек для побудови графічного інтерфейсу GTK+ та додатка для створення програм мовою C++. Рідним форматом запису програми Inkscape є SVG, підтримує читання і запис зображення у форматах SVG, OpenDocument Drawing, DXF, PNG, WMF, EMF, sk1, PDF, EPS, PostScript. Inkscape може використовувати згладжування, регулювання прозорості, створення градієнтів заповнення, має вбудований набір готових шаблонів, може працювати з векторними шрифтами.

Перевагою програми є широкий набір ефектів, які можна застосовувати для створення різноманітного роду зображень художнього та дизайнерського призначення.

7.9. Файловий менеджер Dolphin і Nautilus

Файловий менеджер (file manager) – комп'ютерна програма, яка надає інтерфейс користувача для роботи з файловою системою та файлами. Файлові менеджери призначені для виконання таких операцій над об'єктами як: створення, перегляд, відкриття, редагування, переміщення, перейменування, видалення, вилучення, зміна атрибутів, пошук, призначення прав доступу.

Функціями сучасних файлових менеджерів є: підтримка клавіатурних скорочень; підтримка функції Drag&Drop; робота з архівами; підтримка архівів ZIP, GZ, TAR, TGZ, RAR, ARJ, LZH; робота з командним рядком; можливості розширеного пошуку файлів (пошук файлів за датою, властивостями, розміром, шаблонами тощо); вбудований переглядач файлів; головне меню, яке можна налаштувати; робота з FTP-клієнтом.

Для спрощення роботи з файлами в операційній системі розроблено так звані файлові менеджери. У графічному середовищі KDE таким менеджером є програма Dolphin. Її можна запустити, якщо двічі клацнути на робочому столі, наприклад, на піктограмі каталогу «Домівка».

Під час запуску файлового менеджера відкриється вікно, поділене на дві частини. Зовнішній вигляд менеджера Dolphin можна змінювати, тобто налаштувати його відповідно до своїх потреб. Стандартно у лівій частині є дерево каталогів, а у правій – робоче поле активного каталога. Якщо у лівому полі деякий каталог містить підкаталоги, то біля його назви буде значок «+». У випадку, якщо каталог згорнутий або закритий наявний значок «-». У Dolphin передбачено перегляд декількох каталогів в окремих вкладках вікна програми. Нову вкладку можна створити за допомогою пункту меню «Файл» → «Нова вкладка» або комбінацією клавіш Ctrl+T. Dolphin є реалізацією нової концепції використання панелі адреси для пришвидшення та спрощення перегляду: у програмі використано панель послідовної навігації. Замість показу адреси повністю, на панелі адрес відображено лише основні пункти переходів.

У Dolphin передбачено підтримку трьох різних режимів перегляду: «Піктограми», «Подробиці» і «Компактний». Змінити режим перегляду можна за допомогою меню «Керування» → «Режим перегляду» або «Перегляд» → «Режим перегляду» (якщо ввімкнено відображення смужки меню) або за допомогою комбінацій клавіш: Ctrl+1 («Піктограми»), Ctrl+2 («Компактний») і Ctrl+3 («Подробиці»). Типово, файли у поточному каталозі будуть впорядковані за назвою. Спосіб впорядкування можна змінити за допомогою пункту меню «Керування» → «Впорядкувати за» (критерії впорядкування: назва, розмір, дата, права доступу, власник, група, тип, призначення посилання, шлях).

У графічному середовищі GNOME файловим менеджером за замовчуванням є Nautilus. Він розповсюджується за ліцензією GPL і є вільним програмним

забезпеченням. Після запуску Nautilus відкривається вікно, поділене на дві складові – схоже на «Провідник» у Windows. Перша стабільна версія Nautilus 1.0 вийшла 13 березня 2001 року. Далі в історичній хронології, здійснено наступні удосконалення: перехід на GTK+, підтримка 3D, вдосконалення системи пошуку, перехід на GVFS (віртуальну файлову систему), оновлення движка скриптів. Слід відмітити, що за допомогою GVFS у Nautilus можна переглядати FTP-сайти, Windows SMB-ресурси, файлові системи мобільних телефонів. Nautilus підтримує перегляд «в іконках» текстових файлів, зображень, звукових та відеофайлів. Nautilus зберігає історію папок, на зразок сучасних веб-браузерів, надає простий доступ до папок та файлів.

7.10. Файловий менеджер Midnight Commander (mc)

Midnight Commander (mc) – консольний файловий менеджер для GNU/Linux і UNIX-подібних операційних систем, клон Norton Commander, розповсюджується безкоштовно згідно ліцензії GNU GPL. Midnight Commander призначена для виконання дій над об'єктами, а вікно програми містить: заголовок, робоче поле, командний рядок і рядок функціональних клавіш.

Робоче поле поділене на два вікна. Одне вікно є активним, інше – пасивним. Перехід між вікнами відбувається за допомогою клавіші [Tab]. Зовнішній вигляд вікон та способи відображення інформації в них можна налаштовувати, користуючись відповідними командами меню: «Ліва», «Права». Щоб зробити меню активним, потрібно натиснути клавішу F9. Курсор завжди міститься в активному вікні. В активному вікні є виділена кольором рамка, яка позначає вибраний об'єкт. Об'єкт обирається за допомогою клавіатури чи клацанням по ньому мишею.

За допомогою Midnight Commander особливо зручно: візуально знаходити каталоги та файли; упорядковувати назви файлів і каталогів; переглядати файли; редагувати текстові документи; копіювати файли та каталоги; переміщати чи перейменовувати об'єкти; створювати каталоги; видаляти об'єкти; архівувати, розархівувати об'єкти; змінювати права доступу до окремих об'єктів; створювати посилання; запускати програми на виконання; налагоджувати FTP-з'єднання.

Оболонка Midnight Commander дуже схожа на оболонки Norton Commander, Volcov Commander, Dos Navigator, Far, Total Commander. У командному меню Midnight Commander:

- F1 – виклик довідки.
- F2 – виклик меню користувача.
- F3 – перегляд файла.
- F4 – редагування файла.
- F5 – копіювання відмічених файлів і каталогів.
- F6 – переміщення або перейменування відмічених файлів і каталогів.
- F7 – створення нового каталога на активній панелі.
- F8 – видалення відмічених файлів і каталогів.
- F9 – перехід у головне меню.
- F10 – вихід з Midnight Commander.

Переміщення по дереву файлів і каталогів здійснюються за допомогою клавіш керування курсором або клацанням лівою клавішею мишки у потрібному рядку. Для входу у каталог потрібно вивести підсвічений рядок на каталог і натиснути клавішу [Enter] (або двічі клацнути лівою клавішею мишки). Для виконання дій з файлом його потрібно вибрати і натиснути відповідну функціональну клавішу. Окрім дій з окремими файлами або каталогами Midnight Commander може виконувати копіювання, переміщення, видалення групи файлів або каталогів.

Меню Midnight Commander містить такі групи команд: «Ліва», «Файл», «Команди», «Параметри», «Права». У групі «Ліва» («Права») містяться команди налаштування лівої (правої) панелі – способи відображення об'єктів, їхнього сортування, налагодження мережевого та FTP-з'єднання тощо. У групі «Файл» повторено команди функціонального рядка. Крім того, за його допомогою можна змінити права доступу до об'єктів, занести об'єкти у групу об'єктів або інвертувати вибір. За допомогою пункту «Команди» можна відобразити дерево каталогів, поміняти або вимкнути панелі, переглянути список поданих раніше команд (історію), відновити знищені файли, змінити розширення файла. Команди з групи «Параметри» призначені для налаштування функціонування менеджера, зберігання внесених змін у конфігурації.

7.11. Програвачі мультимедіа

Amarok – програма для програвання аудіофайлів для операційних систем GNU/Linux і UNIX. Для своєї роботи використовує KDE, однак офіційно не входить у жоден з компонентів KDE і випускається незалежно від нього. Проект спочатку був розпочатий Марком Кретшманом, якого не задовольняв аудіоплеєр XMMS. Пізніше до проекту підключились інші програмісти. З першого релізу плеєр називався amaroK з великою літерою «K» в кінці, яку так полюбляють розробники KDE. У червні 2006 року він був перейменований в Amarok після інтенсивної дискусії на конференції KDE – K3M. З версії 2.0 був повністю перероблений інтерфейс.

Kodi (до 2014 року – XBMC) – це кросплатформне відкрите програмне забезпечення для організації медіацентру. Ще до 2002 року проект XBMC був націлений на створення відкритого мультимедійного плеєра для гральної приставки XBOX, але в процесі розвитку трансформувався в медіацентр, що працює на сучасних програмних платформах. Готові інсталяційні пакети доступні для Mac OS X, FreeBSD, GNU/Linux, Android, Raspberry Pi, Windows, Apple TV і Apple iOS. З цікавих функцій Kodi можна відзначити:

- підтримка широкого спектру форматів мультимедіа і засобів апаратного прискорення декодування відео;
- підтримка пультів дистанційного керування;
- можливість відтворення мультимедіа по FTP, SSH і WebDAV;
- можливість віддаленого управління через веб-інтерфейс;
- наявність гнучкої системи плагінів;
- підготовка плагінів для інтеграції з популярними online-сервісами.

Banshee – музичний програвач з відкритими вихідними кодами для операційної системи GNU/Linux. Banshee розроблений з використанням Mono і GTK#. Він використовує мультимедійну платформу GStreamer для кодування й декодування різних музичних форматів, таких як mp3, ogg, flac. Banshee може програвати, імпортувати й синхронізувати музику з різних портативних пристроїв. Функціональність програвача можна розширити, використовуючи плагіни.

Kaffeine – це медіапрогравач для UNIX-подібних операційних систем з середовищем KDE. Зазвичай Kaffeine використовує «движок» програвача xine, але підтримується і GStreamer. Крім цього, він надає змогу використовувати бінарні кодеки для різних форматів. У рамках проекту створено плагін для веб-браузера Mozilla. До основних можливостей програвача належать відтворення DVB, DVD, Video CD і CD audio.

7.12. Програми для запису дисків і перегляду документів

Brasero – вільна програма для запису CD і DVD для UNIX-подібних операційних систем, є графічним інтерфейсом для cdrtools, dvd+rw-tools та бібліотеки «libburn». Написана на мові C з використанням бібліотеки GTK+. Офіційно включена в графічне середовище GNOME.

K3b – програма для запису компакт-дисків для UNIX-подібних операційних систем. Для своєї роботи використовує KDE (kdelibs), проте офіційно не входить до жодного з компонентів KDE і випускається окремо від нього. На офіційному сайті вказано що вона «оптимізована для KDE». Тісно інтегрована з аудіоплеєром Amarok і дозволяє записувати прямо з нього Audio DVD.

Okular – універсальний застосунок для перегляду документів з набору програм KDE. Основою коду програми є код KPDF. З часу появи KDE 4 програма увібрала до себе код багатьох програм для перегляду документів, щоб домогтися універсальності при перегляді даних у різних форматах. Існують версії для роботи з операційними системами: GNU/Linux, Windows, Mac OS, *BSD. Підтримує систему плагінів, яка дозволяє підтримувати інші формати документів.

7.13. Поштові клієнти

Mozilla Thunderbird. Електронна пошта – це сервіс Інтернет, який забезпечує можливість пересилання повідомлень між двома або більше абонентами. Mozilla Thunderbird – безплатний клієнт електронної пошти, новин, та RSS з відкритим вихідним кодом. Є частиною проекту Mozilla. Інтерфейс Thunderbird, як і веб-браузера Mozilla Firefox, заснований на технології XUL, розробленої Mozilla Foundation. У результаті інтерфейс користувача на всіх платформах виглядає однаково. Thunderbird підтримує візуальні теми. Типово інтерфейс програми схожий на звичний користувачам Windows інтерфейс поштового клієнта Outlook Express. Mozilla Thunderbird поштовий клієнт, в якому можна відображати всі облікові записи як в одній групі папок, так і в окремих групах.

Mozilla Thunderbird підтримує:

- протоколи SMTP, POP3, IMAP, NNTP, RSS;

- кросплатформність (працює на Windows, GNU/Linux, *BSD, Mac OS);
- теми та плагіни;
- фільтрацію спаму;
- віртуальні директорії.

Evolution – графічна програма управління електронною поштою з відкритим вихідним кодом, спочатку написана для платформи GNU/Linux. Нині є збірки для Mac OS і Windows. Створена компанією Ximian, придбаною у 2003 році корпорацією Novell, яка з тих пір здійснює розробку та підтримку продукту. З вересня 2004 року входить до складу графічного середовища GNOME. Включає календар, систему планування часом, адресну книгу. Підтримує всі поширені поштові протоколи – IMAP, POP, SMTP з аутентифікацією через TLS.

KMail – це клієнт електронної пошти, створений в рамках проекту KDE. Поширений в операційних системах сімейства UNIX. Підтримує SMTP, POP3, IMAP, локальні поштові скриньки. Підтримує відображення листів в текстовому форматі і в HTML. Широко представлені можливості безпеки: підтримка з'єднань із захистом SSL, TLS, аутентифікації MD5, включено антивірусне програмне забезпечення.

7.14. Емулятор Wine

Wine – це вільна реалізація Windows API для забезпечення запуску програм Windows на UNIX-подібних операційних системах. Є копією більшості функцій ядра Windows, але написаних з нуля без застосування напрацювань цієї корпорації. Назва є рекурсивним акронімом і розшифровується, як «Wine Is Not an Emulator». Wine є відкритим програмним забезпеченням і розповсюджується під ліцензією GNU GPL. Приблизно половина вихідного коду написана добровольцями, а інша частина комерційно зацікавленою компанією CodeWeavers, яка реалізує технічну підтримку Wine. Основними підтримуваними платформами є GNU/Linux, *BSD, Solaris і Mac OS.

7.15. Secure Shell – захищена командна оболонка

Secure Shell, SSH (англ. Secure SHell) – мережевий протокол рівня застосувань, який дозволяє здійснювати віддалене управління комп'ютером і тунелюванням TCP-з'єднань. З протоколів, що забезпечують захист даних при їх передачі, серед користувачів співтовариства UNIX найбільш популярний ssh (secure shell – захищена оболонка). Дані, призначені для передачі по мережі за допомогою цього протоколу, шифруються. Очевидно, що шифровані дані можуть бути перехоплені на маршрутизаторі або в локальній мережі, але технологія, яка дозволяє декодувати їх, у цей час відсутня. Теоретично, потужний комп'ютер здатний впоратись із завданням розшифровки інформації, але для цього йому буде потрібно дуже багато часу. За останні роки сервер ssh став універсальним інструментом взаємодії.

7.16. Система портів і пакетів у FreeBSD

Система портів – це засіб компіляції програмного забезпечення з вихідного коду за допомогою суворо регламентованої, структурованої та автоматизованої процедури, що гарантує безпечність та цілісність встановлюваних програм, код яких отримано безпосередньо з сайту розробника. Система працює завдяки великій мережі розробників портів, що слідкують за змінами і підтримують сценарії, забезпечуючи коректне виконання процедур компіляції та встановлення програм у FreeBSD. Отже, «порт» у FreeBSD – це набір сценаріїв та виправлень, що розташовані у визначеній точці системи FreeBSD і містить спеціальний makefile, який дозволяє встановлювати програмне забезпечення однією командою – `make install`. Завантаження вихідного коду, конфігурування та компіляція, здійснюється в автоматичному режимі.

Дерево портів знаходиться у каталозі `/usr/ports` і має наступну структуру. На першому рівні дерева портів знаходяться каталоги категорій (наприклад, `audio`, `graphics`, `mail`, `emulators`). На другому рівні знаходяться каталоги програмних пакетів (`xmms`, `gimp`, `mutt`, `wine`). Ось приклади для деяких пакетів: `XMMS: /usr/ports/audio/xmms`, `GIMP: /usr/ports/graphics/gimp`, `Mozilla: /usr/ports/www/mozilla`, `KDE: /usr/ports/x11/kde3`, `Blackbox: /usr/ports/x11-wm/blackbox`. У каталозі кожного порту містяться наступні файли:

- `makefile` – файл, що містить інформацію, необхідну для встановлення програми;
- `distinfo` – контрольна сума файла вихідного коду за алгоритмом MD5;
- `pkg-comment` – короткий опис порту;
- `pkg-descr` – детальний опис порту, іноді містить URL сайту розробника;
- `pkg-plist` – список, що містить всі встановлювані файли та дії, які потрібно виконати при деінсталяції.

Якщо прямого виходу в Інтернет немає, то необхідні файли можна скачати «вручну».

Розглянемо, основні команди для керування портами у FreeBSD:

- `make fetch` – скачати з Інтернету;
- `make extract` – розпакувати;
- `make configure` – сконфігурувати;
- `make[build]` – скомпілювати;
- `make install` – встановити;
- `make deinstall` – видалити;
- `make clean` – видалити тимчасові файли, які залишилися після встановлення.

Звичайно, система портів є дуже зручною, але у випадках, коли програмні додатки треба встановити швидко, цей метод є не дуже вдалим. У системі FreeBSD існує метод, що дозволяє встановлювати вже скомпільовані програмні додатки. Він називається системою пакетів. У системі GNU/Linux теж є аналог такого методу, відомий як RPM. Система пакетів – це засіб пакування програмного забезпечення (а

також файлів конфігурації, потрібних бібліотек та документації), що дозволяє встановити все необхідне на комп'ютері. Вона веде базу даних встановлених пакетів, що дозволяє встановлювати програми з ftp-сервера. Встановлює потрібні для коректної роботи пакети, забезпечує можливість оновлення або деінсталяції. Готові пакети для операційної системи FreeBSD можна завантажити з ftp.freebsd.org або інших його дзеркал.

Розглянемо, команди для роботи з пакетами:

- `pkg_info` – інформація про встановлені пакети;
- `pkg_add` – процес встановлення пакету;
- `pkg_delete some_pkg` – видалення пакету.

КОНТРОЛЬНІ ПИТАННЯ

- 1) Охарактеризуйте поняття – системне програмне забезпечення.
- 2) Що які додатки включає офісний пакет Apache OpenOffice?
- 3) Охарактеризуйте поняття – інструментальне програмне забезпечення.
- 4) Охарактеризуйте поняття – прикладне програмне забезпечення.
- 5) Що таке OpenDocument Format?
- 6) Які є стандарти розробки програмного забезпечення?
- 7) Чим важливий екранно-орієнтований текстовий редактор vi?
- 8) Що таке The Document Foundation?
- 9) Які є ліцензії на програмне забезпечення?
- 10) Чим важливий файловий менеджер Midnight Commander?
- 11) Які є програми для запису дисків і перегляду документів?
- 12) Які є типи програмного забезпечення?
- 13) Чим важливий емулятор Wine?
- 14) Які є види програмного забезпечення?
- 15) Охарактеризуйте поняття – консольний текстовий редактор.

СПИСОК ПИТАНЬ, ВИНЕСЕНИХ НА САМОСТІЙНУ РОБОТУ

1. Використання GNU/Linux у сучасних смартфонах. Android OS.
2. Аналіз розвитку Open Source в Україні.
3. Мережеве конфігурування GNU/Linux.
4. Основи роботи з операційною системою GNU/Linux.
5. Особливості архітектури операційної системи GNU/Linux.
6. Структура файлової системи в UNIX.
7. Підтримка мережі в UNIX.
8. Порівняльний аналіз операційних систем: GNU/Linux і MS Windows.
9. Використання GNU/Linux в школі.
10. Суперкомп'ютери на базі UNIX.
11. Мережі TCP/IP.
12. Інсталювання GNU/Linux.
13. Вільне програмне забезпечення в навчанні інформатики.
14. Інформаційна безпека при роботі з ssh в GNU/Linux.
15. Інтеграція Unix-платформ у MS Windows-мережі.
16. Утиліти для роботи з дисковими носіями.
17. Графічний інтерфейс. Xorg.
18. Графічний інтерфейс. XFree86.
19. Засоби налаштування користувача графічного інтерфейсу.
20. Класифікація сучасних операційних систем в історичному плані.
21. Управління пристроями в UNIX.
22. Призначення та організаційні відмінності сценаріїв запуску у вільних операційних системах.
23. Термінальний інтерфейс користувача.
24. Основні поняття та принципи функціонування консольного інтерактивного режиму роботи операційної системи.
25. Основні поняття та принципи побудови GUI.
26. Налаштування правил доступу до об'єктів файлової системи GNU/Linux.
27. Налаштування робочого середовища в GNU/Linux.
28. Перегляд та обробка мультимедійного контенту.
29. Системи друку в GNU/Linux.
30. Використання утиліт su та sudo.
31. Періодичні процеси.
32. Засоби shell як мови програмування високого рівня.
33. Дисковий кеш.
34. Огляд комп'ютерних систем UNIX.
35. Операції вводу/виводу в GNU/Linux.

36. Резервне копіювання в UNIX.
37. Системне журналювання в GNU/Linux.
38. Основи мережного адміністрування засобами UNIX.
39. Моніторинг системи.
40. Управління процесами.
41. Віддалене адміністрування UNIX.
42. Огляд комп'ютерних систем GNU/Linux.
43. Еволюція операційних систем.
44. Паралельні обчислення.
45. Віртуальна пам'ять.
46. Алгоритми планування процесів.
47. Управління розподіленими процесами.
48. Архітектура протоколів TCP/IP.
49. Принцип роботи віртуальних машин.
50. Модифікації операційних систем GNU/Linux.
51. Поняття гостьової операційної системи.
52. Поняття хост-системи.
53. Етапи встановлення операційної системи GNU/Linux.
54. Основні відмінності встановлення серверної операційної системи від класичної.
55. Порядок створення локальних користувачів та груп на сервері під управлінням GNU/Linux.
56. Вбудовані користувачі та групи користувачів GNU/Linux.
57. Призначення DNS- сервера.
58. Призначення DHCP-сервера.
59. Призначення контролера домена.
60. Завантаження операційної системи через мережу.
61. Основні течії розвитку UNIX-подібних операційних систем.
62. Файлові системи працюючі в UNIX.
63. Точка монтування у файловій системі.
64. Коренева файлова система.
65. Призначення області підкачки операційної системи.
66. Віртуальна пам'ять.
67. Яку інформацію містить файл /etc/fstab.
68. Яку інформацію містить файл /etc/rc.conf.
69. Яку інформацію містить файл /etc/hosts.
70. Засоби конфігурування ядра Linux.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Гладкий М. В. Операційні системи: навчальний посібник. – Київ: КПІ ім. Ігоря Сікорського, 2018. – 312 с.
2. Копійка О. В., Костюк Ю. І. Операційні системи та системне програмування: навчальний посібник. – Львів: Львівська політехніка, 2019. – 356 с.
3. Литвиненко В. І., Пасічник В. В. Операційні системи: принципи побудови та адміністрування: навчальний посібник. – Львів: Новий Світ-2000, 2018. – 400 с.
4. Мельник А. О., Шпак З. Я. Комп'ютерні системи та мережі. Частина 2. Операційні системи: навчальний посібник. – Львів: Львівська політехніка, 2017. – 340 с.
5. Яворський Б. І. Адміністрування операційних систем Linux: навчальний посібник. – Київ: Ліра-К, 2020. – 296 с.
6. Love R. Linux Kernel Development. – 3rd ed. – Boston: Addison-Wesley Professional, 2018. – 456 p.
7. Kerrisk M. The Linux Programming Interface. – San Francisco: No Starch Press, 2019. – 1552 p.
8. Ward B. How Linux Works: What Every Superuser Should Know. – 2nd ed. – San Francisco: No Starch Press, 2019. – 464 p.
9. Nemeth E., Snyder G., Hein T., Whaley B. UNIX and Linux System Administration Handbook. – 5th ed. – Boston: Addison-Wesley Professional, 2017. – 1232 p.
10. Shotts W. The Linux Command Line. – 2nd ed. – San Francisco: No Starch Press, 2019. – 504 p.
11. Ahmed M. Learning Linux Administration. – Birmingham: Packt Publishing, 2018. – 370 p.
12. Bresnahan C., Blum R. LPIC-1 Linux Professional Institute Certification Study Guide. – 5th ed. – Indianapolis: Sybex, 2019. – 768 p.
13. van Vugt S. Red Hat RHCSA 8 Cert Guide (EX200). – Indianapolis: Pearson IT Certification, 2020. – 640 p.
14. Turnbull J. The Docker Book: Containerization is the New Virtualization. – 2017 edition. – 2017. – 282 p.
15. Hunt C. TCP/IP Network Administration. – 3rd ed. – Sebastopol: O'Reilly Media, 2017. – 746 p.