

Міністерство освіти і науки України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

ОПОРНИЙ КОНСПЕКТ ЛЕКЦІЙ З КУРСУ

«Дослідження сучасних операційні системи»
з освітньо-кваліфікаційного рівня «спеціаліст»
спеціальність «Комп'ютерні системи та мережі»

Дослідження сучасних операційних систем
Конспект лекцій

Тема 1. Огляд операційних систем.....	4
Основні терміни.....	4
1.1 Типи операційних систем.....	4
Функції та компоненти ОС.....	5
1.2 Архітектура сучасних ОС.....	6
1.3 Призначення окремих блоків сучасних операційних систем.....	6
Ядро.....	7
Підсистема управління введенням-виведенням.....	7
Підсистема управління оперативною пам'яттю.....	8
Підсистема управління задачами (процесами).....	8
Графічний інтерфейс користувача.....	9
1.4 Мережеві функції операційних систем.....	10
Тема 2. Операційні системи Windows server 2003 та 2008.....	13
2.1 Основні функції ОС Windows 2003 та 2008.....	13
Дистрибутиви.....	13
2.2 Системні вимоги.....	13
2.3 Архітектура ОС Windows 2003.....	13
Диспетчер процесів та атрибути процесів.....	15
Тема 3. Операційна система Linux.....	18
3.1 Історія виникнення Linux.....	18
GNU (Самостійно).....	18
Вільний Unix!.....	18
Minix.....	19
Linux.....	20
3.2 Основні функції ОС і архітектура.....	21
Архітектура Linux.....	21
3.3 Ядро ОС.....	21
3.4 Файлова система.....	22
Типи файлів.....	22
3.4.1. Файли фізичних пристроїв.....	22
3.4.2. Іменовані канали (pipes).....	23
3.4.3. Доменні гнізда (sockets).....	24
3.4.4. Символічні посилки.....	24
3.4 Пакети прикладних програм.....	25
Робочий стіл.....	27
Диспетчери файлів.....	27
Емулятор терміналу.....	28
Мережа WWW-браузери.....	28
Поштові програми.....	29
Обмін повідомленнями.....	30

Офісні програми	31
Графіка.....	31
Векторна графіка	32
Робота з PostScript та PDF	32
Тривимірна графіка	33
Мультимедіа.Музичні скриньки	33
Музичні редактори	34
Редактори й фільтри оцифрованого звуку.....	34
Відеопрогравачі	35
3.5 Дистрибутиви.....	35
Debian та Ubuntu	36
RedHat та Fedora	38
Red Hat Enterprise Linux.....	38
Fedora Linux	40
Suse та Novell	40
Novell	42
Slackware та GNU	44
GNU	45
Дистрибутиви GNU на основі HURD Bee GNU/Hurd	45
Debian GNU/Hurd.....	45
Тема 4. Процес завантаження Linux.....	47
4.1 BIOS та завантажувач системи (Boot Loader).....	47
4.2 Завантажувачі GRUB, syslinux, lilo.....	48
GRUB	48
Syslinux	48
LILO	49
Процес завантаження Linux	49
4.3 Процеси.....	49
4.3.1 Загальні відомості	49
4.3.2 Демони Лінукс	50
4.4 Процес init.....	53
Диспетчер пристроїв udev.....	53
Робота зі службами.....	56
Тема 5. Забезпечення безпеки в операційних системах.....	57
5.0 Основні терміни і поняття	57
5.1 Основні функції захисту.....	57
Головні захисні механізми ОС Windows	58
Головні захисні механізми ОС сімейства UNIX	58
5.2 Антивірусний захист.....	58
5.3 Мережевий фільтр.....	60

ТЕМА 1. ОГЛЯД ОПЕРАЦІЙНИХ СИСТЕМ.

ОСНОВНІ ТЕРМІНИ

Операційна система - це програмний комплекс, що забезпечує керування апаратними засобами комп'ютера, обчислювальними процесами, а також **надає середовище** для виконання прикладних програм.

Ресурс – всякий об'єкт, який може бути розподілений між обчислювальними процесами.

Апаратні ресурси комп'ютера: процесори, пам'ять, пристрої.

Програмні ресурси: засоби керування пристроями та файлами; бібліотеки програм; засоби керування задачами.

Файл - іменованій впорядкований набір даних на пристрої зберігання інформації; операційна система забезпечує організацію файлів в Файловій системі.

Файлова система - набір файлів, організованих за наперед визначеними правилами. Якщо організація файлів в файлову систему відбувається з використанням каталогів, то така файлова система називається ієрархічною.

Програма - файл, що містить набір інструкцій для виконання. В якості виконавця інструкцій програми можуть виступати:

- центральний процесор;
- інтерпретатор - інша програма, яка забезпечує розпізнавання і виконання інструкцій (в окремих випадках інтерпретатор також називають віртуальною машиною).

Обчислювальний процес (або задача) - виконання послідовності дій, яка задається програмою.

Команда - ім'я, яке використовує користувач ОС або інша програма для виконання вказаної програми (може збігатися з іменем файлу програми) або поіменованої дії (вбудованої команди).

Командний інтерпретатор - середовище, яке забезпечує інтерфейс з користувачем і виконання команд.

Мова сценаріїв або скриптів (англ. *scripting language*) - високорівнева мова програмування для написання *сценаріїв* - коротких описів дій, виконуваних системою. Сценарій - це програма, що має справу з готовими програмними компонентами.

Драйвер - програма, що забезпечує фізичну взаємодію ОС з фізичним пристроєм. Драйвер обробляє переривання пристрою, підтримує чергу запитів і перетворює запити в команди управління пристроєм.

1.1 ТИПИ ОПЕРАЦІЙНИХ СИСТЕМ.

Відносно свого **призначення**, операційні системи бувають:

- універсальні (для загального використання);
- спеціальні (для розв'язання спеціальних задач);
- спеціалізовані (виконуються на спеціальному обладнанні);
- однозадачні (в окремий момент часу можуть виконувати лише одну задачу);
- багатозадачні (в окремий момент часу здатні виконувати більше однієї задачі);

- однокористувацькі (в системі відсутні механізми обмеження доступу до файлів та на використання ресурсів системи);
- багатокористувацькі (система впроваджує поняття "власник файлу" та забезпечує механізми обмеження на використання ресурсів системи (квоти)), всі багатокористувацькі операційні системи також є багатозадачними;
- реального часу (система підтримує механізми виконання задач реального часу, тобто такі, для яких будь які операції завжди виконуються за наперед передбачуваний і незмінний при наступних виконаннях час).

Відносно способу **встановлення** (інсталяції) операційної системи, операційні системи бувають:

- вбудовані (такі, що зберігаються в енергонезалежній пам'яті обчислювальної машини або пристрою без можливості заміни в процесі експлуатації обладнання);
- невбудовані (такі, що інсталиються на один з пристроїв зберігання інформації обчислювальної машини з можливістю подальшої заміни в процесі експлуатації).

Відносно **відповідності стандартам** операційні системи бувають:

- стандартні (відповідають одному з загальноприйнятих відкритих стандартів, найчастіше POSIX);
- нестандартні (в тому числі такі, що розробляються відповідно до корпоративних стандартів).

Відносно **можливостей розширення** операційні системи бувають:

- закриті (не дозволяють розширення функціональності ОС);
- відкриті (будуються за технологіями, що забезпечують можливості розширення функціональності ОС).

Відносно можливостей **внесення змін до вихідного коду** операційні системи бувають:

- відкриті (англ. open source) - з відкритим програмним кодом;
- власницькі (англ. proprietary) - комерційні з закритим кодом.

Функції та компоненти ОС

Функції: розподіл ресурсів обчислювальної системи, ввід/вивід інформації та обмін даними між різними компонентами системи, діагностика несправностей, планування використання ресурсів системи.

Компоненти ОС: підсистеми управління процесами, підсистеми управління пам'яттю, підсистеми управління файлами і зовнішніми пристроями, підсистеми захисту даних і адміністрування, інтерфейс прикладного програмування, підсистеми інтерфейсу користувача.

Задачі **підсистеми керування ресурсами:** планування ресурсу, задоволення запитів на ресурси, відслідковування стану ресурсу, облік використання ресурсу, вирішення конфліктів між процесами.

Задачі **підсистеми управління пам'яттю:** Відстежування вільної та використаної пам'яті, Виділення пам'яті процесам, захист пам'яті, Звільнення пам'яті при завершенні процесів, Витіснення процесів з оперативної пам'яті на диск і повернення їх назад, Налаштування адрес програми на конкретну область фізичної пам'яті.

Завдання **підсистеми управління файлами і зовнішніми пристроями:** Віртуалізація окремих наборів даних, управління конкретним типом периферійних

зовнішніх пристроїв, перетворення символічних імен файлів у фізичні адреси, передача даних між зовнішніми пристроями і оперативною пам'яттю, реалізація концепції файлового доступу для обміну даними з зовнішніми пристроями.

Основні задачі захисту даних і адміністрування:

- Захист від збоїв і відмов апаратури,
- Захист від несанкціонованого доступу,
- Аудит (протоколювання, auditing) фіксація в системному журналі подій, пов'язаних з доступом до захищається системних ресурсів. Засоби обліку і спостереження забезпечують можливість виявити і зафіксувати важливі події, пов'язані з безпекою, або будь-які спроби створити, одержати доступ або видалити системні ресурси. Аудит використовується для того, щоб визначати навіть невдалі спроби «злому» системи.

Протоколювання/аудит традиційно були останнім рубежем оборони, що забезпечує аналіз наслідків порушення інформаційної безпеки і виявлення зловмисників. Такий аудит можна назвати пасивним. У сучасний арсенал захисних засобів увійшов активний аудит, спрямований на виявлення підозрілих дій в реальному масштабі часу. Активний аудит включає два види дій:

- виявлення нетипової поведінки (користувачів, програм або апаратури);
- виявлення початку зловмисної активності.

(Детальніше розглянуто в темі 10)

1.2 АРХІТЕКТУРА СУЧАСНИХ ОС



1.3 ПРИЗНАЧЕННЯ ОКРЕМИХ БЛОКІВ СУЧАСНИХ ОПЕРАЦІЙНИХ СИСТЕМ.

До складу операційної системи входять:

- Ядро операційної системи, що забезпечує розподіл і управління ресурсами обчислювальної системи;

– базовий набір системного програмного забезпечення, системні бібліотеки та програми обслуговування.

Ядро системи — це набір функцій, структур даних і окремих програмних модулів, які завантажуються в пам'ять комп'ютера при завантаженні операційної системи і забезпечують три типи системних сервісів:

– управління введенням-виведенням інформації (підсистема вводу-виводу ядра ОС);

– управління оперативною пам'яттю (підсистема управління оперативною пам'яттю ядра ОС);

– управління процесами (підсистема управління процесами ядра ОС). Кожна з цих підсистем представлена відповідними функціями ядра системи.

Ядро

Ядро - центральна частина операційної системи (ОС), що забезпечує додаткам координований доступ до ресурсів комп'ютера, таких як процесорний час, пам'ять і зовнішнє апаратне забезпечення. Також зазвичай ядро надає сервіси файлової системи та мережевих протоколів. Як основний елемент ОС, ядро являє собою найбільш низький рівень абстракції для доступу програм до ресурсів системи, необхідними для їх роботи. Як правило, ядро надає такий доступ виконуваним процесам відповідних додатків за рахунок використання механізмів взаємодії між процесами й звернення додатків до системних викликів ОС. Основною характерною ознакою ядра є те, що воно виконується у привілейованому режимі.

Розглянемо особливості цього режиму. Для забезпечення ефективного керування ресурсами комп'ютера ОС повинна мати певні привілеї щодо прикладних програм. Треба, щоб прикладні програми не втручалися в роботу ОС, а ОС могла втрутитися в роботу будь-якої програми, наприклад для перемикання задачі або розв'язання конфлікту через ресурси. Для реалізації таких привілеїв потрібна апаратна підтримка: процесор має підтримувати принаймні два режими роботи - **привілейований (захищений режим, режим ядра, kernel mode)** і **режим користувача (user mode)**. У режимі користувача недопустимі команди, які є критичними для роботи системи (перемикання задач, звертання до пам'яті за заданими межами, доступ до пристроїв введення-виведення тощо). Функції ядра операційної системи можуть бути виконані в наслідок виконання в прикладних програмах спеціальних функцій - системних викликів. Призначення системного виклику полягає в тому, що прикладні програми не взмозі самотужки визначити, за якими адресами знаходяться функції ядра. Системний виклик в один з машинно-залежних способів реалізує механізм отримання адрес функцій ядра та передачу в ці функції необхідних параметрів системного виклику, а також отримання результату системного виклику. Найчастіше системні виклики забезпечуються через систему переривань, завдяки чому адреса функції ядра не тільки обраховується апаратно (в процесі обробки переривання), але й забезпечується захист інформаційних ресурсів ядра. Системні виклики найчастіше мають синтаксис функції мови програмування, на якій написано ядро ОС.

Підсистема управління введенням-виведенням

Підсистема управління оперативною пам'яттю

Обсяг пам'яті, доступний процесу, може змінюватися в ході виконання, в цьому разі говорять про динамічний розподіл пам'яті. ОС повинна забезпечувати можливість виконання програм, які окремо або в сукупності перевищують за обсягом доступну основну пам'ять. Для цього в ній має бути реалізована **технологія віртуальної пам'яті**. Така технологія дає можливість розміщувати в основній пам'яті тільки ті інструкції дані процесу, які потрібні в поточний момент часу, при цьому вміст решти адресного простору зберігається на диску. Будь яка програма може виконуватись лише тоді, коли вона завантажена в оперативну пам'ять, так само, будь які дані з файлів можуть оброблятися лише тоді, коли вони завантажені в оперативну пам'ять. Завантаження програми та даних в оперативну пам'ять призводить до того, що в оперативній пам'яті одночасно знаходяться одразу кілька компонентів: ядро операційної системи, командний інтерпретатор, програма, що виконується, та дані, що обробляються. Крім того, програма в процесі свого виконання може звертатись до підсистеми управління оперативною пам'яттю з запитами на виділення додаткової - динамічної - оперативної пам'яті. В багатозадачних операційних системах кількість компонентів, що одночасно можуть знаходитись в оперативній пам'яті зростає пропорційно кількості задач і може сягати сотень. Підсистема управління оперативною пам'яттю забезпечує розподіл оперативної пам'яті між різними компонентами, а також розподіляє пам'ять під кеш системи введення-виведення. В окремих багатозадачних операційних системах підсистема управління оперативною пам'яттю також забезпечує віртуалізацію оперативної пам'яті, завдяки чому кожна задача (процес) отримує власну віртуальну пам'ять, причому таким чином, що нестача реальної (фізичної) пам'яті покривається за рахунок перерозподілу даних між оперативною пам'яттю та зовнішнім накопичувачем і переміщення даних між фізичною оперативною пам'яттю і зовнішнім накопичувачем приховується від задач. Це переміщення називається пейджінгом (англ. paging) або свопінгом (англ. swapping — обмін) — в залежності від термінології конкретної ОС. Введення механізму віртуалізації оперативної пам'яті дозволяє отримати два корисних наслідки:

- кожна задача фактично виконується у власному адресовому просторі, тобто таким чином, якби вона виконувалась в одно-задачній операційній системі, завдяки чому значно зменшується вплив окремих задач однією на одну та на ядро системи, а завдяки цьому - і надійність системи;
- кожна задача отримує стільки віртуальної оперативної пам'яті, скільки потрібно, а не стільки, скільки є наявної фізичної оперативної пам'яті. Віртуалізація оперативної пам'яті вимагає апаратної підтримки і звичайно забезпечується через спеціальну таблицю сторінок пам'яті, котра містить відповідності між віртуальними та фізичними адресами.

Підсистема управління задачами (процесами).

Процес — це одне з найважливіших понять інформатики. За визначенням це екземпляр виконуваної програми, включаючи змінні та стан програми. Процеси слід відрізнити від нитей виконання — ниті є складовими процесу, так що кожен процес має власний адресний простір, а ниті розподіляють спільний адресний простір

процесу. Процес складається щонайменше з одної, головної ниті виконання. Кожному процесу мають бути виділені такі ресурси:

- процесор
- пам'ять
- доступ до пристроїв вводу-виводу
- файли

Кожен процес має «батька» (батьківський процес). Він також може мати (але не мусить) «нащадків» (дочірні процеси). Таким чином створюється дерево процесів. Керування процесами здійснює ядро операційної системи. Під час виконання процес може знаходитися в одному із станів:

- виконання,
- очікування на доступ до ресурсів, які надає операційна система,
- готовності до виконання,
- щойно створений,
- завершений.

Планування виконання завдань (англ. Scheduling) є однією з ключових концепцій в багатозадачності і багатопроцесорних систем, як в операційних системах загального призначення, так і в операційних системах реального часу. **Планування процесів** полягає в призначенні пріоритетів процесам в черзі з пріоритетами. Утиліта, що виконує це завдання, називається планувальником (англ. Scheduler). Найважливішою метою планування завдань є якнайповніше завантаження доступних ресурсів. Для забезпечення **загальної продуктивності системи** планувальник має опиратися на:

- Використання процесора (-ів) - дати завдання процесору, якщо це можливо.
- Пропускна здатність - кількість процесів, що виконуються за одиницю часу.
- Час на завдання - кількість часу, для повного виконання певного процесу.
- Очікування - кількість часу, який процес очікує в черзі готових.
- Час відповіді - час, який проходить від подання запиту до першої відповіді на запит.
- Справедливість - Рівність процесорного часу для кожної ниті.

У середовищах обчислень реального часу, наприклад, на пристроях, призначених для автоматичного управління в промисловості (наприклад, робототехніка), планувальник завдань повинен забезпечити виконання процесів в перебігу заданих часових проміжків (час відгуку); це критично для підтримки коректної роботи системи реального часу. Підсистема управління задачами (процесами) забезпечує створення задачі (процесу), завантаження програмного коду і його виконання та завершення задачі (процесу). В багатозадачних системах підсистема управління задачами (процесами) також забезпечує механізми залежностей між задачами, в тому числі: синхронізацію задач та успадкування властивостей.

Графічний інтерфейс користувача.

Графічний інтерфейс користувача — інтерфейс між комп'ютером і його користувачем, що використовує піктограми, меню, і вказівний засіб для вибору функцій та виконання команд. Зазвичай, можливе відкриття більше, ніж одного вікна на одному екрані. ГІК — система засобів для взаємодії користувача з комп'ютером, заснована на представленні всіх доступних користувачеві системних

об'єктів і функцій у вигляді графічних компонентів екрану (вікон, значків, меню, кнопок, списків і т. п.). При цьому, на відміну від інтерфейса командного рядка, користувач має довільний доступ (за допомогою клавіатури або пристрою координатного введення типу «миша») до всіх видимих екранних об'єктів. Вперше концепція ГІК була запропонована вченими з дослідницької лабораторії Xerox PARC в 1970-х, але отримала комерційне втілення лише в продуктах корпорації Apple Computer. У операційній системі AMIGAOS ГІК з багатозадачністю був використаний в 1985 р. В даний час ГІК є стандартній складовій більшості доступних на ринку операційних систем і застосунків. Приклади операційних систем, що використовують ГІК: Mac OS, Ubuntu, Microsoft Windows, NEXTSTEP, OS/2. Більшість сучасних операційних систем мають графічний інтерфейс користувача (ГІК, англ. Graphical User Interfaces, GUIs, вимовляється як «гуїз»). В деяких старіших ОС ГІК вбудований у ядро, як наприклад у оригінальних реалізаціях Microsoft Windows чи MacOS. Більшість сучасних ОС є модульними і графічна підсистема у них відділена від ядра (як наприклад у Лінукс, Mac OS X і частково у Віндовз). Багато операційних систем дозволяють користувачеві встановити будь-який графічний інтерфейс на власний вибір. Типовим прикладом у більшості Юнікс-систем (BSD, Лінукс, Minix) є віконна система X у поєднанні з графічним менеджером KDE чи Gnome. Для Юнікс-систем графічний інтерфейс не є необхідним. Графічний інтерфейс користувача невпинно розвивається. Наприклад, інтерфейс Windows модифікується щоразу при випуску нової основної версії, а ГІК MacOS було кардинально змінено після випуску MacOS X у 2001 році.

1.4 Мережеві функції операційних систем.

Комп'ютерна мережа - система зв'язку між двома чи більше комп'ютерами. У ширшому розумінні комп'ютерна мережа - це система зв'язку через кабельне чи повітряне середовище, самі комп'ютери різного функціонального призначення і мережеве обладнання. Середовищами передавання у комп'ютерних мережах можуть бути телефонні кабелі, виті пари, волоконно-оптичні кабелі, радіохвилі, світлові сигнали.

Головним завданням **стеку TCP / IP** є об'єднання підмереж в мережу через шлюзи. Кожна мережа працює за своїми власними законами, проте передбачається, що шлюз може прийняти пакет з іншої мережі і доставити його за вказаною адресою. Реально, пакет з однієї мережі передається в іншу підмережу через послідовність шлюзів, які забезпечують наскрізну маршрутизацію пакетів по всій мережі.

Під **шлюзом** розуміється точка з'єднання мереж. При цьому з'єднуватися можуть як локальні мережі, так і глобальні мережі. В якості шлюзу можуть виступати як спеціальні пристрої, маршрутизатори, так і комп'ютери, які мають програмне забезпечення, що виконує функції маршрутизації пакетів.

Маршрутизація - це процедура визначення шляху проходження пакету з однієї мережі в іншу.

Мережева операційна система (англ. Network operating system) - це операційна система, яка забезпечує обробку, зберігання та передачу даних в інформаційній мережі.

Головними завданнями мережної ОС є розподіл ресурсів мережі (наприклад, дискового простору) і адміністрування мережі. Системний адміністратор визначає колективні ресурси, задає паролі, визначає права доступу для кожного користувача або групи користувачів. Звідси мережеві ОС ділять на мережеві ОС для серверів і мережеві ОС для користувачів.

Якщо комп'ютер надає свої ресурси іншим користувачам мережі, то він грає роль **сервера**. При цьому комп'ютер, який звертається до ресурсів іншої машини, є **клієнтом**. Комп'ютер, що працює в мережі, може виконувати функції або клієнта, або сервера, або поєднувати обидві функції.

Мережеві засоби ОС поділяються на три компоненти:

- серверна частина ОС – засоби надання локальних ресурсів і сервісів у загальне користування;
- клієнтська частина ОС – засоби запиту на доступ до віддалених ресурсів і сервісів;
- Транспортні засоби ОС, разом з комунікаційною системою забезпечують передачу повідомлень між комп'ютерами.

Мережевою службою називається сукупність серверної і клієнтської частин ОС, що надають доступ до конкретного типу ресурсу комп'ютера через мережу

Сервіс - інтерфейс між споживачем послуг (користувачем або додатком) та постачальником послуг (службою).

Якщо виконання серверних функцій є основним призначенням комп'ютера, такий комп'ютер називається виділеним сервером. Залежно від того, який ресурс сервера розділяється, він називається файл-сервером, факс-сервером, принт-сервером, сервером додатків і т.д. Виділений сервер не прийнято використовувати в якості комп'ютера для виконання поточних завдань, не пов'язаних з його основним призначенням, так як це може зменшити продуктивність його роботи як сервера.

До основних **функцій мережевих** ОС відносять управління каталогами та файлами; управління ресурсами; комунікаційні функції; захист від несанкціонованого доступу; забезпечення відмовостійкості; управління мережею.

Управління каталогами та файлами в мережах полягає в забезпеченні доступу до даних, фізично розташованих в інших вузлах мережі.

Управління ресурсами включає обслуговування запитів на надання ресурсів, доступних по мережі.

Комунікаційні функції забезпечують адресацію, буферизацію, вибір напрямку для руху даних у розгалуженій мережі (маршрутизацію), управління потоками даних та ін

Захист від несанкціонованого доступу - підтримка цілісності даних та їх конфіденційність.

Відмовостійкість характеризується збереженням працездатності системи при впливі дестабілізуючих факторів. Відмовостійкість забезпечується застосуванням для серверів автономних джерел живлення, відображенням або дублюванням інформації в дискових накопичувачах.

Управління мережею пов'язано із застосуванням відповідних протоколів управління. У більшості випадків в мережевому програмному забезпеченні реалізуються протоколи ICMP і SNMP з стека TCP / IP, рідше використовується

протокол CMIP (Common Management Information Protocol) з семиуровневої моделі протоколів ISO.

ТЕМА 2. ОПЕРАЦІЙНІ СИСТЕМИ WINDOWS SERVER 2003 ТА 2008.

2.1 ОСНОВНІ ФУНКЦІЇ ОС WINDOWS 2003 ТА 2008

Дистрибутиви

Windows Server 2003 Standard Edition призначена для корпоративного використання і таких задач: служба друку, файлові сервіси, маршрутизація і віддалений доступ, забезпечення роботи СУБД і т.д.

Windows Server 2003 Enterprise Edition Є платформою для розгортання бізнес-задач будь-якого масштабу, включаючи служби Інтернету.

Windows Server 2003 Datacenter Edition орієнтована на забезпечення максимального рівня продуктивності й надійності для критично важливих додатків і завдань. У цій редакції відсутні деякі служби, використання яких доцільно лише в невеликих компаніях або групах.

Windows Server 2003 Web Edition Призначена для веб-хостингу й підтримки XML веб-служб у невеликих організаціях і підрозділах.

Windows Compute Cluster Server 2003 (2006 р.) розроблена для високотехнологічних додатків, які вимагають кластерних обчислень.

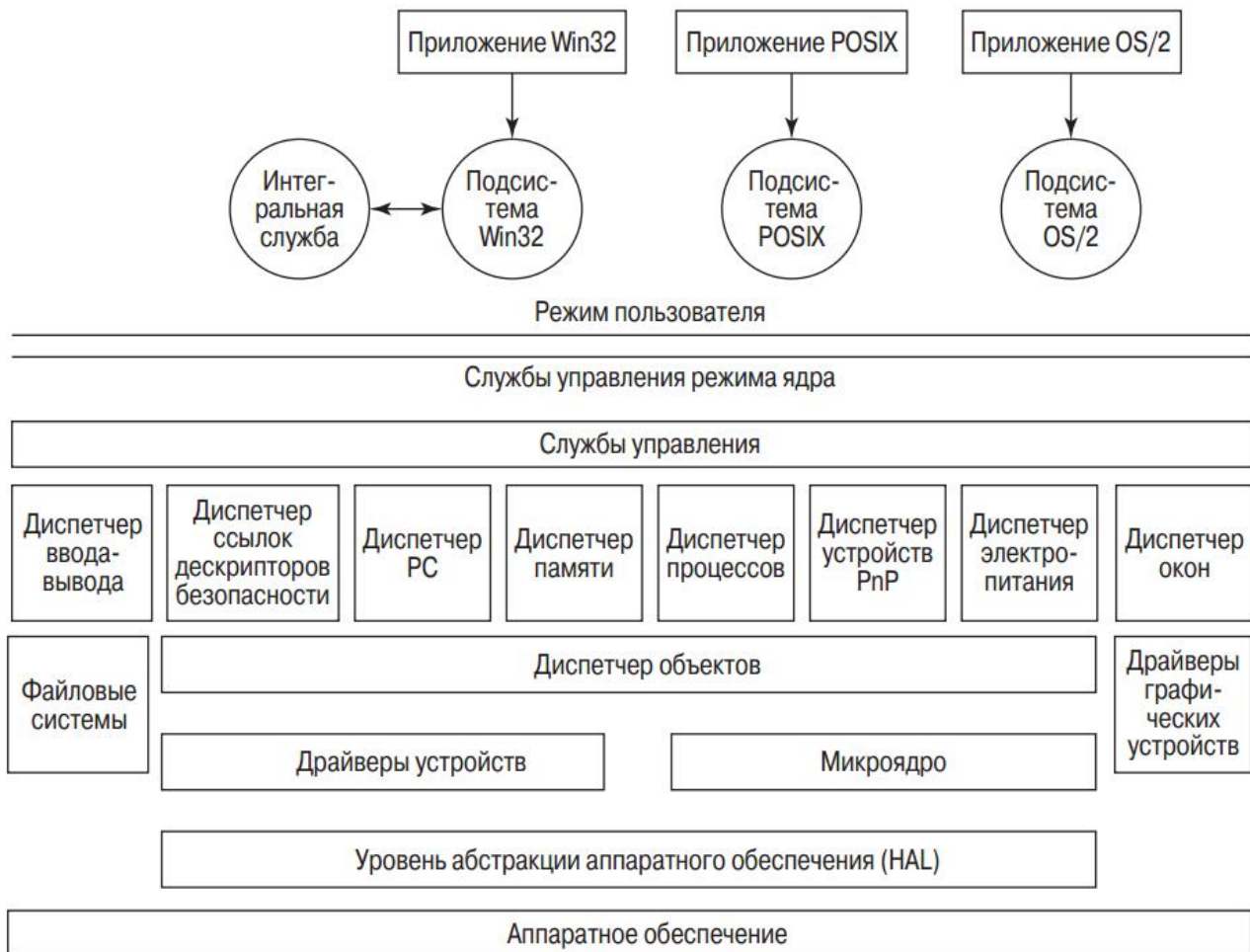
Оновлення: *Windows Server 2003 R2* (2005 р.).

2.2 Системні вимоги

	x86	Itanium	x64
Standard Edition	ОЗП: 128 МВ- 4(GB) Макс к-сть ЦП: 4	-	512 МВ - 4(GB) Макс к-сть ЦП: 4
Enterprise Edition	128 МВ- 32 (GB) Intel Pentium/Celeron, AMD K6/Athlon/Duron Макс к-сть ЦП: 8	1 GB - 1024 GB Макс к-сть ЦП: 8	1 GB - 1024 GB Макс к-сть ЦП: 8
Datacenter Edition	512 МВ – 128 (GB) Макс к-сть ЦП: 32	1 GB - 1024 GB Макс к-сть ЦП: 64	1 GB - 1024 GB Макс к-сть ЦП: 64

2.3 Архітектура ОС Windows 2003

Архітектура Windows Server 2003 включає в себе два основних рівня - пользовательский режим (user mode) і режим ядра (kernel mode). Рисунок



У режимі користувача сторонні виробники програмного забезпечення можуть виконувати системні виклики готових інтерфейсів API і об'єктно - орієнтованих компонентів.

Підсистеми середовища перехоплюють виклики додатків, звернені до інтерфейсу API конкретної операційної системи, і перетворюють ці виклики у формат, зрозумілий Windows 2003. Підсистеми середовища, відмінні від Win32, забезпечують базову підтримку застарілих додатків, що не відносяться до Win32, і не більше того.

Обмеження на запуск програм в режимі користувача

- Програми не мають прямого доступу до обладнання. Додаток звертається до об'єктів режиму користувача, які взаємодіють з об'єктами режиму ядра, а ті, у свою чергу, спускаються ще нижче і звертаються до апаратно - залежного рівня (Hardware Abstraction Layer , HAL).

- Програми не мають прямого доступу до драйверів пристроїв.

- Програмам виділяється тільки обмежений адресний простір в оперативній пам'яті.

- Windows Server 2003 використовує простір жорсткого диска в якості квазіоперативної пам'яті. Віртуальна пам'ять (virtual memory) - це сукупність всіх типів пам'яті в системі , яка являє собою комбінацію фізичної пам'яті комп'ютера і файлу підкачки .

- Програми, запущені в режимі користувача, виконуються процесами з нижчим пріоритетом, ніж будь-які служби і функції, запущені в режимі ядра.

Диспетчер процесів та атрибути процесів

Task Manager (Диспетчер задач) служить для перегляду поточних даних про продуктивність системи. У цій утиліті основними є три індикатори: використання процесора, використання віртуальної пам'яті й запущені процеси й програми.

Основні лічильники диспетчера задач

Лічильник	Опис
Image Name (Имя образа)	Ім'я процесу
PID (Process Identifier) (Идентификатор процесса)	Числове значення, яке унікальним чином визначає процес під час його роботи
CPU Usage (Загрузка ЦП)	Виразений у відсотках час, впродовж якого процес використав час процесора з моменту останнього оновлення
CPU Time (Время ЦП)	Сумарний час процесора, використаний процесом із часу його запуску (у секундах)
Memory Usage (Память— использование)	Обсяг віртуальної пам'яті, використовуваної процесом (у кілобайтах)
Memory Usage Delta (Память— изменение)	Зміна обсягу пам'яті з моменту останнього відновлення. Диспетчер задач відображає негативні значення
Peak Memory Usage (Память— максимум)	Максимальний обсяг виділеної пам'яті, використаної процесом з моменту запуску. Виділеною пам'яттю є пам'ять, яку процес використовує на фізичному носії (наприклад, в ОЗП) або у файлі підкачування
Page Faults (Ошибки страницы)	Число переривань, які виникають, коли застосування намагається прочитати або записати дані в неіснуючу віртуальну пам'ять
USER Objects (Объекты USER)	Число об'єктів USER, які використовуються тепер певним процесом
I/O Reads (Число чтений)	Число операцій введення/виведення, згенерованих процесом читання, включаючи операції для файлів, мережі й пристроїв
I/O Read Bytes (Прочитано байт)	Число байт, прочитаних у ході операцій уведення/виводу, згенерованих процесом читання, включаючи операції для файлів, мережі й пристроїв
Session ID (Код сеанса)	Ідентифікатор сеансу служб терміналів (Terminal Services), якщо вони встановлені
User Name (Имя пользователя)	Ім'я користувача, який володіє процесом служб терміналів (zareestrovaniy віддалено)
Page Faults Delta (Ошибки страницы— изменение)	Зміна числа помилок сторінок з моменту останнього

	відновлення
Virtual Memory Size (Объем виртуальной памяти)	Обсяг віртуальної пам'яті або адресного простору, виділеного процесу
Paged Pool (Выгружаемый пул)	Віртуальна пам'ять, доступна для кешування на диск, яка містить у собі всю користувацьку пам'ять і частина системної пам'яті. Кешування являє собою переміщення рідко використовуваних компонентів робочої пам'яті з ОЗП на інший носій, звичайно на жорсткий диск
Non-Paged Pool (Невыгружаемый пул)	Обсяг пам'яті операційної системи, використовуваної, процесом (у кілобайтах). Дана пам'ять ніколи не вивантажується на диск
Base Priority (Базовый приоритет)	Визначає порядок диспетчеризації потоків процесу для обробки процесором. У Службах черги повідомлень (Microsoft Message Queuing Services, MSMQ) базовий пріоритет (або пріоритет черги) визначає гроху-пріоритет черги в загальній черзі. Базовий пріоритет може бути встановлений у діапазоні від -32 768 до 32 767 (значення за замовчуванням рівно 0) будь-яким застосуванням MSMQ з дозволами на запис для черги. Приватні черги не підтримують базовий пріоритет. MSMQ маршрутизує і передає повідомлення на основі комбінації базового пріоритету й пріоритету повідомлення
Handle Count (Счетчик дескрипторов)	Число дескрипторів об'єктів у таблиці об'єктів процесу
Thread Count (Счетчик потоков)	Число потоків, запущених у процесі
GDI Objects (Объекты GDI)	Число об'єктів GDI, використовуваних у цей момент процесом. Об'єкти з бібліотеки графічного користувацького інтерфейсу (Graphics Device Interface, GDI), що входить в інтерфейс прикладного програмування (API) для пристроїв виводу графіки
I/O Writes (Число записей)	Число операцій уведення/виводу, згенерованих процесом запису, включаючи операції для файлів, мережі й пристроїв
I/O Write Bytes (Записано байт)	Число байт, записаних у ході операцій уведення/виводу, згенерованих процесом запису, включаючи операції для файлів, мережі й пристроїв
I/O Other (Прочий ввод/вывод)	Число операцій уведення/виводу, згенерованих процесом, який не є ні читанням, ні записом, включаючи операції для файлів, мережі й пристроїв. Прикладом такого типу операції є функція керування
I/O Other Bytes (Прочих байт)	Число байт, переданих у ході операцій

при вводе/выводе)	уведення/виводу, згенерованих процесом, який не є ні читанням, ні записом, включаючи операції для файлів, мережі й пристроїв
-------------------	--

Засіб перегляду подій Оснащення (Event Viewer, Просмотр событий) дозволяє переглядати журнали подій, що згенеровані програмами, службою безпеки й системою.

Три типи стандартних (основних) журналів.

- *Журнал приложений (Application log)* - фіксує події, зареєстровані застосуваннями. Наприклад, текстовий редактор може зареєструвати в даному журналі помилку при відкритті файлу.
- *Журнал системи (System log)* - записує події, які реєструються системними компонентами **Windows Server 2003**. Наприклад, у системний журнал записуються такі події, як збій у процесі завантаження драйвера або іншого системного компонента при запуску системи.
- *Журнал безпеки (Security log)* - містить запису, пов'язані із системою безпеки. За допомогою цього журналу можна відслідковувати зміни в системі безпеки й ідентифікувати проломи в захисті. У даному журналі можна реєструвати спроби входу в систему. Для перегляду журналу необхідно мати права адміністратора. За замовчуванням реєстрація подій у журналі безпеки відключена.

Performance (Производительность) - містить у собі два компоненти:

Activex-Елемент System Monitor

оснащення *Performance Logs and Alerts* (Оповещения и журналы безопасности).

Графічні засоби *System Monitor* дозволяють візуально відслідковувати зміну продуктивності системи. За допомогою *System Monitor* можна одночасно переглядати дані з декількох комп'ютерів у вигляді динамічних діаграм, на яких відображається поточний стан системи й показання лічильників.

Оснащення *Performance Logs and Alerts* дозволяє створювати звіти на основі поточних даних продуктивності або інформації з журналів. При перевищенні лічильниками заданого значення або зменшення нижче вказаного рівня дане оснащення за допомогою служби повідомлень (Messenger) посилає оповіщення користувачеві.

ТЕМА 3. ОПЕРАЦІЙНА СИСТЕМА LINUX.

3.1 ІСТОРІЯ ВИНИКНЕННЯ LINUX.

Коріння ОС Linux бере ще з 70-х років ХХ-го століття. Точкою відліку можна вважати появу операційної системи Unix в 1969-му році в США у фірмі Bell Laboratories, дочірнього підрозділу компанії AT&T. Unix стала основною для великої кількості операційних систем промислового класу.

Linux найбільше зобов'язаний своїм життям двом проектам - GNU та Minix.

GNU (Самостійно)

Історія проекту GNU почалася у вересні 1983-го року. Основоположник проекту GNU, Річард Столлман (Richard M. Stallman) працював у цей час в лабораторії штучного інтелекту Массачусетського технологічного інституту (Massachusetts Institute of Technology, MIT, Cambridge, Massachusetts). Столлмана називають одним з найвидатніших програмістів нашого часу.

У тому середовищі, до якої належав Столлман, було прийнято вільно обмінюватися програмами і їх вихідними кодами. Ліцензія ж на Unix від AT&T, наприклад, коштувала 40 000 доларів. Дозволити собі купити її могли тільки досить великі фірми. А без володіння ліцензією, програміст не мав права використовувати вихідні коди системи в своїх розробках. Це перешкоджало обміну ідеями у сфері програмування і сильно гальмувало процес створення програм, оскільки замість того, щоб запозичити готовий шматок коду для вирішення того чи іншого завдання, розробник програми був змушений писати цю частину коду заново.

Столлман вирішив змінити цей стан речей в програмуванні. У 1983-му році він оголосив про початок розробки проекту GNU, метою якого було створення повністю відкритої операційної системи:

Четвер, 27 вересня 1983, 12:35:59 EST

Вільний Unix!

Після Дня Подяки я починаю писати Unix-сумісну програмну систему GNU (Gnu's Not Unix), яку буду надавати вільно (!) Всім, хто може її використати. Потрібна допомога у вигляді часу, грошей, програм і обладнання.

GNU міститиме ядро плюс всі утиліти, необхідні для того, щоб писати і запускати програми на C: редактор, оболонку, компілятор C, линкер, асемблер і ще кілька речей. Після цього будуть додані програма форматування тексту, YACC, гра Empire, електронна таблиця і сотні інших речей. Ми сподіваємося включити все, що зазвичай поставляється з Unix-системами, і все, що ще може виявитися корисним, у тому числі онлайн-ову і друковану документацію.

GNU буде здатна запускати програми Unix, але не буде ідентична Unix. Ми будемо вносити в систему поліпшення, ґрунтуючись на нашому досвіді роботи з іншими операційними системами ...

Абревіатура GNU розшифровується як «GNU - це не Unix» (GNU is Not Unix). Unix завжди була закритим, тобто вона позбавляє своїх користувачів свободи співробітництва, а також контролю над своїми комп'ютерами (як Windows в наші дні). Трохи пізніше Столлман написав свій знаменитий Маніфест GNU, який став

основою для ліцензії GPL (GNU General Public License). Роль цієї ліцензії не можна переоцінити, вона змінила всю комп'ютерну індустрію.

Основна ідея GPL полягає в тому, що користувач повинен володіти чотирма правами (або чотирма свободами): -Правом запускати програму для будь-яких цілей (свобода 0); -Правом вивчати пристрій програми і пристосовувати її до своїх потреб (свобода 1), що передбачає доступ до вихідного коду програми; -Правом поширювати програму, маючи можливість допомогти іншим (свобода 2); -Правом поліпшувати програму і публікувати поліпшення, на користь всієї спільноти (свобода 3), що теж передбачає доступ до вихідного коду програми.

Програмне забезпечення, яке розповсюджується під цією ліцензією, можна як завгодно використовувати, копіювати, допрацьовувати, модифікувати, передавати або продавати модифіковані (або навіть немодифіковані) версії іншим особам за умови, що результат такої переробки теж буде поширюватися під ліцензією GPL. Остання умова - найважливіше і визначальне у цій ліцензії. Вона гарантує, що результати зусиль розробників вільного ПЗ залишаться відкритими і не стануть частиною будь-якого ліцензійного продукту. Воно також відрізняє тим, що розповсюджуваного безкоштовно. Одна з вимог цієї ліцензії полягає в тому, що продаючи ПЗ під ліцензією GPL, ви повинні надати вихідні коди цього ПЗ кожному, хто захоче отримати до них доступ. Ліцензія GPL «робить ПЗ вільним до змін».

До 1990-го року в рамках проекту GNU було створено більшість компонентів, необхідних для функціонування вільної операційної системи. Крім текстового редактора Emacs, Столлман створив компілятор gcc (GNU C Compiler) і відладчик gdb. Будучи видатним програмістом, Річард Столлман поодиночці зумів створити ефективний і надійний компілятор, який перевершує за своїми якостями продукти комерційних постачальників, створювані цілими групами програмістів. Оскільки спочатку при його створенні ставилося завдання забезпечення переносимості, сьогодні існують версії цього компілятора практично для всіх операційних систем. Пізніше були створені компілятори і для інших мов програмування, включаючи C++, Pascal та Fortran. Тому зараз аббревіатура GCC розшифровується як GNU Compiler Collection.

Як пише Річард Столлман: «До 1990-го року система GNU була практично закінчена, не вистачало тільки одного з базових компонентів - ядра." Очікувалося, що ядро (воно отримало назву Hurd) буде реалізовано як набір серверних процесів, що працюють на Mach - мікроядрі, що створюється в Університеті Карнегі-Меллона, а потім в Університеті штату Юта. Початок розробки відкладалося в очікуванні випуску Mach, яке, як було обіцяно, буде випущено у вигляді вільно поширюваного програмного забезпечення. Але його поява все відкладалося, і тут з'явилося ядро, розроблене фінським студентом Лінусом Торвальдсом, що отримало назву Linux. Лінус створив його в спробах удосконалити свою домашню операційну систему Minix, про яку варто згадати окремо.

Minix

Протягом 1990-х років персональні комп'ютери на основі мікропроцесора Intel, оснащені операційними системами від Microsoft, зайняли панівне становище на ринку настільних систем і захопили також істотну частку ринку серверів - традиційної сфери застосування Unix-систем. Комп'ютери на основі Intel та Intel-сумісних процесорів досягли обчислювальної потужності, порівнянної з потужністю

робочих станцій з Unix. Але більшість комерційних Unix-систем не мали версії, здатні працювати на обладнанні Intel. Виробники Unix зазвичай тісно співпрацювали з виробниками конкретних процесорів або навіть мали частку власності в компаніях, які виробляли ці процесори, а тому були зацікавлені у використанні своїх власних розробок. Прикладами можуть служити лінійки процесорів SGI і MIPS. Оскільки апаратні можливості ПК стрімко зростали, природньо, то рано, чи пізно повинні були з'явитися варіанти Unix для комп'ютерів на основі Intel-сумісних процесорів. Один з таких варіантів Unix-подібної операційної системи, який зіграв особливу роль в історії Linux, був розроблений в січні 1987-го року Е.Таненбаумом (Andrew S. Tanenbaum), професором Університету Вріе, Амстердам, Нідерланди. Таненбаум був одним з провідних фахівців в області розробки операційних систем. Свою операційну систему Minix (Мінікс) він розробив як навчальний посібник, на прикладі якого показував студентам внутрішній устрій реальної операційної системи.

Звичайно, як операційна система, Minix не була вершиною досконалості. Вона була орієнтована на мікропроцесор Intel 80286, який в той час панував на ринку. Але в неї було одне дуже важливе якість - відкриті вихідні коди. Кожен, хто мав книгу Таненбаума «Операційні системи», міг вивчити і проаналізувати 12000 рядків коду, написаного на мові C і на асемблері. Це був той рідкісний випадок, коли вихідні коди не були замкнені під сімома печатками в сейфах розробника. Чудовий автор, Таненбаум зумів залучити найвидатніші уми комп'ютерної науки в обговорення мистецтва створення операційних систем. Minix можна було придбати і окремо від книжки, він міг бути реально встановленим на персональний комп'ютер. Студенти комп'ютерних факультетів по всьому світу працювали над книгою Таненбаума, вчитуючись в коди з метою зрозуміти, як працює та сама система, яка управляє їх комп'ютером. І одним з таких студентів був Лінус Торвальдс.

Linux

У 1991-му році, Лінус Торвальдс, фінський студент, надзвичайно захопився ідеєю написати сумісне з Unix ядро операційної системи для свого персонального комп'ютера з процесором Intel. Прототипом для майбутнього ядра стала операційна система Minix: сумісна з Unix операційна система для персональних комп'ютерів, яка завантажувалася з дискет і вміщувалася в дуже обмеженою у ті часи пам'яті персонального комп'ютера.

Нова система отримала назву «Linux». Самого Торвальдса трохи бентежило співзвуччя цієї назви з його ім'ям, тому він намагався назвати свою розробку Freax. Ця назва можна виявити у файлі kernl/Makefile версії 0.11, і у вихідних кодах інших програм. Але Арі Лемке, який надав місце для викладання системи на FTP сайті, назвав каталог pub/OS/Linux. І ця назва закріпилася за нової ОС.

Той факт, що Лінус виклав код своєї ОС в інтернет, був вирішальним у подальшій долі Linux. Хоча в 1991-му році Інтернет ще не був так широко поширений, як у наші дні, зате користувалися ним в основному люди, які мають достатню технічну підготовку. І вже з самого початку Торвальдс отримав кілька зацікавлених відгуків.

Приблизно в лютому 1992-го року Лінус висловив прохання до всіх, хто вже користувався або тестував Linux, надіслати йому листівку. Таких листівок було отримано кілька сотень з усіх кінців світу - з Нової Зеландії, Японії, Нідерландів, США. Це говорило про те, що Linux почала набувати певної популярності.

До цього часу сформувалася окрема конференція в інтернеті, присвячена Linux, - comp.os.linux. Ентузіасти утворили безліч груп користувачів і на початку 1994-го року вийшов перший номер журналу «Linux Journal». Linux привернула увагу промислових фірм і кілька невеликих компаній почали розробляти і продавати власні версії Linux.

Спочатку Лінус Торвальдс не хотів продавати свою розробку. І не хотів, щоб її продавав хтось інший. Це було чітко прописано в повідомленні про авторські права, вміщеному в файл COPYING самої першої версії - 0.01. Причому вимога Лінуса накладало значно жорсткіші обмеження на поширення Linux, ніж ті, які проголошувалися в ліцензії GNU: не дозволялося стягувати ніяких грошей за передачу або використання Linux. Але вже в лютому 1992-го року до нього стали звертатися за дозволом брати плату за розповсюдження дискет з Linux, щоб покрити тимчасові витрати і вартість дискет. Крім того, необхідно було рахуватися і з тим, що при створенні Linux використовувалося безліч вільно розповсюджуваних по Інтернету інструментів, найважливішим з яких був компілятор GCC. Авторські права на нього обумовлені в суспільному ліцензії GPL, яку винайшов Річард Столлман. Торвальдсу довелося переглянути свою заяву про авторські права, і, починаючи з версії 0.12, він теж перейшов на використання ліцензії GPL.

З технічної точки зору, Linux являє собою тільки ядро Unix-подібної операційної системи, що відповідає за взаємодію з апаратною частиною комп'ютера і виконання таких завдань, як розподіл пам'яті виділення процесорного часу різними програмами і так далі. Успіх Linux як операційної системи багато в чому обумовлений тим, що до 1991-го року в рамках проекту GNU вже було розроблено безліч утиліт, вільно розповсюджуваних в інтернеті. Лінус Торвальдс опинився зі своєю розробкою в потрібному місці в потрібний час. І Річард Столлман правий, коли стам наполягати на тому, що операційну систему слід називати не Linux, а GNU/Linux. Але назва Linux історично закріпилося за цією ОС, тому ми теж будемо називати її просто Linux (не забуваючи про заслуги Столлман і його попередників).

3.2 ОСНОВНІ ФУНКЦІЇ ОС І АРХІТЕКТУРА.

Архітектура Linux

В ОС Linux можна виділити три основні частини:

- **ядро**, яке реалізує основні функції ОС (керування процесами, пам'яттю, введенням-виведенням тощо);
- **системні бібліотеки**, що визначають стандартний набір функцій для використання у застосуваннях (виконання таких функцій не потребує переходу в привілейований режим);
- **системні утиліти** (прикладні програми, які виконують спеціалізовані задачі).

3.3 ЯДРО ОС.

Linux реалізує технологію монолітного ядра. Весь код і структури даних ядра перебувають в одному адресному просторі.

3.4 ФАЙЛОВА СИСТЕМА

Стандартні директорії:

- / Кореневий каталог файлової системи Linux в якому знаходяться системні файли та каталоги.
- /bin системні програми, основні команди ядра Linux.
- /boot завантажувач системи, та інші файли необхідні для старту Linux.
- /dev дані про фізичні пристрої та дані про керування ними (можна назвати це драйверами)
- /etc системні файли конфігурації.
- /home домашній каталог.
- /lib системні бібліотеки та модулі ядра.
- /lost+found резервні файли та файли для відновлення системи.
- /media монтуються змінні носії. Для кожного носія створюється підкаталог.
- /mnt В цей каталог монтують (на деякий час) додаткові файлові системи.
- /opt В даному каталозі зазвичай встановлюють додаткове ПЗ. Яке є доповненням до базового програмного забезпечення.
- /proc В цей каталог занесені дані про псевдофайлову систему, або віртуальну файлову систему з інформацією про процеси та ядро.
- /root Домашній каталог користувача root.
- /sbin важливі системні виконувані файли (бінарні) файли. Файли для запуску системи та команди які зазвичай використовує системний адміністратор.
- /srv В цьому каталозі знаходяться незмінні дані про сервіси (тільки для читання)
- /tmp В цьому каталозі знаходяться тимчасові файли, вони зазвичай стираються коли потрібно. Це наприклад як область підкачки.
- /usr В даному каталозі знаходяться допоміжні файли (бібліотеки). А також каталог /usr містить ієрархію підкаталогів.
- /var Каталог /var містить в собі різноманітні файли які зазвичай змінюються з часом. Наприклад тут знаходяться збереження до ігор.

Типи файлів

З погляду операційної системи файл являє собою просто потік байтів. Такий підхід дозволяє поширити концепцію файлу на фізичні пристрої й деякі інші об'єкти. Це дозволяє спростити організацію даних і обмін ними, тому що аналогічним чином здійснюється запис даних у файл, передача їх на фізичні пристрої та обмін даними між процесами. У всіх цих випадках використовується той самий підхід, заснований на ідеї байтового потоку. Тому поряд зі звичайними файлами й каталогами, файлами з погляду Linux є також:

- файли фізичних пристроїв;
- іменовані канали (named pipes);
- гнізда (*рос.* гнезда, sockets);
- символічні посилання (symlinks).

3.4.1. Файли фізичних пристроїв

З погляду ОС Linux, усі пристрої, що підключаються до комп'ютера (жорсткі і знімні диски, термінал, принтер, модем і т.д.), представляються файлами.

Фізичні пристрої бувають **двох типів**: символними (або байт-орієнтованими) і блоковими (або блок-орієнтованими). Взаємодія із символними пристроями проводиться посимвольно, у режимі потоку байтів. До таких пристроїв відносяться, наприклад, термінали. На блок-орієнтованих пристроях інформація записується (і зчитується) блоками. Прикладом пристроїв цього типу є жорсткі диски. На диск неможливо записати або зчитати з нього один байт.

Взаємодією з фізичними пристроями в Linux управляють **драйвери** пристроїв, які або вбудовані в ядро, або підключаються як окремі модулі. Для взаємодії з іншими частинами операційної системи кожний драйвер утворює комунікаційний інтерфейс, який виглядає як файл. Більшість таких файлів для різних пристроїв як би "заготовлені заздалегідь" і розташовуються в каталозі /dev. У табл. 1 наведена невелика довідка по іменам найбільше часто використовуваних спеціальних файлів

Таблиця 1 - Основні спеціальні файли.

Ім'я	Значення
/dev/console	Системна консоль, тобто монітор і клавіатура, фізично підключені до комп'ютера
/dev/hd	Жорсткі диски з Ide-Інтерфейсом. Пристрій /dev/hda1 відповідає першому розділу на першому жорсткому диску (/dev/hda), тобто на диску, підключеному як Primary Master
/dev/sd	Жорсткі диски з SCSI-Інтерфейсом
/dev/fd	Файли дисководів для гнучких дисків. Першому дисководу відповідає /dev/fd0, другому /dev/fd1
/dev/tty	Файли підтримки користувачьких консолей. Назва збереглася з тих пор, коли до системи UNIX підключалися телетайпи як термінали. В Linux ці файли пристроїв забезпечують роботу віртуальних консолей (перемикається між якими можна за допомогою <Alt>+<F1> - <Alt>+<F6>)
/dev/pty	Файли підтримки псевдо-терміналів. Застосовуються для віддалених робочих сесій з використанням telnet
/dev/ttys	Файли, що забезпечують роботу з послідовними портами. /dev/ttys0 відповідає COM1 в MS-DOS, /dev/ttys1 - COM2.
/dev/cua	Спеціальні пристрої для роботи з модемами
/dev/null	Цей пристрій — просто чорна діра. Усе, що записується в /dev/null, назавжди загублене. На цей пристрій можна перенаправляти вивід непотрібних повідомлень. Якщо /dev/null використовується як пристрій уведення, то воно поводить як файл нульової довжини

Кожному типу пристроїв у системі може відповідати кілька файлів пристроїв. Тому файли пристроїв характеризуються двома номерами: старшим і молодшим. Старший номер пристрою говорить ядру про те, до якого драйвера відноситься даний файл, а молодший номер показує, до якого саме пристрою даного типу слід звертатися.

Для файлів пристроїв команда `ls -l` замість розміру файлу видає старший і молодший номер даного пристрою.

3.4.2. Іменовані канали (pipes)

Ще один тип спеціальних файлів - іменовані канали, або буфери FIFO (First In — First Out). Файли цього типу служать в основному для того, щоб організувати

обмін даними між різними застосуваннями (pipe переводиться з англійського як труба).

Канал - це дуже зручне й широко застосовуване засіб обміну інформацією між процесами. Усе, що один процес поміщає в канал, іншої може звідти прочитати. Якщо два процеси, що обмінюються інформацією, породжені тим самим батьківським процесом (а так найчастіше й відбувається), канал може бути неіменованим. А якщо ні, то потрібно створити іменованій канал, що можна зробити за допомогою програми `mkfifo`. При цьому файл іменованого каналу бере участь тільки в ініціації обміну даними.

3.4.3. Доменні гнізда (sockets)

Гнізда - це з'єднання між процесами, які дозволяють їм взаємодіяти, не піддаючись впливу інших процесів. Взагалі гнізда (і взаємодія програм за допомогою гнізд) відіграють дуже важливу роль у Linux: вони є ключовим поняттям TCP/IP і відповідно на них цілком будується Інтернет. Однак з погляду файлової системи гнізда практично не відрізняються від іменованих каналів: це просто мітки, що дозволяють зв'язати кілька програм. Після того як зв'язок установлений, спілкування програм відбувається без участі файлу гнізда: дані передаються ядром ОС безпосередньо від однієї програми до іншої.

Незважаючи на те, що інші процеси можуть бачити файли гнізд як елементи каталогу, процеси, не що брали участь у даному конкретному з'єднанні, не можуть здійснювати над файлами гнізд операції читання/запису. Серед стандартних засобів, що використовують гнізда — система X Window, система друку й система `syslog`.

3.4.4. Символічні посилки

Жорстке посилання є просто ще одним іменем для вихідного файлу. Воно прописується в індексному дескрипторі вихідного файлу. Після створення жорсткого посилання неможливо розрізнити, де вихідне ім'я файлу, а де посилання. Якщо ви видаляєте один із цих файлів (точніше одне із цих імен), то файл ще зберігається на диску (поки в нього є хоч одне ім'я-посилання).

Дуже важко розрізнити первісне ім'я файлу й пізніше створені жорсткі посилання на нього. Тому жорсткі посилання застосовуються там, де відслідковувати відмінності й не потрібно. Одне із застосувань жорстких посилань полягає в тому, щоб запобігти можливості випадкового видалення файлу.

Особливістю жорстких посилань є те, що вони прямо вказують на номер індексного дескриптора, а, отже, такі імена можуть вказувати тільки на файли усередині тієї ж самої файлової системи (тобто, на тому ж самому носії, на якому перебуває каталог, що містить це ім'я).

Але в Linux є інший тип посилань, так звані символічні посилання. Ці посилання теж можуть розглядатися як додаткові імена файлів, але в той же час вони представляються окремими файлами — файлами типу символічних посилань. На відміну від жорстких посилань символічні посилання можуть вказувати на файли, розташовані в іншій файлової системі, наприклад, на змонтованому носії, або навіть на іншому комп'ютері. Якщо вихідний файл віддалений, символічне посилання не видаляється, але стає не потрібним. Використовуйте символічні

посилання в тих випадках, коли прагнете уникнути плутанини, пов'язаної із застосуванням жорстких посилань.

3.4 ПАКЕТИ ПРИКЛАДНИХ ПРОГРАМ

Основне призначення прикладних програм – це розв'язання задач у конкретній предметній галузі.

При розв'язанні задач за допомогою комп'ютера її розбивають на кілька частин – підзадач. Кожна з цих під задач розв'язується за допомогою своєї прикладної програми. Розв'язання вихідної задачі забезпечується сукупністю всіх використовуваних прикладних програм. Для розв'язування задач однакового типу створено системи прикладних програм, за допомогою яких розв'язуються різні конкретні задачі даного типу. Системи прикладних програм, що дозволяють розв'язувати задачі певного типу, називаються пакетами прикладних програм.

За типом розв'язуваних задач серед наявних нині прикладних програм виділяються такі основні групи:

- текстові редактори і текстові процеси
- електронні таблиці
- бази даних
- графічні пакети
- системи штучного інтелекту й експертні системи
- навчальні програми
- системи мультимедіа, комп'ютерні ігри та розваги.

У кожній із зазначених груп є багато прикладних програм, що різняться особливостями і можливостями. Однією з умов ефективного впровадження обчислювальної техніки в практику є створення спеціалізованих пакетів прикладних програм (ППП). Доступність і простота використання їх створює передумови більш широкого впровадження ЕОМ в інженерну роботу, вирішення конкретних завдань наукової області, економіки, культури, освіти. Існуючі ППП охоплюють майже всі сфери людської діяльності пов'язаної з обробкою інформації. ППП зазвичай будуються на базі спеціальних систем і є подальшим їх розвитком в конкретному напрямку. Вони поставляються окремо від програмного забезпечення обчислювальних засобів, мають свою документацію і не входять до складу операційних систем. ППП в даний час стали невід'ємною частиною програмного забезпечення персональних комп'ютерів. Це пояснюється великим набором функцій для роботи з даними, простотою освоєння і роботи. В даний час, коли користувач все більше звертає увагу на оперативність, наочність наданої інформації, а для інженерно - технічних працівників все важливішим стає обробка та зберігання великих обсягів даних, відіграють велику роль такі функції табличного процесора, як складання списків, зведених таблиць, можливість використання формул, копіювання даних, форматування та оформлення, аналіз і надання даних за допомогою діаграм і зведених таблиць, вилучення інформації із зовнішніх баз даних, забезпечення безпеки. Цим же пояснюється широке застосування їх в різних областях діяльності людини. Вони є невід'ємною частиною інформаційних систем, які полегшують доступ користувача до інформації з практично будь-якої області науки, техніки, культури, охорони здоров'я, навчання. Постійне вдосконалення

наявних пакетів електронних таблиць, поява нових роблять роботу з даними все більш і більш простої, доступної для користувача будь-якої кваліфікації. Останнім часом стало можливим в табличних процесорах створення гіпертекстових посилань в таблиці з метою переходу в інші файли, що знаходяться на комп'ютері користувача, в локальній мережі або в мережі Internet. Дані та діаграми можна зберегти як окрему Web - сторінку або додавати до існуючої сторінці.

Використання пакета прикладних програм потребує наявності певного набору пристроїв у апаратної частини, певного об'єму оперативної пам'яті комп'ютера, певної операційної системи. Під час вибору пакета прикладних програм для роботи користувач мусить враховувати можливості свого конкретного комп'ютера.

Основна особливість програмного забезпечення Linux - різноманіття продуктів, які вирішують подібні завдання, особливо якщо справа стосується області, в якій існує кілька підходів до їх вирішення. Відкрита модель розробки програм, дозволяє будь-якому вибрати найбільш підходящий для нього інструмент і розвивати саме його. Тому список проектів, так чи інакше пов'язаних з Linux, налічує десятки (або навіть сотні) тисяч найменувань.

Звичайно ж, робота з самою операційною системою не може бути самоціллю. Всі зусилля з вивчення операційної системи Linux та основних утиліт потрібні для того, щоб згодом найкращим чином вирішувати в цій операційній системі будь-які з своїх прикладних задач, вирішуваних за допомогою комп'ютера. Для дуже багатьох завдань досить стандартних інструментів Linux та текстового редактора, однак є випадки, в яких все-таки необхідна спеціальна прикладна програма, саме для цього призначена, або в яких спеціальна програма зручніше комбінації стандартних утиліт.

Цей розділ присвячений короткому огляду прикладних програм для Linux, спеціально призначених для вирішення найрізноманітніших користувача завдань. Увійшовши сюди матеріал потрібно сприймати тільки як приклад, демонстрацію того, що і як можна робити в Linux, але зовсім не вичерпний список. На відміну від основних принципів устрою системи або стандартних утиліт, які не змінюються (майже) протягом десятиліть, прикладне програмне забезпечення - це область, де все змінюється дуже швидко. Технології, сьогодні вважаються найбільш передовими, вже через кілька місяців можуть застаріти. Разом з ними можуть застаріти використовують їх програми, а інші програми можуть, навпаки, перейти в розряд найбільш сучасних і розвинених. Тому перелічені тут прикладні програми - це не безумовна рекомендація, а досить випадкова вибірка, що відображає поточний стан справ у розробці додатків для Linux. Найкращий спосіб знайти і вибрати самі відповідні прикладні програми для своїх задач - порадитися з людьми, які вирішують подібні завдання в Linux в даний час - і спробувати.

Потрібно віддавати собі звіт в тому, що прикладні програми для Linux не є частиною самої Linux, тому будь-який з названих нижче програм може не виявитися в якомусь з конкретних дистрибутивів Linux. Але майже напевно в будь-якому дистрибутиві знайдеться не менше однієї або декількох програм для вирішення кожної з перерахованих нижче прикладних задач. Щоб не захаращувати виклад, ми зупинимося лише на найбільш поширених програмних продуктах, що входять в багато дистрибутиви Linux.

Робочий стіл

Перше, що варто зробити, почавши постійно використовувати Linux - організувати для себе зручне «робоче місце»: підібрати і налаштувати програми, з якими доводиться працювати щодня. Робоче місце в Linux може виглядати дуже по-різному. Можна взагалі обійтися без графічного інтерфейсу, використовуючи тільки текстовий термінал для управління системою. Такий вибір буде правильним, якщо робоче місце знаходиться на сервері, підключеному до мережі Internet, доступ до якого здійснюється тільки за допомогою ssh або аналогічних клієнтів віддаленого доступу. Втім, деякі користувачі воліють працювати в текстовому інтерфейсі, можливо, по ергономічним причин - ніщо не відволікає? Якщо графічний інтерфейс використовується, то й у цьому випадку є величезний вибір, як його організувати. Перш за все, вирішити: чи потрібно влаштувати «робочий стіл» (для цього підходять GNOME, KDE, XFCE) або можна обійтися можливостями однієї з розвинених диспетчерів вікон (Enlightenment, FVWM2, WindowMaker і багато інших). Крім функціональності, у виборі графічного середовища вирішальне значення можуть зіграти і естетичні критерії. Далі за всіх в цьому напрямку просунулася середу Enlightenment, робота з якою в деяких варіантах настройки кількістю прикрас та ефектів нагадує участь у комп'ютерній грі (швидше за все, мережевий).

Диспетчери файлів

Багато користувачів звикли оперувати з файлами і каталогами як з наочними штучними об'єктами (папками і документами), вони можуть обрати для себе програму, яка дозволяє наочно і поштучно працювати з об'єктами файлової системи - диспетчер файлів (file manager). Оскільки подання файлів і каталогів як папок і документів потрібно в першу чергу в рамках метафори робочого столу, то і диспетчери файлів для Linux розробляються насамперед як додатки тій чи іншій середовища робочого столу. Зокрема, і в KDE, і в GNOME є свої диспетчери файлів - konqueror і nautilus відповідно, які за сумісництвом служать www-браузерами. Таке поєднання функцій цілком логічно, оскільки в середовищі робочого столу потрібно представляти доступні локальні і віддалені ресурси як єдиний простір, наповнений об'єктами, якими можна маніпулювати, можна «відкривати», тобто запускати відповідну програму для перегляду та / або редагування.

Для багатьох користувачів найбільш зручний спосіб роботи з файловою системою - «класичний» двохпанельний диспетчер файлів, що працює в текстовому режимі (у терміналі) - Midnight Commander (назва утиліти - mc) 1 . Його функціональність також ширше просто операцій з файлами - він дозволяє відкривати файли для перегляду і редагування, викликати допоміжні програми для роботи з архівами (і навіть «заходити» в архіви, як у каталоги), передавати дані по мережі і т. п. Midnight Commander має також непоганий вбудований текстовий редактор, знову-таки «класичного» стилю.

Далеко не все, що потрібно робити в Linux, в середовищі mc так само зручно, як і у повноцінній командному рядку. Крім того, при роботі з графічними файлами сильно не вистачає подання цих файлів у вигляді мініатюр (thumbnails), щоб вибрати серед них по вмісту, а не тільки по імені. Такими можливостями мають численні графічні диспетчери файлів; крім тих, що включені до середовища KDE та GNOME, є безліч незалежних: dfm (схожий на диспетчер файлів OS / 2), emelfm2,

EZFM і X Northern Captain (двохпанельний, причому автор останнього - наша людина з Дубни), gentoo і worker (двохпанельний, в стилі диспетчера файлів DirectoryOpus з AmigaOS), FSV і XCruiser (трехмерние! причому останній схожий швидше на космічний симулятор). Серед них зустрічаються і орієнтовані спеціально на перегляд зображень, такі як GQView, endeavour, gvview, qiv, xzgv і деякі інші, - з можливостями слайд-шоу, автоматичного зміни розміру, показу картинки на повний екран і т. п.

Емулятор терміналу

Навіть для такої на перший погляд тривіальної функції, як емуляція терміналу для X Window System, існує ціле коло програм. Сама стандартна з них поставляється разом з XOrg - xterm . Варіант xterm , що підтримує відображення шрифтів в кодуванні UNICODE, викликається командою uxterm . Однак кожен додаток, що організує сферу робочого столу, включає власний емулятор терміналу, зовнішній вигляд і поведінка якого налаштовується централізовано разом з усіма іншими додатками робочого столу. Є й інші емулятори терміналу, не пов'язані з конкретним робочим столом, до таких відноситься 9term , що повторює можливості «вікна» системи Plan9, mlterm , що має багатомовну підтримку, gxvt - дуже невимогливий до ресурсів емулятор терміналу, або його нащадки, на зразок aterm .

Велика кількість термінальних вікон на робочому столі може утворитися, навіть якщо розкласти їх по різних віртуальних екранів. Деякі версії xterm (наприклад, konsole), дозволяють відкривати вікна «стопками», перемикаючись між ними за допомогою «закладок», як в записній книжці.

Мережа WWW-броузери

WWW-броузер - програма для перегляду гіпертексту, доступного через Internet - на сьогодні чи не найважливіше додаток для персонального комп'ютера. Сьогоднішній www-броузер повинен «уміти» набагато більше, ніж просто відображати сторінки HTML і переходити за гіперпосиланнями. Фактично, на нього лягає завдання роботи даними Internet у всьому їх різноманітті, сюди входить і підтримка постійно розвиваються стандартів, і забезпечення безпеки, і багато іншого.

У Linux є досить великий вибір www-броузерів, однак першим дійсно сучасним вільним додатком для роботи з Internet стала Mozilla, а потім її нащадки, які сьогодні цілком успішно конкурують з аналогічними комерційними програмами. Mozilla - це цілий пакет програм для роботи з Інтернетом: потужний, насичений функціями комунікаційний центр для персонального комп'ютера. До складу пакету входить броузер, програма для роботи з електронною поштою і редактор www-сторінок. Історія Mozilla почалася в 1998 році, коли фірма Netscape опублікувала вихідні тексти свого броузера Netscape Navigator. Одне з важливих властивостей пакета Mozilla - його принципова розширюваність. У Mozilla реалізований мову XUL на основі XML, за допомогою якого дуже легко розробляти додаткові компоненти Mozilla, орієнтовані на виконання спеціальних функцій.

Більш сучасний FireFox, що розробляється командою Mozilla на основі вихідних кодів, що відповідають тільки WWW-броузеру. Інша частина FireFox написана повністю на XUL, тому розробка цієї молоді програми йде істотно бадьоріше і простіше, її налаштування вважається самою гнучкою серед www-

броузерів, а головне, будь-хто може написати на високорівневих мовах програмування XUL / JavaScript і опублікувати свій модуль розширення (т. н. plugin; на сьогодні таких модулів відомо більше півтораста).

Про www-броузерах, розроблених спеціально для тієї або іншої середовища робочого столу, вже йшла мова вище (вони чудово справляються з роллю файлових диспетчерів). Важлива різновид www-броузерів - текстові броузери, тобто ті, які можуть бути запущені в будь-якому текстовому терміналі Linux. Самий старий і відомий з них, один із прототипів сучасних www-броузерів - Lynx. Він не має можливості відображати графічну інформацію, але відмінно підтримує HTML, форми і таблиці. Сучасні версії підтримують також з'єднання, захищені за допомогою SSL. Links - це текстовий броузер, на перший погляд дуже схожий на Lynx, але все ж таки дещо відрізняється від нього: умінням працювати з таблицями і фреймами; відображенням кольорів, зазначених в HTML-сторінці; використанням випадаючих меню (як в Midnight Commander); можливістю завантаження файлів у фоновому режимі.

Крім можливості перегляду WWW-сторінок часто випадає необхідність їх «скачування», тобто записи у файл. Це ж відноситься і до ресурсів, доступним по протоколу FTP. Всі описані вище броузери здатні записувати HTTP-і FTP-ресурс в файл, але для зручної роботи з командного рядка вони, як правило, непридатні. Окрім старої і вельми простої утиліти ftp, є два її потужних розширення: lftp і wget. Обидві утиліти підтримують як FTP, так і HTTP, причому lftp може працювати, як і ftp, в режимі «оболонки», а wget призначена саме для роботи з командного рядка. Якщо при отриманні файлів з якогось сервера або групи серверів необхідно описувати безліч винятків (щоб не скачати зайвого), виконувати якісь дії (наприклад, заповнювати форми або виконувати java-сценарії), можна скористатися більш складними програмами ravuk або htrack.

Поштові програми

Приблизно так, як Firefox відроджує WWW-іпостась Mozilla, Thunderbird повторює - і розширює - поштову складову Mozilla. Більшість сказаного про Thunderbird на сьогодні справедливо і для MozillaMail. Так само, як і в Firefox, у Thunderbird використовується частина вихідного коду Mozilla, яка працює з мережею (на цей раз - з відсиленням пошти та доступом до поштових скриньок), а інтерфейс і архітектура програми в цілому - перероблені для того, щоб позбутися старожитніх частин Netscape і полегшити подальшу розробку. Thunderbird (як і MozillaMail) володіє найпотужнішим на сьогоднішній день вбудованим антиспам-фільтром. Якщо непрохана пошта все-таки потрапляє у вашу поштову скриньку, просто покажіть її Thunderbird зі словами «це - спам!». Через деякий час програма сама навчиться відрізняти непрохану пошту від корисною. Як і Firefox, Thunderbird легко розширювати власними модулями, написаними на високорівневих мовах, і можна дуже гнучко налаштовувати. Ще один поштовий клієнт, декілька поступається Thunderbird по можливостях, але переважаючий його за швидкістю, називається Sylpheed. Інтерфейс цієї програми вельми схожий на стандартну поштову програму для Windows, Outlook Express, що може допомогти уникнути зайвих клопотів при зміні операційної системи. Автор цієї програми, Хіроюкі Ямамото, людина акуратний і пунктуальний, так що деякий брак можливостей (ця програма вміє стільки ж, скільки і Outlook Express) компенсується безвідмовною

роботою і гнучкою системою інтеграції з іншими утилітами системи (антивірусом, антиспам-фільтром і т. п.). Крім того, існує гілка Sylpheed, звана Sylpheed-Claws, в якій проходять перевірку всі нововведення. Стабільна версія Sylpheed-Claws працює нітрохи не гірше авторської Sylpheed, а можливостей у неї більше.

Оскільки управління електронним листуванням - одне із завдань робочого стола, в кожній середовищі робочого столу є свій власний поштовий клієнт. Поштовий клієнт для KDE називається KMail, він підтримує як локальну доставку пошти, так і безліч поштових протоколів (POP3, IMAP, SMTP). Поштовий клієнт для GNOME називається Evolution, він інтегрований з календарем, адресною книгою і претендує на функції індивідуальної «записної книжки».

Електронне листування сама по собі не вимагає графічного інтерфейсу, для читання та написання електронних листів цілком достатньо можливостей терміналу і текстового редактора. Серед текстових поштових клієнтів для Linux найбільш відомі Mutt і Pine, обидва дуже функціональні, підтримують безліч протоколів і форматів поштових скриньок, добре налаштовуються. Вимогливим користувачам, які хочуть мати можливість змінювати зовнішній вигляд і спосіб роботи поштового клієнта, доповнювати його сценаріями і отримувати від поштових служб все, що ті можуть дати, рекомендується Mutt. Тим же, кому головне - просто отримувати, читати і відправляти пошту (з усіма належними зручностями), варто почати з Pine. Любителі Emacs використовують вбудований в нього модуль GNUS, дуже багатий функціями.

Обмін повідомленнями

Якщо комп'ютер підключений до Internet постійно, буває зручно користуватися службами, які передають повідомлення в реальному часі (instant messaging service). Таких служб досить багато, найпопулярніша з них - ICQ. Множинність пояснюється тим, що в більшості випадків цей сервіс надається централізовано, якоїсь великої корпорацією. У багатьох випадках сервери цих служб не доступні під вільною ліцензією. Виняток в ряду «власників» - служба Jabber, заснована на повністю відкритому протоколі XMPP. Jabber дозволяє будь-якому спільноті створювати власні сервера, керовані власними адміністраторами. Сам Jabber-сервер має можливість з'єднувати своїх клієнтів не тільки з іншими Jabber-серверами, а й зі службами ICQ, MSN, Yahoo і AIM. У Linux є кілька клієнтських програм для обміну миттєвими повідомленнями. Окремо стоять клієнти IRC (Internet Relay Chat), служби з більш довгою історією і складним протоколом (мається на увазі і мережевий протокол, і протокол роботи користувача в IRC).

Psi - зручний графічний клієнт мережі швидкого обміну повідомленнями Jabber (а значить, по всіх протоколах, які підтримує обраний Jabber-сервер). Psi підтримує такі можливості Jabber, як одночасна робота з декількома серверами, конференції, криптозахист переданої інформації (через SSL і GnuPG), роботу через HTTP(S) проксі-сервер і т. д. SIM - багатопротокольний клієнт обміну миттєвими повідомленнями. Підтримуються протоколи ICQ, Jabber, MSN, AIM, YIM, а також LiveJournal. Крім того, є безліч модулів, що реалізують додаткові можливості. Є варіант SIM, орієнтований на середу KDE. «Прицільно» на середу KDE орієнтований і інший потужний клієнт, який має підтримку також і IRC, - Kopete. На середу Gnome орієнтований Gaim - найбільш потужний і найбільш гнучко

настроюється клієнт. Має модулі доступу майже до всіх мислимих протоколам, дозволяє писати сценарії на Perl і TCL. Для IRC є і спеціальні клієнти: ChatZilla (як можна здогадатися з назви, він «вбудований» в Mozilla, але доступний і як доповнення до Firefox) або X-Chat - вельми потужна програма, орієнтована на «хитрощі» IRC. Не обійдений стороною і інтерфейс текстової консолі: CenterICQ, що підтримує кілька протоколів (серед них Jabber і IRC); liscq, що володіє як текстових, так і графічним інтерфейсами (отже, їм можна користуватися і перебуваючи за робочою станцією, і дистанційно); irssi, націлений на служби типу IRC (на сьогоднішній день підтримуються IRC, SILC і ICB), і т. д.

Офісні програми

Важливою частиною сучасної робочої станції є так звані офісні засоби обробки інформації. Під офісними додатками звичайно розуміють стандартний набір з текстового процесора, засоби роботи з електронними таблицями, засоби створення презентацій, засоби для роботи з базами даних. Всі перераховані офісні програми входять в пакет OpenOffice.org - це вільний набір офісних програм, що не поступається за можливостями невіЛЬНОГО Microsoft Office, а де в чому навіть перевершує його. Наприклад, випадковість, яка може мати дуже важливе значення: компонент OpenOffice.org OpenWriter дозволяє експортувати документи безпосередньо в формат PDF. Інтерфейс OpenOffice.org влаштований принципово так само, як і в аналогічних продуктів Microsoft, так що користувачеві, який звик до Microsoft Office, не складе великих труднощів перейти до роботи в OpenOffice.org. Крім того, OpenOffice.org дозволяє працювати з усіма форматами файлів Microsoft Office.

Історія OpenOffice.org нагадує історію Mozilla: спочатку проект (під ім'ям StarOffice) розвивався закрито, без доступу світового співтовариства програміста до вихідних текстів. Однак у 2000-му році компанія Sun Microsystems відкрила вихідні тексти програмного продукту, утворивши OpenOffice.org. Так само, як і у випадку Netscape / Mozilla, пара StarOffice / OpenOffice.org використовує подвійне ліцензування, що дає право як вільного доступу до вихідних текстів, так і використання їх в закритих комерційних продуктах.

За можливостями OpenOffice.org залишається найбільш розвиненим і повним офісним пакетом для Linux, проте є і інші офісні кошти. Зокрема, офісний пакет Koffice, орієнтований на середу KDE, в якому є приблизно той же набір офісних додатків, що і в OpenOffice.org. Крім того, є окремі офісні додатки, що не становлять пакетів - словниковий процесор Abiword та електронні таблиці GNUMeric.

Графіка

Чим простіше для користувача завдання, тим більше програм під Linux її вирішують. Зокрема, маніпуляція геометричними фігурами з можливістю зміни їх параметрів (кольору, розміру і т. п.), зберіганням набору фігур у файлі і перетворенням отриманого зображення в растровий формат - досить проста задача, що вимагає акуратної реалізації основних функцій якийсь високорівневої бібліотеки (або двох - інтерфейсної та графічної). Не дивно, що редактори з подібними можливостями є і для кожного робочого столу, і незалежно від них. Це твердження стосується і ще більш простим програмами роботи з растровою графікою. Нижче описані тільки істотно більш складні програми.

Векторна графіка

Векторна графіка називається спосіб роботи з зображеннями, при якому воно представлено у вигляді фігур, кожна з яких має власний опис (тип, розміри, кривизну чи інші параметри складових частин, їх кольору, спосіб представлення і т. п.). Деякі графічні пристрої (наприклад, що розпізнають формат PostScript) вміють самі інтерпретувати опису фігур, для інших необхідно заздалегідь прорахувати і сформувати картинку програмним шляхом.

Робота з PostScript та PDF

Сучасна поліграфія вже не мислиться у відриві від комп'ютерів, все додрукарські матеріали зазвичай існують в електронній формі, і саме електронні документи подаються на друкуючі пристрої для виводу. Причому для сучасної поліграфії de facto стандартом є формат PostScript. PostScript - це мова опису сторінки, що дозволяє представити будь-які поліграфічні матеріали у векторному форматі (однак він допускає і включення растрових фрагментів). Файл у форматі PostScript фактично являє собою програму, яка описує, які дії потрібно зробити, щоб отримати необхідний висновок. Професійні друкуючі пристрої вміють самостійно інтерпретувати документи на мові PostScript.

PDF (P ortable D ocument F ormat, стерпний формат документів) створений на основі мови PostScript. Його основне завдання - забезпечити однаковий зовнішній вигляд документа в будь-якій операційній системі. У PDF є спеціальні можливості для публікації документів в Мережі, зокрема, підтримка гіперпосилань, а деякі можливості мови PostScript звідти, навпаки, виключені.

Ghostscript - інтерпретатор мови опису сторінок PostScript і файлів у форматі PDF (формат переносимих документів). Ghostscript перетворює PostScript в багато растрових форматів, які підходять для виведення даних на екран або на принтер, який не підтримує PostScript. Зазвичай Ghostscript використовується для перегляду файлів PostScript і для друку на принтерах, які не підтримують мову PostScript, GhostScript використовується безліччю додатків для виведення даних на друк. Графічний інтерфейс для GhostScript надає програма GhostView (команда gv), вона дозволяє відображати документи у форматах PostScript та PDF у графічному середовищі X Window System. Для різних маніпуляцій з файлами у форматі PostScript призначений пакет утиліт командного рядка psutils , з їх допомогою можна вибрати, змінювати порядок, масштабувати сторінки в PostScript-файлах, змінювати параметри тексту і робити багато чого іншого.

Спеціально для перегляду PDF-файлів призначена програма xpdf, вона дозволяє переходити за гіперпосиланнями в документі, переглядати структуру документа, проводити пошук і підтримує згладжування шрифтів. На основі вихідних текстів xpdf створено бібліотеку poppler, призначена для відтворення PDF, яка стала основою інших популярних програм для перегляду PDF: kpdf (компонент графічного середовища KDE, відрізняється більш багатими інтерфейсними можливостями) і Evince. Багато дистрибутиви Linux включають Acroread - версію відомого програми Adobe Acrobat для Linux, проте, на відміну від названих вище, воно є невірним програмним продуктом.

Тривимірна графіка

Для Linux створено кілька програмних пакетів, що працюють з просторовими поданням об'єктів.

Вихідні тексти одного з найпотужніших пакетів тривимірного моделювання, перерахунку (рендеринга) та анімації - Blender - в 2002 році були відкриті і весь проект повністю переведений під вільну ліцензію. Автори Blender прийшли до висновку, що відкрита розробка інструменту більш ефективна і прибуткова для тих, хто цим інструментом (а не його продажем) заробляє. Для цього довелося викупити знаходяться у власності спонсорів частини проекту у господарів за суму сто тисяч євро. Шукану суму надало спільнота користувачів Blender, вже тоді чимале: кожен вніс скільки зміг, і менш ніж за два місяці грошей на рахунку виявилось достатньо. З тих пір коло користувачів і можливості Blender продовжують постійно зростати. Для виконання завдань, що суміщаються Blender, є й окремі програмні засоби. Наприклад, популярний пакет трасування променів (тривимірного проектування та сценографії) POV-Ray, за допомогою якого створюються проекти дивовижною складності і краси (наприклад, переклад картини Вільяма Марлоу «Капріччіо» в тривимірне уявлення - з тим тільки, щоб з певної точки повторити її) . Багато графічні редактори мають вбудовані засоби анімації, а інші (як, наприклад, CinePaint, що називався раніше FilmGimp) спеціально розробляються для покадрової обробки відео.

Не варто забувати, що популярний нині стандарт OpenGL - відкритий, він розроблявся для UNIX-подібних систем і використовується великою кількістю програм для Linux (у тому числі і Blender). На жаль, виробники апаратного забезпечення (відеокарт), як правило, приховують не тільки пристрій своїх карт, але навіть і спосіб їх низкоуровневого використання. Тому у відкритому доступі виявляються лише готові драйвери (без вихідних текстів) до деякими версіями ядра Linux і певним зборках XOrg. Відображення тривимірних об'єктів з перерахунком на програмному рівні поки працює істотно повільніше, хоча нітрохи не гірше, тому використовуючи OpenGL для ігор та інших програм, що вимагають дійсно швидкої роботи графічної підсистеми, потрібно завжди пам'ятати про необхідність отримати - можливо, невірний - драйвер.

Мультимедіа.Музичні скриньки

Програм-програвачів звукових файлів в Linux не перелічити. Очевидний лідер за популярністю серед них - XMMS (X M ulti M edia S ystem). Крім основної функції - грати музику (підтримується безліч форматів) - в ньому реалізовано чимало звукових та візуальних ефектів завдяки великій кількості розширень. Інтерфейс XMMS аналогічний інтерфейсу не менш популярного в системах Windows програми WinAMP (до речі, XMMS вміє використовувати «шкурки» WinAMP2). Майже не поступається XMMS його «брат» BEEP, що використовує графічну бібліотеку GTK2, а не GTK. Є й інші програми, які нітрохи не гірше цих грають музику. Зазвичай кожна середа робочого столу реалізує власний програвач звукових файлів, хоча б для того, щоб відтворювати власні звукові ефекти, пов'язані з різними системними подіями, проте з їх допомогою прослуховувати файли може і користувач.

Очевидно, що для прослуховування звуку зовсім не обов'язково використовувати графічний інтерфейс, тому в Linux є велика кількість термінальних

утиліт для відтворення звуку. Деякі з них, наприклад, mpg123 , mpg321 , ogg123 або splay , призначені для програвання оцифрованого звуку (можливо, в стислих форматах), інші, такі як lazy або cd-console , управляють музичними лазерними дисками, є утиліти, що грають музику в нотному (midi) та інших форматах - timidity (вона відрізняється тим, що перетворює ноти, записані для інструментів в оцифроване звучання цих інструментів, а значить, не вимагає MIDI-пристрої), mikmod (розпізнає безліч форматів: MOD, STM, S3M, XM і т . д.), sidplay та інші. Щоб користувач не заплутався, спеціальні оболонки, наприклад mpfc або cplay , надають загальний інтерфейс до всіх консольним програвачів.

Музичні редактори

Частина професійних музикантів воліє використовувати для запису і зведення багатоканального звуку дорогі спеціалізовані цифрові станції: у цьому винна і реклама, і низька, з точки зору професіонала, якість звуку більшості звукових карт в комп'ютерах загального призначення. Незважаючи на це і для таких комп'ютерів існує чимало програм, що працюють зі звуком на професійному рівні. Такі програми можна розділити на дві категорії: нотні редактори, завдання яких - створення, редагування, запис і нотне представлення музичних композицій, і звукові редактори для власне звуку, а також перетворення його, накладення ефектів і т. п.

Редактори й фільтри оцифрованого звуку

Популярний вільний редактор звуку - Audacity. Він уміє записувати звук відразу в формати WAV, AIFF, AU, IRCAM або MP3. У ньому є всілякі інструменти для редагування записаного звуку, в тому числі вбудований редактор амплітуди, що настроюється режим відображення спектрограми і засоби частотного аналізу звукових доріжок. Вбудовані найпростіші аудіоефекти включають посилення баса, WahWah, видалення шуму і т. д. Audacity підтримує модульні додатки, в яких зазвичай поставляються більш складні аудіоефекти. У список підтримуваних форматів модулів входять VST, LADSPA і Nyquist.

Sweep - це багатоканальний звуковий редактор, в якому реалізовані всі основні операції, такі як видалення, копіювання, вставка і застосування ефектів, оформлених у вигляді плагінів, до будь-якої частини звукового файлу. Приблизно тими ж можливостями володіють і інші редактори звуку - Rezound, WaveSurfer і GNUSound.

Як і у випадку з іншими мультимедіа-форматами, в Linux існують термінальні утиліти для обробки звуку, які не потребують графічного інтерфейсу. Основний пакет термінальних утиліт для роботи зі звуком називається SOX, в нього входять утиліти для перетворення, запису та відтворення звукових файлів, підтримується безліч форматів.

За допомогою консольних утиліт можна також стискати звукові файли в різні формати з втратою якості. Вміст файлу, стисненого «з втратою якості», може бути неотличимо на слух від вмісту вихідного файлу: алгоритми перетворення враховують людську фізіологію, наприклад, формат MP3 не відтворює дуже високих звуків, а занадто низькі не поділяє на два канали. Сенс терміна «втрата якості» - в тому, що з упакованого файлу вихідний відновити вже не можна. Стиснення з втратою якості можна налаштовувати на певну потужність потоку упакованих даних: чим більше даних можна передавати в одиницю часу, тим

чистіше звук, тому такі формати підходять для передачі по мережі (наприклад, інтернет-радіо).

Основні формати з втратою якості - це MP3 (з ним працюють пакувальники lame / toolame, bladeenc) і OGG Vorbis (утиліта oggenc). Ці формати (особливо OGG) добре підходять для упаковки якісної музики. Файли у форматі OGG, упаковані семикратно (192 kbp / s), майже не відрізняються на слух від вихідних. Якщо необхідно стиснути звук з відчутною втратою якості (але без втрати членороздільності і раз на двадцять), можна використовувати інші формати - gsm, aiff, adpcm, speex (стиснення мови) або bonk. Останній формат - нестандартний, він підтримується однойменної утилітою і відрізняється великою гнучкістю, так як може працювати і в режимі «стиснення без втрат». Для стиснення без втрат розроблений спеціальний формат - FLAC, його розпізнають багато програм (у тому числі і утиліти від авторів цього формату - flac і metaflac) і навіть аудіопристрої.

Відеопрогравачі

Найбільш повнофункціональним і зручним «домашнім кінотеатром» для Linux є програма xine. Xine підтримує mpeg-2 і mpeg-1 (включаючи DVD) потоки, MPEG-4 і інші формати. Альтернативний йому універсальний програвач - MPlayer. Існує додаток для програвання відеопотоку, одержуваного по Мережі - VideoLAN (vlc), яке працює з форматами MPEG1, MPEG2, MPEG4 (також відомий як DivX) і DVD. Hawtv - програма для перегляду і запису відеопотоків Video4Linux, то є програма для перегляду ТБ. Hawtv використовує набір графічних елементів Athena. Може використовуватися разом з VDR для перегляду цифрового супутникового, кабельного та ефірного ТБ формату DVB.

3.5 Дистрибутиви

Дистрибутив Лінукс, часто просто дистрибутив або навіть дистро — це член сімейства Лінукс з Юнікс-подібних операційних систем, що складається з ядра Лінукс, системних бібліотек та інструментів, розроблених проектом GNU та інших програм. Існує величезна кількість різноманітних дистрибутивів Лінукс від повноцінних стільничних та серверних операційних систем до мінімальних середовищ (що як правило використовуються у вбудованих системах чи мають спеціалізоване призначення). Щоб забезпечити Юнікс-подібне середовище дистрибутиви Лінукс містять набір Юнікс-подібних інструментів та бібліотек. У повнофункціональних дистрибутивах він як правило береться з проекту GNU. Дистрибутиви, орієнтовані на мінімальний розмір, користуються компактнішими альтернативами, наприклад, busybox та uclibc. Типовий дистрибутив, призначений для запуску як офісна чи розважальна система, складається з ядра Лінукс, інструментів та бібліотек проекту GNU, додаткових програм, документації, графічної системи, віконного менеджера та стільничного середовища. Більшість з включених програм є вільними та/або програмами з відкритим кодом, що поширюються супровідниками як у двійковій формі, так і у вигляді джерельних текстів, що дає можливість користувачам при потребі самостійно змінювати джерельний код. Частина включених програм, проте, можуть бути власницькими, без доступних джерельних кодів.

Debian та Ubuntu

Debian

Проект Debian - це асоціація людей, спільною справою яких є створення вільної операційної системи. Ця операційна система називається Debian GNU / Linux або просто Debian.

Операційна система - це набір базових програм, які забезпечують працездатність вашого комп'ютера. Основою операційної системи є ядро. Ядро виконує всю основну роботу і дає можливість запускати інші програми.

Debian використовує ядро Linux. Linux почав розробляти Лінус Торвальдс (Linus Torvalds), і яке підтримують тисячі програмістів по всьому світу.

Тим не менш, ведеться робота по створенню систем Debian на інших ядрах, в першу чергу, на the Hurd. Hurd - це набір серверів, які працюють під керуванням мікроядра (такого, як Mach) та реалізують різні можливості. Hurd - це вільне програмне забезпечення, створене проектом GNU (GNU Project).

Більша частина основних інструментів, які наповнюють операційну систему, була взята з (Проекту GNU (GNU project)); про що і говорять назви: GNU / Linux та GNU / Hurd. Ці інструменти також є вільним.

Зрозуміло, в першу чергу, люди хочуть мати прикладне програмне забезпечення, тобто програми, які допомагають їм робити те, що їм хочеться: від редагування документів до керування бізнесом, від іграшок до створення нового програмного забезпечення. Debian містить понад 18733 пакунків (завчасно скомпільованого програмного забезпечення в зручному для встановлення на вашому комп'ютері форматі) - і всі вони є вільними.

У деякому роді, все це нагадує вежу. В основі знаходиться - ядро. На ньому всі основні інструменти. Далі йде все програмне забезпечення, яке ви використовуєте на комп'ютері. На самій вершині башти - Debian - ретельно організовуючи та складаючи все в єдине ціле так, щоб воно працювало разом.

Проект Debian розпочав в серпні 1993 року Ян Мердок (Ian Murdock). Він мав намір створити новий дистрибутив, який буде відкритий у дусі Linux та GNU. Debian задумувався як дистрибутив, який буде зібраний ретельно і сумлінно, і так само якісно буде підтримуватися. Він починався як маленька, тісно зв'язана група хакерів вільного ПЗ, а потім поступово зростав, щоб стати великою, добре організованою спільнотою розробників і користувачів.

Перші версії 0.9x були випущені у 1994 та 1995 роках. Перший випуск 1.x було здійснено у 1996 році. Тоді ж Брюс Перенс замінив Іена Мердока на посаді лідера проекту. В цьому ж році один з розробників, Іен Шуйслер запропонував укласти суспільну угоду між проектом та його користувачами. За результатами обговорення у поштовій розсилці було укладено Соціальну Угоду та Посібник по вільних програмах Дебієн, котрі описували базові вимоги до розробників дистрибутиву. Брюс Перенс також ініціював створення офіційної організації «Програми в інтересах суспільства», котра б опікувалась проектом і вирішували фінансові та юридичні питання.

У 1998 році, перед випуском версії 2.0, першої, яка базувалась на бібліотеці glibc, Брюс Перенс покинув проект. Учасники проекту провели вибори нового лідера та випустили два нових релізи гілки 2.x, кожен з яких містив все більше

пакунків та портів на різні архітектури. В цей час було розроблено АРТ та започатковано перший проект на ядрі, відмінному від Лінукс — Дебієн ГНУ/Хард (Debian GNU/Hurd). У 1999 році було запущено перші дистрибутиви на основі Дебієн — Corel Linux та Stormix's Storm Linux, які нині більше не підтримуються. Наприкінці 2000 у Проекті відбулися значні зміни в практиці архівування та випуску стабільної версії. Реорганізовано архіви програм, введено сховища пакунків («package pools») та створено тестову гілку у вигляді тестового майданчику для запланованого випуску. В цьому ж році розробники почали проводити конференції під назвою Дебконф (Debconf), на яких обговорюються задачі для розробників та технічних користувачів.

У версії 4.0 вперше за історію проекту з дистрибутиву було викинуто одну з підтримуваних процесорних архітектур — m68k.

Останній стабільний випуск - це Debian — 6.0. Останніми оновленнями до нього є 6.0.3, 08 жовтня 2011 р.

Слово Debian вимовляється як 'дебіан' або 'дебієн'. Воно походить від імені творця Debian, Яна Мердока і його дружини Дебри (Debra). Debian Debian не личить для тих, хто хоче, не бажаючи знатися на основах системи, робити все за допомогою миші. Але роки роботи багатьох сотень розробників дозволили зробити Debian найпростішим, зрозумілішим, логічнішим в налаштуванні і адмініструванні дистрибутивом. Якщо ви хочете зрозуміти Linux, серйозно вивчити його, не займаючись при цьому безперервним пошуком файлів налаштування, документації, бракуючих пакетів, то Debian - для вас. Даний дистрибутив - повністю некомерційний проект, що базується на ідеалах розробки Linux. Добровільні розробники зі всього світу вносять вклад до загального проекту, який добре організований і має строгі правила, і це гарантує якість дистрибутива Debian. Протягом всього процесу розробки існують 3 вітки - "стабільна", така, що "знаходиться в процесі тестування", і "нестабільна". Коли з'являється нова версія пакету, вона поміщається спочатку в нестабільну категорію. Debian відомий своєю дуже незручною установкою, що вимагає від користувача обширних знань і великих витрат часу. Але цей недолік компенсується наявністю зручного інсталятора пакетів під назвою apt-get. Користувачі Debian вважають, що їх інсталятор такий незручний, тому що він потрібний їм лише один раз. А коли дистрибутив вже встановлений, всі оновлення будь-якого масштабу можуть бути зроблені за допомогою утиліти apt-get. Debian - справжня енциклопедія вільних програм. Ви знайдете в ній все, що потрібне для роботи в найширшому спектрі областей інформаційних технологій. Ubuntu Ubuntu — дистрибутив, що використовує ядро Linux і заснований на дистрибутиві Debian. Основним розробником і спонсором є компанія Canonical. В даний час проект активно розвивається і підтримується вільним співтовариством. Звичайна нова версія дистрибутива виходить кожні 6 місяців і підтримується оновленнями безпеки протягом 18 місяців, а версія LTS (від англ. LongTerm Support — «довгострокова підтримка»), що виходить раз в 2 роки, підтримується протягом 3 років (серверна версія — протягом 5 років). Ubuntu поставляється з підбіркою програмного забезпечення для серверів і робочих станцій. Використовуючи CD-, DVD- або USB флеш-носій, вона встановлюється на настільні персональні комп'ютери за допомогою LIVECD, LIVEUSB або текстового установника.

RedHat та Fedora

Red Hat Enterprise Linux - платформа, яка добре підходить для широкого спектру додатків в ІТ-інфраструктуру. Останній реліз, Red Hat Enterprise Linux 6, являє собою новий стандарт для Red Hat, пропонуючи більше гнучкості, ефективності та контролю. Він працює в широкому спектрі апаратних архітектур. Юридичні особи та установи, які стандартизувати Red Hat Enterprise Linux вільно можуть сконцентруватися на будівництві свого бізнесу, знаючи, що у них є платформа, яка забезпечує більш того, що їм потрібно.

Red Hat Enterprise Linux

Red Hat Enterprise Linux забезпечує підтримку нових та існуючих програм, сучасних програмних архітектур, і нові рамки виконання. Але Red Hat надає набагато більше можливостей, ніж просто універсальний інтерфейс для додатків. Red Hat Enterprise Linux пропонує поширеною основоположні умови для розвитку центрів обробки даних, процедур і послуг. Від управління ідентифікаційними даними і управління, гнучкий і повний стек веб-додатків для ефективного управління десятків серверів і масові зберігання, Red Hat Enterprise Linux повинні бути важливою частиною вашої інфраструктури центрів обробки даних. З початку відкритої операційної системи грає високу роль для підприємства концепцію. Область застосування дістатися з мініатюрних пристроїв за роботою станції до серверних систем на Intel x86, DEC Alpha і Sun SPARC системах. Один з зміцнює Red Hat Linux, є застосування в Інтернет і Інтранет. Широка підтримка, навчання та підготовка пропозицій, а також широка підтримка ІТ-компаній відносять до зростання Red Hat постійно. Red Hat досягла своєї дистрибутива близько 15% від валового доходу, більшість доходів здійснюється компетентними службами, як Redhat мережі - RHN. У форматі RPM пакет програмного забезпечення, Red Hat встановив стандарт, який у багатьох дистрибутивах було.

Проект Fedora є одним з Red Hat затвердженій споживчій версії дистрибутива Linux Red Hat Linux. Це проект з відкритим кодом спонсорується Red Hat, але незалежно один від одного полягає в управлінні спільнотою Linux. У травні 2004 року ядро Fedora 2 для x86-64 і для архітектури i386 було опубліковано, використовувати ядра Linux 2.6, новий X-сервер X.org і розширена захист доступу SELinux в ядрі.

З новою Fedora Linux Core 4, містяться програмного забезпечення зріс на сьогоднішній день. GCC 4.0, гноми 2.10, KDE 3.4, OpenOffice 2.0 бета, середовища розробки Eclipse 3.1 і універсальний переглядач документів Evince 0.2.1 належить дане програмне забезпечення в даний час. Крім x86 32-розрядна та x86 64-розрядна архітектура тепер Fedora також встановлюватися на системах Power PC. Глобальної файлової системи (GFS) у версії 6.1 використовується для кластерних систем, Xen 2 створює віртуальне середовище для гостей систем. Для мінімальної установки 620 Мб до 7 Гб (все для установки) вільного простору необхідно. Fedora Core 5 Linux містить нові графічні інструменти для управління програмним забезпеченням, GNOME 2.14 та KDE 3.5 робочий стіл і стандартний веб-браузер Firefox 1.5. Ядро Linux оновлюється на версії 2.6.16, Apache Server 2.2, X.org X11R7.0, GCC 4.1 і SELinux були переглянуті. Система друку був повністю переглянутий у Fedora Linux Core 6 і на основі CUPS 1.2. Fedora Core 6 будує основи для Red Hat Enterprise Linux 5 (RHEL). Цей реліз без будь-якого комерційного програмного забезпечення, як

Java, підтримка NTFS, спалах, MP3-кодека і DVD-програвач програмне забезпечення для закодованих DVD-дисків. Графічні драйвери ATI і Nvidia від, також пропали без вісті. Програмне забезпечення оновлюється, для установки X-сервер X.org 7.1, GNOME 2.16, KDE 3.5.4, Firefox 1.5.0.6, Thunderbird 1.5.0.5, OpenOffice.org 2.0.4, Apache Web Server 2.2.3, MySQL 5.0 0,22, PHP 5.1.6 і безкоштовно. Чисте реалізації Mono 1.1.17.1 можуть бути вибрані. У лютому 2006 року Red Hat Enterprise Linux (RHEL) 4-й версії операційної системи була сертифікована на безпеку відповідно до Common-Criteria EAL4 +. CentOS, операційна спільноти Enterprise System, заснований на Red Hat Enterprise Linux (RHEL) операційної системи і ліберальної і вільної альтернативою RHEL. Міститься в гнома і KDE робочий стіл. Дистрибутив RedHat Цей дистрибутив вважається класичним і є одним із найпопулярніших і найбільш розповсюджених дистрибутивів. «Червона шапочка» (саме так перекладається назва дистрибутива) є своєрідним прообразом в світі Linux. Не даром же, коли порівнюють дистрибутиви, говорять про їх сумісність з RedHat. Придбати цей дистрибутив можна купивши коробочний варіант або скачати версію безкоштовно з сайту . RedHat веде розробку операційної системи Linux у двох напрямках: для серверних рішень і для клієнтних комп'ютерів. Другий варіант почав все більше здобувати дружній інтерфейс і здатен вирішити будь-які домашні задачі. Найперше, потрібно відмітити простоту установки цього дистрибутива. Програма установки RedHat відрізняється інтуїтивно-зрозумілим інтерфейсом і багатими функціональними можливостями. При цьому, як і для багатьох сучасних дистрибутивів, установка може виконуватися як в графічному, так і в текстовому режимах. До речі, першу графічну програму установки Linux запропонувала саме компанія RedHat. Необхідно помітити, що установка системи в текстовому режимі виконується трошки швидше, ніж в графічному. Після установки ви отримуєте практично функціонуючу систему: все в ній налаштовано і працює, залишається налаштувати систему «під себе». На сьогоднішній день останніми версіями є версії 7.2(7.3) і 8.0. При порівнянні з попередніми версіями програма установки стала ще більш простою, гнучкою і функціональною. Наприклад, з'явилась можливість збереження параметрів установки, що дозволяє встановлювати таку ж конфігурацію системи на інші комп'ютери мережі. При установці здійснюється конвертування файлової системи ext2 в ext3. Перехід на нову файлову систему забезпечує більш надійну її роботу. Серед нововведень можна відмітити віконне середовище Gnome1.4 і файловий менеджер Nautilus, який істотно полегшує операції копіювання, переміщення і видалення файлів. Всі параметри системи повністю налаштовуються. Можна навіть виставити рівень складності інтерфейсу користувача: від новачка до експерта. Також потрібно відмітити, що зникла межа між локальними і мережевими ресурсами. Налаштування пристроїв здійснюється тепер набагато простіше завдяки новій утиліті конфігурування системи. Тепер всі конфігуратори системи зібрані в одній оболонці, яка називається Контрольною панеллю. Розширена підтримка пристроїв, в тому числі додана підтримка пристроїв USB і Firewire. Особливу увагу виділено безпеці системи: firewall тепер налаштовується в процесі установки системи, а графічна утиліта конфігурування значно спрощує процес створення ланцюгів. Fedora

Fedora Linux

З моменту виходу першої версії у 2003, *Fedora Linux* від Red Hat був кращим місцем для слідкування за найсвіжішими тенденціями у світі Linux та програмного забезпечення з відкритим джерельним кодом. При використанні Fedora у вашому розпорядженні декілька робочих оточень. Кожне з них дещо відрізняється від інших і пропонує різні способи індивідуального налаштування. Вони визначають вигляд та поведінку вікон, піктограм, меню та інших базових елементів інтерфейсу. Fedora Commons відноситься до оточуючих співтовариством Проект Fedora Repository. Це спільнота об'єднує із загальними потребами, варіанти використання, і проекти. Спільноті Fedora Commons дуже активний у виробництві додаткові інструменти, програми та утиліти, які збільшують сховище Fedora. Багато з цих творінь доступні для всього співтовариства з відкритим вихідним кодом.

Програмне забезпечення Fedora репозиторій був встановлений установами в усьому світі, для підтримки різних цифрових потреб змісту. Репозиторії Fedora є дуже гнучким і може бути використаний для підтримки будь-якого типу цифрового контенту. Існують численні приклади того, Fedora використовується для цифрових колекцій, електронних досліджень, електронні бібліотеки, архіви, цифрові збереження, інституційних репозитаріїв, відкритий доступ до публікацій, управління документами, цифровими активами, і багато іншого. Нижче наведено короткий список категоричне типів організацій в призначеному для користувача Fedora Commons співтовариства (в алфавітному порядку): Fedora (колишня Fedora Core) — дистрибутив Linux на основі RPM, котрий розробляється спільнотою Fedora Project, і спонсорується компанією Red Hat. Він 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 Hat і інших виробників. Підтримкою користувачів на добровільних засадах займається велика спільнота (серед якої і працівники Red Hat, однак, сама компанія підтримки не надає). Нові випуски Fedora виходять кожні 6-8 місяців. У розробці платформи бере активну участь більше 27 тисяч чоловік, кількість користувачів приблизно оцінюється в декілька мільйонів. Fedora, зокрема, використовує на більшості своїх машин автор ядра Linux Лінус Торвальдс. Fedora 10 Випуск ОС базується на версії ядра Linux 2.6.27.5, яке було випущено 7 листопада 2008. До складу Fedora 10 входять робоче середовище Gnome 2.24, менеджер робочого столу KDE 4.1, браузер Firefox 3.0.4 і офісний пакет OpenOffice 3.0. Учасники проекту Fedora реалізували в десятій версії дистрибутива покращені засоби віртуалізації і безпеки, а також підвищили продуктивність програмної платформи. Нова графічна система Plymouth дозволяє зменшити час завантаження комп'ютера; крім того, в Fedora 10 розширена база даних драйверів устаткування і покращена підтримка принтерів. Образ FedoraLive може бути доданий на флеш-брелок без необхідності його переформатування або видалення даних.

Suse та Novell

Suse

Дистрибутив під назвою SuSE Linux 1.0 вийшов в 1994 р. Його випустила німецька компанія Gesellschaft für Software-und System-Entwicklung. Довгий час цей продукт вважався зразком дружелюбності до початківцю: всі налаштування системи проводилися за допомогою зручної графічної програми YaST2. Однак повністю вільним даний дистрибутив не був. Оригінальний текст YaST2 був закритий. У 2003-му проект був куплений фірмою Novell, яка відкрила проект для розробників і користувачів. Оригінальний текст YaSY2 також було опубліковано, тому дистрибутив став вільним і отримав назву openSUSE.

Спадок SUSE було явно помітно у версії 9.1 (випущеної після її придбання фірмою Novell) з характерними утилітами адміністрування, великою кількістю програмних пакетів і широкої документацією. Починаючи з 10-ї версії, проект розробляється спільнотою openSUSE.

На основі openSUSE також збираються дистрибутиви для корпоративних клієнтів Novell Linux Desktop (NLD), Suse Linux Enterprise Desktop (SLED) і Suse Linux Enterprise Server (SLES).

У 2007-2008 роках Novell активно співпрацює з Microsoft, яка в результаті угоди про патенти підтримує і поширює дистрибутиви SuSE, в яких важливу роль відіграє Mono - запатентованою Microsoft платформи. NET., що викликало невдоволення частини спільноти користувачів та розробників вільного ПЗ. Починаючи з версії 11.0 система управління пакетами використовує більш могутній чим bzip2, компресор даних - LZMA. У версії 11.1 додана підтримка SELinux. Причому підтримка AppArmor (за замовчуванням) збережена в повній мірі. Рішення по активації тієї чи іншої системи приймає користувач.

Останній повноцінний реліз OpenSUSE 12,1 з версією ядра Linux 3.1.0 вийшов 2011-11-16. Історія SUSE Linux почалась з того моменту, коли у середині 1992 року Пітер МакДональд (Peter McDonald) випустив один з перших дистрибутивів GNU/Linux — SLS. Це був дистрибутив, який вперше пропонував користувачу не просто ядро і набір системних утиліт, а також широкий набір іншого програмного забезпечення (включаючи підтримку XFree86 і TCP/IP). Після цього, Патрік Фолкердинг (Patric Volkerding) створив знаменитий дистрибутив Slackware Linux, який в свою чергу, став основою для багатьох інших дистрибутивів. У цей же час, у Німеччині з'являється компанія «Gesellschaft für Software- und System-Entwicklung» («Програмна і системна розробка»), скорочено «S.u.S.E.», яка займалася консалтингом у питаннях Unix-систем. Крім того, фірма регулярно випускала програмні пакунки, що включали SLS і Slackware, друкувала керівництва для Unix і GNU/Linux. У 1994 році вони випустили першу CD-версію SLS/Slackware під назвою S.u.S.E. Linux 1.0. Пізніше, у 1996 році, вони інтегрувалися із ще одним дистрибутивом, Jurix, також заснованим на Slackware. Це вже був самостійний дистрибутив, що отримав назву S.u.S.E. Linux 4.2. 4 листопада 2003 року компанія SUSE Linux була куплена американською компанією Novell. Novell не стала втручатись у справи компанії, а, навпаки, випустила під ліцензією GPL вихідні коди однієї з головних особливостей дистрибутиву — програми адміністрування YaST. Після придбання корпорацією Novell у 2003 році та появи openSUSE ситуація змінилась: починаючи з версії 9.2, був доступний для завантаження непідтримуваний образ DVD, так само як і ознайомчий live CD. Працюють ftp-сервери із пакунками, надаючи користувачам переваги встановлення на вимогу, завантажувались тільки необхідні пакунки. Перевагою образів є легкість

встановлення, можливість роботи навіть якщо мережна карта не працює «з коробки» та менший мінімальний досвід (наприклад, недосвідчені користувачі можуть не знати, чи потрібно встановлювати той чи інший пакунок, тому образи пропонують передвизначені набори). Перший стабільний випуск openSUSE Project, SUSE Linux 10.0, був доступний для завантаження ще перед офіційним випуском SUSE 10.0. Крім того, Novell припинила розробку персональної версії, перейменувавши професійну в просто SUSE Linux та змінивши ціну на приблизно рівну колишній персональній версії. З версії 10.2, SUSE Linux було офіційно перейменовано на openSUSE. З часом SuSE Linux пройшов шлях від дистрибутиву з пропрієтарними компонентами, з затриманим відкриттям коду (2 місяці очікування для тих, хто не хотів купувати фабричну версію) та закритою моделлю розробки до вільного розповсюдження та прозорі і відкритої моделі розробки. Популярність дистрибутиву продовжує зростати, навіть на тлі невдоволення частини спільноти лінукс співпрацею Novell із Microsoft. Від 2007 року Novell активно співробітнічає з Microsoft, яка в результаті угоди про патенти підтримує і поширює дистрибутиви SUSE, в яких важливу роль грає Mono, — реалізація запатентованої Microsoft платформи .NET[1], що викликало незадоволеність частини співтовариства користувачів і розробників вільного ПЗ.[2] Також Novell надає допомогу при переході на Windows Vista, і рекламує SUSE як єдиний дистрибутив Linux, сертифікований для роботи під Windows.

Novell

Novell Linux Desktop — один из дистрибутивів Linux. Система Novell Linux Desktop, заснована на технологіях SUSE LINUX, являє собою настільну операційну систему і ділове середовище для офісу, що дозволяє підприємствам з упевненістю використовувати Linux і програмне забезпечення open source. Система Novell Linux Desktop призначена для широкого кола користувачів і може бути оптимізована для вирішення таких постійних або спеціальних завдань, як робочі станції загального користування, інформаційні кіоски і термінали call-центрів. Компанії можуть впевнено скористатися захищеністю, економічністю, продуктивністю і свободою, які гарантує настільна платформа open source, знаючи, що вона надана Novell, провідним постачальником корпоративних інформаційних рішень.

Поточна версія 11 використовує програму ZENworks Linux Management для оновлення системи. Novell Linux Desktop заснований на SUSE 10 Enterprise Server і використовує безліч його пакетів: універсальний конфігуратор YaST, Графічні оболонки GNOME 2.6 і KDE 3.2.1, браузер Mozilla Firefox 3.0 і поштовий клієнт Novell Evolution 2.0 і багато інших open source програми. У дистрибутив також включені і деякі пропрієтарні програми, наприклад універсальний програвач Real Player. Заснована компанія була в Прово, штат Юта, як Novell Data Systems Inc. у 1979 році. Займалася виробництвом систем, що працюють під управлінням CP/M. Спів-засновником фірми був Джорж Канова (George Canova). Назва фірми — Novell — було запропонована його дружиною, яка помилково вважала, що «Novell» означає по-французьки «новий». У січні 1983, компанія була перейменована в Novell Inc., а її головою став Реймонд Нурда (Raymond J. Noorda). Того ж 1983 року компанія випустила свій найзначніший продукт — мережеву операційну систему NetWare, перше програмне забезпечення для мереж, засноване на технології файл-

сервера. Novell розробила мережеву систему для ПК, в якій одна машина виділяється для управління мережею і контролю за доступом до загальних пристроїв, таким як жорсткі диски і принтери. За 1980-і роки вимоги компаній до мереж значно виросли, і локальні мережі все частіше стали замінювати територіально розподіленими мережами, об'єднуючими відділення великих корпорацій. На початок 1990-х операційна система Novell NetWare, доповнена функціями, необхідними для розподілених підприємств, лідирувала в цій області з часткою ринку близько 70%. Novell базував свій мережевий протокол на XNS і створив свої стандарти IDP і SPP, яким дали імена IPX (Internet Packet eXchange) і SPX (Sequenced Packet eXchange). Файлові і принтерні сервіси працювали за протоколом NCP (Netware Core Protocol) над IPX, як і протоколи Routing Information Protocol (RIP) і Service Advertising Protocol (SAP). На додаток, фірма стала просувати свою Novell DOS, аналогічну MS-DOS. У 1996 році під тиском поширення Windows і визнаючи, що інтернет починає радикально міняти традиційний ринок мереж, тимчасово виконуючий обов'язки генерального директора Джон Янг (John Young) ініціював програму по оснащенню продуктів компанії засобами підтримки інтернету. У березні 1997 року кермо влади Novell узяв Ерік Шмідт (Eric Schmidt), який прискорив роботу з використання мережевих технологій Novell на арені інтернету. Наступного року почалися постачання серверної операційної системи NetWare 5 і Novell Directory Services (NDS) із стандартною підтримкою комунікаційного протоколу інтернету IP. У 1998 році зважаючи на поглиблення неоднорідності корпоративних мереж і потреби у взаємодії через інтернет Novell почала пропонувати NDS як засіб взаємозв'язку між різними платформами. Компанія випустила також продукти, що використовують інформацію, яка зберігається в каталозі, для спрощення управління мережами і безпечнішого доступу до них з використанням облікових записів користувачів. В кінці 1999 року Novell випустила eDirectory, крос-платформову службу каталогів відкритих стандартів.

В серпні 2003 року Novell придбав компанію Ximian з її провідними розробниками Linux і рішеннями для робочих станцій Linux, управління ПК і серверами Linux і колективної роботи під Linux і Windows. Разом з Ximian в Novell прийшли два лідери руху open-source, Мігель де Ікаса (Miguel de Icaza) і Натан Фрідман (Nat Friedman), а з ними — довіра до Novell з боку Linux-співтовариства. Ximian (а зараз Novell) веде два найбільші проекти open-source: проект робочого столу GNOME і Mono — платформу open-source для виконання застосунків Microsoft .NET. У січні 2004 року Novell зробила ще один важливий крок в своїй Linux-історії, завершивши придбання SUSE Linux, провідного європейського постачальника одного з популярних комерційних дистрибутивів Linux. Тепер Novell пропонує повний спектр Linux-рішень від сервера до робочих станцій з додатковими мережевими службами корпоративного рівня і технічною підтримкою. У серпні 2005 року в розширення своєї прихильності до співтовариства open-source Novell відкрила сайт www.openSUSE.org, призначений для стимулювання використання Linux у всьому світі шляхом надання вільного і простого доступу до дистрибутива SUSE Linux. Проект openSUSE робить відкритим і процес розробки SUSE Linux, запрошуючи програмістів open-source до участі в перевірці, тестуванні і розробці.

Slackware та GNU

Slackware 13,37 майже рік в процесі становлення, ви зможете оцінити продуктивність і стабільність, яка може прийти тільки при ретельному тестуванні. Slackware 13,37 використовує Linux ядро 2.6.37.6 (звідси і наше нове \$ SLACKWARE_VERSION. KERNEL_VERSION \$ іменування система, яка використовується для цього релізу), а також поставляється з 2.6.38.4 ядра для тих, хто хоче запустити останню (а також включає в себе файли конфігурації для 2.6.35.12 і 2.6.39-RC4). Slackware 13,1 доступна як в 32-розрядних x86 і 64-розрядні версії x86_64, і приносить із собою багато значних поліпшень, таких як KDE SC версії 4.4.3, 2.6.33.4 Linux ядра, новий набір інструментів, оновлені бібліотеки та основних програм таких як Firefox і Thunderbird.

Slackware має новий офіційний порт для ARM-архітектури, на ім'я ARMedslack, яка недавно випустила порт Slackware версії 12.2. ARMedslack почалося в 2002 році Стюарт Зима, з головною метою забезпечення повної порт Slackware для машин ARM робочий стіл - спочатку з орієнтацією на Acorn StrongARM RiscPC, а потім і вбудованих пристроїв.

Slackware ARM тепер підтримує рідну установки за допомогою звичайної програми установки Slackware на ARM Універсальна плата (за допомогою емулятора QEMU) і недавно оголосила про вбудованого пристрою: Marvell SheevaPlug.

Slackware ARM версії 12,2 побудований для ARMv4, прямий порядок байтів, "Спадщина" ABI. Новий порт в EABI вже ведеться і буде доступна в "струм" формі протягом кількох місяців.

Slackware Linux - один з найстаріших дистрибутивів GNU / Linux. Він трохи відрізняється від інших популярних дистрибутивів тим, що Slackware був і залишається «найбільш Unix'ів» - його рисами є стабільність і безпека, а також простота установки, підтримки і використання. Шанувальникам цього дистрибутиву приписують такий вислів: «Якщо ви знаєте Slackware - ви знаєте GNU / Linux. Якщо ви знаєте Red Hat, то все, що ви знаєте - це Red Hat». Slackware був першим Linux дистрибутивом, який отримав широке поширення. Початок був покладений Патріком Волькердіногм (Патрік Фолькердінг (Patrick J. Volkerding, також відомий як Mr. Slackware і The Man) - творець дистрибутиву Slackware і компанії Slackware Linux, Inc. По випуску даного дистрибутива.) В кінці 1992 року. Патрік зіткнувся з Linux, коли шукав недорогий інтерпретатор мови LISP для свого проекту. У той час вибір серед Linux дистрибутивів був невеликий, існувало всього кілька Linux дистрибутивів, і Патрік вибрав Soft Landing Systems (SLS Linux). Тим не менш, у SLS Linux були свої проблеми і Патрік почав виправляти дрібні помилки, по мірі їх виявлення.

У підсумку, він вирішив об'єднати всі виправлення та доповнення в свій власний Linux дистрибутив. Цей приватний Linux дистрибутив дуже швидко знайшов популярність і Патрік зробив його доступним для громадськості під ім'ям Slackware Linux. Патрік так само додав нові риси свого Linux дистрибутива. Такі як процес установки з дружнім (user friendly) інтерфейсом, заснований на системах меню і ввів концепцію менеджменту пакетів програм. Що дозволило користувачам легко додавати, видаляти або оновлювати пакети програм в їх системі.

GNU

Лінукс (англ. Linux, також відомий, як GNU/Linux) — загальна назва UNIX-подібних операційних систем на основі однойменного ядра. Це один із найвидатніших прикладів розробки з відкритим кодом (open source) та вільного програмного забезпечення (free software); на відміну від пропріетарних операційних систем, на кшталт Microsoft Windows та MacOS X, її вихідні коди доступні усім для використання, модифікації та розповсюдження абсолютно вільно (в т.ч. безкоштовно). Спершу розроблювалася та використовувалася індивідуальними ентузіастами на персональних комп'ютерах, з тих пір Linux завдяки підтримці таких компаній як IBM, Sun Microsystems, HP, Novell та інших, набув неабиякої популярності, як серверна операційна система; так, 8 із 10 найбільших компаній, що надають послуги веб-хостингу використовують Лінукс на своїх веб-серверах. Лінукс портовано на велику кількість апаратних платформ. Тепер вона досить успішно використовується, як на суперкомп'ютерах, так і на мобільних телефонах. Значна кількість спеціалізованих дистрибутивів Linux, котрі розробляються та підтримуються різними спільнотами, дає широкі можливості вибору програмного забезпечення. Станом на травень 2009 року операційні системи Linux займали долю біля 1 % світового ринку операційних систем на персональних комп'ютерах (без врахування використання на серверах та спеціалізованих пристроях)[1]. GNU-вільна UNIX-подібна операційна система, що розробляється Проектом GNU.

Дистрибутиви GNU на основі HURD Bee GNU/Hurd

Debian GNU/Hurd

GNUAB

Superunprivileged.org'івський Hurd LiveCD

Unofficial Gentoo/Hurd Project Операційна система (ОС) складається з декількох основних програм, які потрібні вашому комп'ютеру, щоб він міг спілкуватися і отримувати інструкції від користувачів; читати і записувати дані на жорсткі диски, стрічки і принтери; контролювати використання пам'яті; запускати інші програми. Найважливіша частина ОС - ядро. В системі GNU / Linux ядром є Linux. Частина, що залишилася системи складається з інших програм, багато з яких написані проектом GNU або для проекту GNU. Оскільки одне ядро Linux - це ще не вся працююча ОС, ми воліємо використовувати термін «GNU / Linux» для позначення систем, які багато людей помилково називають «Linux». Linux зроблена за зразком операційної системи Unix. З самого початку Linux розроблялася як багатозадачна, багатокористувацька система. Цих факторів досить, щоб зробити Linux відмінною від інших широко відомих операційних систем. Однак, Linux відрізняється набагато сильніше, ніж ви можете собі уявити. На відміну від інших операційних систем, Linux нікому не належить. Значна частина системи розроблена добровольцями безкоштовно. Розробка того, що пізніше стане GNU / Linux, почалася в 1984 році, коли Фонд Вільного Програмного забезпечення почав розробку вільної Unix-подібної операційної системи, названої GNU. Проект GNU розробив всеосяжний набір вільного програмного забезпечення для використання в Unix™ і Unix-подібних операційних системах, таких як Linux. Ці програми дозволяють користувачам виконувати завдання широкого спектра, від простих (копіювання або видалення файлів із системи) до складних (написання та компіляція

програм або виконання складного редагування в різних форматах документів). Хоча багато робочі групи і окремі люди вклали свою працю в створення Linux, найбільшим поки ще залишається внесок Фонду Вільного Програмного Забезпечення. Крім розробки більшості використовуваних інструментів, фонд також створив філософію і організував спільнота (незалежних програмістів і ентузіастів-користувачів). Від цього, в свою чергу, стало можливим поява налагоджених Linux систем. Ядро Linux вперше з'явилося в 1991 році, коли фінський студент-комп'ютерник по імені Лінус Торвальдс (Linus Torvalds) анонсував першу версію заміни ядра Minix в групі новин comp.os.minix. Детальніше дивіться на сайті Linux International's сторінку історії Linux. Лінус Торвальдс продовжує координувати роботу кількох сотень розробників за допомогою декількох супроводжуючих підсистем. Є офіційний сайт для ядра Linux. Детальну інформацію про список розсилки linux-kernel можна знайти в FAQ списку розсилки linux-kernel. Користувачі Linux мають величезну свободу у виборі програмного забезпечення. Наприклад, користувачі Linux можуть вибирати з дюжини різних оболонок командного рядка і декількох графічних робочих столів. Цей вибір часто дивує користувачів інших операційних систем, які і не думали, що командні оболонки або робочий стіл можна замінити. Linux також менш схильний збоїв, краще пристосований для одночасного виконання більш ніж однієї програми і більш захищений, ніж багато інших систем. Завдяки цим перевагам, Linux швидко завойовує ринок серверних операційних систем. З недавніх пір, Linux почав ставати популярним серед домашніх користувачів і в корпоративному середовищі.

ТЕМА 4. ПРОЦЕС ЗАВАНТАЖЕННЯ LINUX.

4.1 BIOS ТА ЗАВАНТАЖУВАЧ СИСТЕМИ (BOOT LOADER).

BIOS BIOS- це базова система введення-виведення (Base Input-Output System). Призначенням BIOS є виконання найбільш простих і універсальних послуг операційної системи, що пов'язані зі здійсненням вводу-виводу. BIOS призначена для:

- а) тестування основних компонентів ПК після включення живлення;
- б) виклик блоку початкового завантаження BOOT RECORD;
- в) обслуговування системних переривань нижнього рівня, тобто тих які вимагають безпосереднього керування апаратними компонентами (дисплеєм, клавіатурою, магнітними дисками, принтерами, комунікаційними каналами).

Таким чином, BIOS є програмною оболонкою навколо апаратних засобів комп'ютера, яка надає можливість іншим програмам, у тому числі й самій ОС, звертатися до апаратних компонент через механізм переривань. Суть механізму переривань полягає в тім, що поточна робота машини, у чому б вона не складалася, припиняється на короткий проміжок часу визначеним сигналом, що однозначно вказує на виникнення ситуації, що вимагає негайної обробки. Очевидно, що від того, як саме настроїти BIOS той чи інший пристрій, залежить швидкодія і стабільність усієї системи в цілому.

Програма Setup, доступ до якої можна одержати, натиснувши "DEL" (чи "F2") при завантаженні, саме і дозволяє змінювати ті значення, що завантажуються в реєстри різних пристроїв, насамперед чіпсета материнської плати. До речі, зберігаються вони в пам'яті, що живиться від батарейки, а пам'ять цю називають CMOS (Complimentary Metal-Oxide-Semiconductor, що споживає невелику потужність у статичному режимі логіка). Робота таких стандартних пристроїв, як клавіатура, може обслуговуватися програмами BIOS, але такими засобами неможливо забезпечити роботу з усіма можливими пристроями (у зв'язку з їх величезною різноманітністю та наявністю великої кількості різних параметрів). Але для своєї роботи програми BIOS вимагають всю інформацію про поточну конфігурацію системи. З очевидних причин цю інформацію не можна зберігати ні в оперативній пам'яті, ні в постійній. Спеціально для цих цілей на материнській платі є мікросхема енергонезалежної пам'яті, яка по технології виготовлення називається CMOS. Від оперативної пам'яті вона відрізняється тим, що її вміст не зникає при вимкненні комп'ютера, а від постійної пам'яті вона відрізняється тим, що дані можна заносити туди і змінювати самостійно, у відповідності з тим, яке обладнання входить до складу системи. Мікросхема пам'яті CMOS постійно живиться від невеликої батарейки, що розташована на материнській платі. У цій пам'яті зберігаються дані про гнучкі та жорсткі диски, процесори і т.д. Той факт, що комп'ютер чітко відслідковує дату і час, також пов'язаний з тим, що ця інформація постійно зберігається (і обновлюється) у пам'яті CMOS. Таким чином, програми BIOS зчитують дані про склад комп'ютерної системи з мікросхеми CMOS, після чого вони можуть здійснювати звертання до жорсткого диска та інших пристроїв. Boot Loader Завантаження операційної системи (booting) — це багатокроковий процес запуску комп'ютера. Завантажувач операційної системи (bootloader) — це

програма, для виконання завантаження операційної системи. Завантажувач операційної системи звичайно міститься у секторі завантажування. Типи завантажень:

- Початкове завантаження (booting up) - Завантажування після того, як комп'ютер увімкнено користувачем.
- Перезавантаження (reboot) .

Функції завантажувача системи:

– забезпечує необхідні засоби для діалогу з користувачем комп'ютера (наприклад, завантажувач дозволяє вибрати ядро операційної системи для завантаження);

– приводить апаратуру комп'ютера в стан, необхідний для старту ядра операційної системи;

– завантажує ядро операційної системи в оперативну пам'ять.

Завантаження ядра операційної системи не обов'язково відбувається з жорсткого диск. Завантажувач може отримувати ядро з мережі. Ядро може зберігатися в постійній пам'яті або завантажуватися через послідовні інтерфейси; } формує параметри, що передаються ядру операційної системи (наприклад, ядру Linux передаються параметри, що вказують спосіб підключення кореневої файлової системи); } передає управління ядру операційної системи.

4.2 Завантажувачі GRUB, syslinux, lilo.

GRUB

GNU GRUB (GRand Unified Bootloader — Грандіозний уніфікований завантажувач) —завантажувач операційної системи відпроєкту GNU. GRUB дозволяє користувачеві мати кілька встановлених операційних систем і при включенні комп'ютера вибирати одну з них для завантаження. GRUB є еталонною реалізацією завантажувача, відповідного специфікаціям Multiboot і може завантажити будь-яку сумісну з нею операційну систему. Серед них: Linux, FreeBSD, Solaris і багато інших. Крім того, GRUB вміє по ланцюжку передавати управління іншому завантажувачу, що дозволяє йому завантажувати Windows. Після налаштування GRUB користувач при включенні комп'ютера бачить список операційних систем, які встановлені на його комп'ютер і які можна завантажити, вибравши відповідну і натиснувши Enter. GRUB дозволяю проводити захист паролем пунктів меню підтримує файлових систем: NTFS, ISO,UFS,UFS2, FFS, FAT16, FAT32, Minix, ext2/ext3/ext4, ReiserFS,JFS і XFS, і читання файлів конфігурації, ядер,initrd та інших файлів прямо з файлової системи.

Syslinux

Syslinux - завантажувач ядра Linux. Оригінальний SYSLINUX, використовуваний для завантаження з FAT-або FAT32 файлових систем (таких як флоппі-диски і USB-накопичувачі). SYSLINUX зазвичай не використовується як завантажувач системи Linux, адже Linux зазвичай не встановлюють на FAT файлових системах. Замість цього, він часто використовується для завантаження з диску або USB-накопичувача.

LILO

LILO (Linux LOader) - один із стандартних завантажувачів для Linux та FreeBSD. LILO не залежить від будь-яких файлової системи, в змозі завантажити ядра операційної системи Linux. Програма підтримує від 1 до 16 різних записів. LILO може бути встановлений в якості основної програми завантаження в MBR (запускати безпосередньо BIOS) або завантажувальний сектор.

Процес завантаження Linux

Процес завантаження Linux являє собою дії, за допомогою яких приводяться у стан готовності операційні системи на основі Linux. При завантаженні комп'ютера відбувається послідовна передача управління від BIOS до завантажувача, а від нього - до ядра. Потім ядро запускає планувальник (для реалізації багатозадачності) і виконує програму `init` (яка налаштовує користувацьке середовище і дозволяє здійснювати взаємодію з користувачем і вхід в систему), після чого ядро переходить в стан бездіяльності до тих пір, поки не отримає зовнішній виклик. Основні етапи завантаження:

BIOS виконує завантажувальні кроки, спеціальні для даного апаратного забезпечення. Коли всі пристрої розпізнані й правильно запущені, BIOS передає управління завантажувачу, який часто пропонує користувачеві меню з доступними варіантами завантаження. Потім він завантажує ядро, яке розпаковується в пам'ять, налаштовує системні функції, такі як робота необхідного обладнання та управління сторінками пам'яті, після чого робить виклик `start_kernel ()`.

Після цього `start_kernel ()` виконує основну настройку системи (переривання, інші функції управління пам'яттю, ініціалізацію пристроїв, драйверів і т. д.), а потім породжує процес бездіяльності, диспетчер і окремо від них - процес `init` (виконується в просторі користувача).

Планувальник починає більш ефективно управляти системою, в той час як ядро переходить до бездіяльності. Процес `init` виконує необхідні сценарії, які налаштовують всі служби і структури, які не відносяться до рівня ядра, в результаті чого буде створено користувацьке середовище, і користувачеві буде наданий екран входу в систему. Коли відбувається завершення роботи, `init` викликається для керованого закриття програм користувацького рівня, теж згідно сценаріями. Після цього `init` закривається, а ядро виробляє своє власне завершення роботи.

4.3 Процеси.

4.3.1 Загальні відомості

Процес - це об'єкт ОС Linux, який складається з адресного простору пам'яті і набору структур даних. По суті, процес це запущена програма або служба. Кожен запущений процес в ОС Linux може породити додаткові процеси. Процес, що запустив новий процес називається батьківським процесом. Новий процес по відношенню до створив його процесові називається дочірнім. Процеси - це не те ж саме, що завдання: процеси є частиною операційної системи, тоді як про завдання відомо тільки командному процесору, в якому вони виконуються. Працююча

програма містить в собі один або більше процесів; завдання складається з однієї або більше програм, виконуваних у вигляді команд командного процесора. Кожен процес в ОС Linux характеризується набором атрибутів, який відрізняє даний процес від всіх інших процесів. До **головних атрибутів** відносяться:

- Ідентифікатор процесу (PID). Кожен процес в системі має унікальний ідентифікатор. Кожен новий запущений процес отримує номер на одиницю більше попереднього.
- Ідентифікатор батьківського процесу (PPID). Даний атрибут процес отримує під час свого запуску та використовується для отримання статусу батьківського процесу.

У Linux процеси діляться на три типи:

- **Системні процеси** - є частиною ядра і завжди розташовані в оперативній пам'яті. Системні процеси не мають відповідних їм програм у вигляді виконуваних файлів і запускаються при ініціалізації ядра системи. Виконувані інструкції і дані цих процесів знаходяться в ядрі системи, таким чином, вони можуть викликати функції і звертатися до даних, недоступним для інших процесів. Системними процесами, наприклад, є: `shd` (диспетчер свопінгу), `vhand` (диспетчер сторінкового заміщення), `kmadaemon` (диспетчер пам'яті ядра).

- **Демони** - це неінтерактивні процеси, які запускаються звичайним чином - шляхом завантаження в пам'ять відповідних їм програм (виконуваних файлів), та виконується у фоновому режимі. Зазвичай демони запускаються при ініціалізації системи (але після ініціалізації ядра) і забезпечують роботу різних підсистем: системи термінального доступу, системи друку, система мережевого доступу і мережевих послуг, поштовий сервер, `dhcp`-сервер і т. п. Демони не пов'язані ні з одним користувальницьким сеансом роботи і не можуть безпосередньо управлятися користувачем. Велику частину часу демони чекають поки той або інший процес запросить певну послугу, наприклад, доступ до файлового архіву або друк документа.

- **Прикладні (користувальницькі) процеси**. До прикладних процесів належать всі інші процеси, що виконуються в системі. Як правило, це процеси, породжені в рамках користувальницького сеансу роботи. Наприклад, команда `ls` породить відповідний процес цього типу. Прикладні процеси Linux можуть виконуватися як в інтерактивному, так і у фоновому режимі, але в будь-якому випадку час їхнього життя (і виконання) обмежена сеансом роботи користувача. При виході з системи всі прикладні процеси будуть знищені.

4.3.2 Демони Лінукс

Отключить ненужные сервисы нужно по трем причинам:

- уменьшается время загрузки - понятно, что чем меньше сервисов запускается при запуске системы, тем быстрее она загружается;

- уменьшается загрузка процессора и оперативной памяти - ведь сервисы находятся в памяти в течение всего времени работы системы, следовательно, они занимают процессорное время и оперативную память;

- каждый сервис нужно расценивать как потенциальную <дыру> в системе безопасности - запущенный сервис не настроен, он работает с настройками по умолчанию. Ясно, что эти настройки предназначены только для запуска сервиса, а

не для реальной работы. Как говорится, стандартные средства стандартно и взламываются.

Основные практические рекомендации:

- Если сервис не нужен, выключите его;
- Нужный вам сервис правильно настройте и используйте;
- Если сервис нужен, но нет времени его настроить (или вы не знаете, как это сделать), выключите его (включить всегда успеете).

Для настройки сервисов (служб) используются следующие конфигураторы

system-config-services - в Fedora и ASPLinux;

drakxservices - в Mandriva;

services-admin - в Debian и Ubuntu (тут могу и ошибиться в названии).

В современных дистрибутивах вы можете обнаружить следующие сервисы:

abrt (Automatic Bug Reporting Too) - собирает информацию о крахе системы и отправляет разработчикам, его можно смело выключать (хотя лично я его оставила);

acpid - управляет ACPI-событиями, по большому счету, он просто не нужен;

apmd - нужен только на ноутбуках;

anacron, atd, crond - демоны-планировщики, которые запускают указанные пользователем команды в определенное время. Домашнему пользователю они вряд ли нужны, во всяком случае три сразу (два можно отключить с чистой совестью);

avahi-daemon - демон ZeroConf, реализующий стек mDNS (Multicast DNS), можно отключить;

auditd - система аудита Linux. Если этот сервис выключен, события аудита будут отправлены на syslog;

bluetooth - если вы не собираетесь использовать технологию Bluetooth, данный сервис можете отключить;

btseed - нужен для BitTorrent, если вы не используете BitTorrent, этот сервис можно выключить;

cpuspeed, haldaemon - не вдаваясь в подробности, просто отключите эти сервисы (например, haldaemon - собирает информацию об аппаратных средствах компьютера);

cups* - система печати CUPS (Common Unix Printing System). Нужна только, если есть принтер;

dm - диспетчер дисплея (display manager), нужен, если вы планируете работать в графическом режиме;

dnsmasq - запускает кэширующий DNS-сервер;

freshclam - обновление антивирусной базы ClamAV, естественно, если вы не используете эту антивирусную программу, то сервис можно отключить;

firstboot - сервис, запускаемый только при первой загрузке Fedora, затем он будет автоматически выключен;

hidd - демон HIDD (Human Interface Device Daemon) обеспечивает поддержку клавиатур, манипуляторов и других устройств для ввода информации, работающих,

через Bluetooth. Если таких устройств у вас нет, можно смело выключить этот демон;

hddtemp - предоставляет информацию о температуре жесткого диска. В большинстве случаев можно отключить;

httpd - Web-сервер Apache (в некоторых дистрибутивах этот сервис называется apache);

isdn - сервис поддержки ISDN-линий. Если у вас нет ISDN, выключите этот сервис;

ip6tables - версия брандмауэра iptables, работающая с протоколом IPv6. На данный момент основной является все-таки четвертая версия протокола IP (IPv4), для которой этот демон бесполезен;

iptables - брандмауэр Linux;

irqbalance - нужен только на SMP-машинах (многопроцессорных машинах);

kheader - выполняет автоматическую генерацию заголовков ядра в /boot. Не отключайте этот сервис;

kudzi (в Mandriva - harddrake2) - сервис определения новых устройств. В целях экономии времени при загрузке системы его можно выключить, а запускать вручную после установки нового устройства;

lm_sensors - используется для мониторинга различных параметров системы (например, температуры процессора). Для правильной настройки этого сервиса нужно потратить много времени (к тому же не все <железо> его поддерживает), поэтому просто выключите его;

mandi - демон мониторинга сети. В большинстве случаев не нужен;

mdadm - демон мониторинга и управления программными RAID-массивами;

messagebus - <шина> сообщений, в современных дистрибутивах отключать этот сервис не рекомендуется (а раньше можно было просто выключить);

mDNSResponder, nifd - можете просто выключить эти два сервиса;

mdmonitor - используется для мониторинга программных RAID-массивов;

msec - включает политику безопасности MSEC, по умолчанию выключен;

netconsole - протоколирование сетевой консоли, по умолчанию отключен;

network - поддержка сети, не выключайте данный сервис (если он изначально был включен). Это устаревший сервис, но он до сих пор используется в современных дистрибутивах, например, в Mandriva 2010;

Network Manager - новый сервис поддержки сети, используется вместо устаревшего сервиса network. В новых дистрибутивах есть оба сервиса; Network Manager и network. Первый по умолчанию включен, а второй - выключен. В некоторых случаях Network Manager нужно отключить. Тогда для поддержки сети следует включить сервис network;

netfs - обеспечивает поддержку различных сетевых файловых систем (в том числе и для поддержки SMB). Нужен в локальной сети, где есть необходимость в сетевых файловых системах (это никак не относится к протоколу FTP);

netplugd (в Mandriva 2008 - network-up) - демон управления нестатическими сетевыми интерфейсами. Можно с чистой совестью отключить;

partmon - проверяет наличие свободного места на разделах жесткого диска;

pcmcia (pccsd) - нужен для поддержки PCMCIA-карт, Если у вас не ноутбук, можете смело выключить этот сервис;

portmap - забезпечує мапінг портів. Домашньому користувачеві не потрібен, а ось на сервері пригодиться;

resolvconf - менеджер інформації про доменні сервери, не відключайте цей сервіс навіть на клієнтах;

rhnsd - можна сміливо відключити;

rpc* - підтримка віддаленого виклику процедур RPC (Remote Procedure Call), в більшості випадків даний сервіс не потрібний;

sendmail (або postfix)- служить для організації власного SMTP-сервера, тобто для сервера відправки повідомлень електронною поштою. Дуже складно налаштувати, тому краще поки відключити його. Коли у вас буде необхідність у власному SMTP-сервері, увімкнути цей сервіс особливих проблем не складе;

shorewall (в інших дистрибутивах - iptables) - пакетний фільтр (брандмауер). Якщо ви ще не налаштовували пакетний фільтр, то краще поки відключити;

smartd - потрібен для підтримки S.M.A.R.T-пристроїв. Якщо у вас немає таких пристроїв, можете відключити цей сервіс;

snmp* - підтримка SNMP (Simple Network Management Protocol). Якщо вам SNMP не потрібен, відключіть цей сервіс;

sshd - необхідний для безпечного віддаленого доступу до консолі системи. В більшості випадків не потрібен;

winbind - частина Samba, потрібен в Windows-мережах;

yum-updatesd - служба оновлення, заснована на технології yum. Існує тільки в Fedora Core і інших дистрибутивах, що використовують yum (Mandriva до нього не відноситься). Якщо ви не плануєте оновлювати свою систему по Інтернету, відключіть цей сервіс.

4.4 Процес init.

init (*initialization* — ініціалізація) — програма, що запускає всі інші процеси. Працює як демон і звичайно має PID 1. Зазвичай розташовується по шляху /sbin/init.

У процесі завантаження, після ініціалізації ядра, ядро запускає /sbin/init як перший процес користувачького режиму. init відповідає за подальше завантаження системи. Для цього він запускає стартові скрипти, які виконують перевірку та монтування файлових систем, запуск необхідних демонів, налаштування ядра (у тому числі завантаження модулів ядра згідно з установленим обладнанням, налаштування IP-адрес, таблиць маршрутизації тощо), запуск графічної оболонки та інші дії.

Диспетчер пристроїв udev.

Udev є диспетчер пристроїв для Linux ядра. Насамперед, він керує пристроїв в / Dev. Він є наступником DevFS і автовизначення, яка означає, що він обробляє / розробника каталог і всі простір користувача дій при додаванні / видаленні пристроїв, включаючи прошивку навантаження

Історія

Udev було новим в Linux 2.5. Версія Linux ядра 2.6.13 введені або оновлені нової версії uevent інтерфейс. Системи з допомогою нової версії Udev не буде

завантажуватися з ядрами старіше ніж 2.6.13, якщо Udev відключається, і традиційні / розробника каталог використовується для доступу до пристрою.

Огляд

На відміну від традиційних Unix- систем, де файли пристроїв в / Dev каталогів були статичного набору файлів, пристрій Linux Udev менеджер динамічно забезпечує тільки вузли для пристроїв, які фактично знаходяться в системі. Хоча DevFS використовується для забезпечення аналогічної функціональності, прихильники Udev навів ряд причин [1] для воліючи її реалізації протягом DevFS: Udev підтримує постійне іменування пристроїв, яка не залежить від, наприклад, порядок, в якому пристрої підключені до системи. Установки за замовчуванням Udev забезпечує постійне імена для пристроїв зберігання даних. Будь жорсткий диск визнається його унікальний ідентифікатор файлової системи, ім'я диска і фізичне розташування на обладнання, яке він підключений. Udev виконує повністю в просторі користувача , на відміну від DevFS ' простору ядра . Одним з наслідків цього є те, що Udev переїхав політику іменування з ядра і можете запускати довільні програми писати ім'я пристрою з властивостей пристрою, перш ніж вузол буде створений.

Udev це загальний диспетчері пристроїв ядра. Він запускається як демон на Linux системи і слухає (через NetLink гніздо) для uevents ядро посилає, якщо новий пристрій ініціалізується або пристрій видаляється з системи. Система надає набір правил, матч проти експортувала значення події та властивості виявленого пристрою. Відповідні правила, можливо, ім'я і створити вузол пристрою і запустити налаштовані програми установки та налаштування пристрою. Udev правила можуть зрівнятися за властивостями, як підсистеми ядра, ім'я пристрою ядра, фізичне місце розташування пристрою, або властивості, такі як серійний номер пристрою. Правила можуть також запросити інформацію із зовнішніх програм, щоб назвати пристрій або вкажіть довільне ім'я, яке завжди буде однаковою, незалежно від порядку пристроїв виявлених в системі. У минулому звичайний спосіб використання Udev на Linux системах, щоб відправити подій через сокет HAL , який буде виконувати подальші пристрої конкретних дій. Наприклад, HAL буде повідомляти інші програми, встановлені в системі, що нове обладнання прибутку за рахунок випуску широкомовне повідомлення про D-Bus IPC системи для всіх зацікавлених процесів . Таким чином, настільних ПК, таких як GNOME або KDE може відкрити файл браузера, щоб знову додається USB флеш-накопичувачі і карти пам'яті SD . [2] До середини 2011 року HAL були застарілими більшість дистрибутивів Linux, а також KDE, GNOME [3] і XFCE [4] оточення робочого столу, серед інших. Функціональність раніше втілені в HAL була інтегрована в Udev себе, або винесена в окремий програми, такі як udisks і upower .

Архітектура

Система розділена на три частини: Бібліотека libudev, що дозволяє отримати доступ до інформації про пристрій. Udevd демон, у просторі користувача, який керує віртуальної / Dev. Адміністративних udevadm команди для діагностики. Система отримує дзвінки від ядра через NetLink сокет. Більш ранні версії використовували автовизначення , додавши посилання на себе в / і т.д. / hotplug.d / за замовчуванням з цією метою.

Udev — менеджер пристроїв для нових версій ядра Linux. Його основне завдання — обслуговування файлів пристроїв у теці /dev і обробка всіх дій, які

виконуються в просторі користувача при додаванні/відключенні зовнішніх пристроїв, включаючи завантаження `firmware`. Завдяки `udev`, в каталозі `/dev` містяться файли тільки тих пристроїв, які зараз підключені до системи. Кожен пристрій має свій відповідний файл. Якщо пристрій відключається від системи, то його файл вилучається. Вміст каталогу `/dev` зберігається на віртуальній файлової системі і всі файли, що перебувають в ньому, створюються при кожному запуску системи. Модифіковані або створені вручну файли не зберігаються після перезавантаження. Файли і теки, які необхідно зберегти або які завжди повинні бути присутніми в каталозі `/dev`, незалежно від стану пристрою, необхідно поміщати в каталог `/lib/udev/devices`. При запуску системи вміст цього каталогу копіюється в `/dev` як `ε` (з тими ж правами доступу). Таким чином, на відміну від традиційної для Unix схеми управління каталогом `/dev`, де його вміст був статичним набором файлів, `udev` забезпечує динамічне управління пристроями, котрі дійсно перебувають у системі. Хоча раніше подібна функціональність забезпечувалася `devfs`, прихильники `udev` відзначають такі переваги свого рішення:

- `udev` підтримує незмінне іменування пристроїв, не залежне, наприклад, від порядку включення пристроїв в систему. Конфігурація за замовчуванням забезпечує незмінне іменування для накопичувачів: тверді диски ідентифікуються за унікальним ідентифікатором файлової системи, іменем диска і розташуванням пристрою, до якого підключений диск.

- `udev` працює цілком у просторі користувача. Це призводить, зокрема, до того, що `udev` може виконувати довільні програми для укладання назви пристрою або його ініціалізації. Принцип роботи. `udev` запускається як демон і приймає через сокет `netlink` події `uevents` від ядра, які генеруються при ініціалізації або вилученні пристрою з системи. Запрограмованих користувачем (системою) правила зв'язуються з властивостями події та відповідного пристрою, і правило, що збіглося, (яких може бути кілька) може назвати і створити відповідний файл пристроїв, а також виконати інші програми для ініціалізації і конфігурації пристрою. Наприклад, таким чином можна реалізувати автоматичне монтування зовнішніх накопичувачів при їхньому підключенні. Правила можуть зв'язуватися з таким властивостям, як конкретна ядерна підсистема, ім'я пристрою в ядрі, фізичне розташування пристрою, або за серійним номером пристрою. Правила також можуть запитувати інформацію за допомогою інших програм або вказати, що ім'я пристрою завжди буде одним і тим же, незалежно від порядку виявлення пристроїв системою.

Типовий спосіб використання `udev` на Linux-системі — дозволити посилати події HAL або DeviceKit, щоб вони виробили подальші залежні від пристроїв дії. Наприклад, HAL/DeviceKit може повідомити інші програми про новий пристрій за допомогою широкомовного повідомлення в D-Bus. Таким чином, робочі середовища типу GNOME або KDE можуть автоматично змонтувати USB-накопичувач і відкрити файлової менеджер для перегляду його вмісту. Архітектура. `udev` розділений на три частини:

- Бібліотека `libudev`, що дозволяє отримувати доступ до інформації про пристрої
- Демон `udev`, що працює у просторі користувача та керує вмістом `/dev`
- Програма `udevadm` для налагодження та діагностики. `udev` отримує події від ядра через `netlink-socket`. Ранні версії використовували `hotplug`, додаючи в `/etc/hotplug.d/default`.

Робота зі службами

Для управління службами в багатьох системах існує команда `service`. Однак, в Linux її немає. Тому була написана простенька пара прог, які забезпечують управління службами з командного рядка і легко вбудовуються в будь-який Linux. Для управління службами у дистрибутиві в консолі наберіть

```
$ srv <Служба> <Дія>
```

де <Служба> - будь-який з сервісів, доступних в системі, а <Дія> - потрібну дію для цієї служби, звичайно `start`, `stop` або `restart`. Ім'я служби і дія є необов'язковими полями. Якщо зазначений тільки один аргумент, буде виведений список запущених в даний момент служб. Якщо не вказано нічого, то до списку запущених служб буде доданий весь перелік існуючих служб, які можна запустити а також сформовано список груп служб. Ця пара програм розуміє і такий формат:

```
srv switcher <Дія> <Група>
```

або

```
$ Srv <Група> <Дія>
```

Обидва формати є рівнозначними. У першому випадку ми явно вказуємо, що служба `switcher` повинна виконати <Дія> для <Групи> служб, прописаних в `/etc/conf.d/switcher`. У другому випадку програма `srv` не знаходячи службу з ім'ям <Група> припускає, що це не назва служби а назва групи і рекурсивно запускає себе першою командою. Якщо у завантажувачі вказати додатковий параметр ядра:

```
kernel / vmlinuz26-lts root = / dev/disk/sda1 ro vga = 791  
nomodeset type = boot </strong>,
```

а в список служб, що запускаються разом з системою помістити `switcher`, то при запуску `switcher` буде додатково завантажена група служб, які були перераховані в якості мети. Це може виявитися зручним, якщо система має багатопротильне призначення і потрібно на етапі завантаження вибрати в якій якості повинна працювати система. Це також може виявитися корисним, якщо хочете передбачити резервний варіант входу в систему, коли більша частина служб (включаючи сервер `X.org`) не вантажиться.

ТЕМА 5. ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ В ОПЕРАЦІЙНИХ СИСТЕМАХ.

5.0 ОСНОВНІ ТЕРМІНИ І ПОНЯТТЯ

НСД – Несанкціонований доступ

Вибіркове керування доступом (англ. *Discretionary access control*, DAC; рос. избирательное управление доступом) — керування доступом суб'єктів до об'єктів на основі списків керування доступом або матриці доступу. Інші назви: дискреційне керуванням доступом, контрольоване керування доступом або розмежувальне керування доступом.

5.1 ОСНОВНІ ФУНКЦІЇ ЗАХИСТУ

Підсистема захисту ОС виконує такі основні функції:

1. Ідентифікація та аутентифікація. Жоден користувач не може почати роботу з ОС, не ідентифікувавши себе і не надавши системі інформації, що підтверджує, що користувач дійсно є тим, ким він себе заявляє.

2. Розмежування доступу. Кожен користувач системи має доступ тільки до тих об'єктів ОС, до яких йому надано доступ відповідно до поточної політики безпеки.

3. Аудит ОС реєструє в спеціальному журналі події, потенційно небезпечні для підтримки безпеки системи.

4. Управління політикою безпеки. Політика безпеки повинна постійно підтримуватися в адекватному стані, тобто повинна гнучко реагувати на зміни умов функціонування ОС. Управління політикою безпеки здійснюється адміністраторами системи з використанням відповідних засобів, вбудованих в ОС.

5. Криптографічні функції. Захист інформації немислима без використання криптографічних засобів захисту. Шифрування використовується в ОС при зберіганні і передачі по каналах зв'язку паролів користувачів і деяких інших даних, критичних для безпеки системи.

6. Мережеві функції. Сучасні ОС, як правило, працюють не ізольовано, а в складі локальних і / або глобальних комп'ютерних мереж. ОС комп'ютерів, що входять в одну мережу, взаємодіють між собою для вирішення різних завдань, у тому числі і завдань, що мають пряме відношення до захисту інформації.

Підсистема захисту зазвичай не представляє собою єдиний програмний модуль. Як правило, кожна з перерахованих функцій підсистеми захисту вирішується одним або декількома програмними модулями. Деякі функції вбудовуються безпосередньо в ядро ОС. Між різними модулями підсистеми захисту повинен існувати чітко визначений інтерфейс, використовуваний при взаємодії модулів для вирішення спільних завдань.

У таких ОС, як Windows, підсистема захисту чітко виділяється в загальній архітектурі ОС, в інших, як UNIX, захисні функції розподілені практично по всіх елементах ОС. Зазвичай підсистема захисту ОС допускає розширення додатковими програмними модулями.

Головні захисні механізми ОС Windows

- можливість захищеного входу в систему з ідентифікацією користувачів;
- контроль доступу до ресурсів по категоріях користувачів;
- аудит.

Головні захисні механізми ОС сімейства UNIX

- ідентифікації й аутентифікації користувача при вході в систему;
- розмежуванні прав доступу до файлової системи, у базі якого лежить реалізація *дискреційної* моделі доступу;
- аудит, іншими словами реєстрація подій.

Для різних клонів ОС сімейства UNIX можливості пристроїв захисту можуть некардинально різнитися, але будемо розглядати ОС UNIX у загальному випадку, без врахування якихось незначущих особливостей окремих ОС цього сімейства.

5.2 АНТИВІРУСНИЙ ЗАХИСТ.

Основу існуючих принципів контролю доступу до ресурсів, становить завдання прав і реалізація розмежувань на підставі певних правил доступу суб'єктів "КОРИСТУВАЧ" до об'єктів, зокрема, до файлових об'єктів. Сам процес, і може нести в собі уразливість по наступних причинах:

- Несанкціоновані (сторонні) процеси. Це процеси, які не потрібні користувачеві для виконання своїх службових обов'язків і можуть несанкціоновано встановлюватися на комп'ютер (локально, або віддалено) з різними цілями (наприклад, так звані, шпигунські програми), у тому числі, і з метою здійснення НСД до інформації;
- Критичні процеси - ті процеси, які запускаються в системі із привілейованими правами, наприклад, під обліковим записом System або root, а також процеси, які найбільше ймовірно можуть бути піддані атакам, у першу чергу, це мережеві служби. Атаки на подібні процеси (як правило, мережні атаки) найбільш критичні, що пов'язане з можливістю розширення привілеїв, у межі – одержання повного керування системою (тому що ОС не забезпечує можливості в необхідному обсязі встановлювати розмежування прав доступу для користувача System або root). У якості зауваження відзначимо, що на сьогоднішній день суттєво зросла частка атак, спрямованих на уразливість не властиво в ОС, а на уразливість застосувань. Однак застосування користуються сервісами ОС, які повинні захищатися ОС, тобто уразливість застосування при коректній реалізації механізмів захисту ОС, не повинні приводити до можливості несанкціонованого доступу до інформації;
- Скомпрометовані процеси – процеси, що містять помилки (уразливості), що стали відомими, використання яких дозволяє здійснити НСД до інформації. Віднесення даних процесів в окрему групу обумовлене тим, що з моменту виявлення уразливості й до моменту усунення її розроблювачем системи або

застосування, може пройти кілька місяців. Протягом цього часу в системі перебуває відома уразливість, тому система не захищена;

- Процеси, що апріорі володіють недекларованими (документально не описаними) властивостями. До цієї групи ми віднесемо процеси, що є середовищем виконання (насамперед, це віртуальні машини, що є середовищем виконання для скриптів і аплетів, і офісні застосування, що є середовищем виконання для макросів).

Висновок.

З урахуванням наведеної класифікації, з якої випливає, що саме процес несе в собі загрозу вірусної атаки, може бути сформульована задача контролю доступу до ресурсів з урахуванням довіри до процесів або довірчого контролю доступу до ресурсів.

Природно, що облік довіри до процесів при контролі доступу до ресурсів можливий тільки в тому випадку, якщо процес можна розглядати, як самостійний суб'єкт доступу, для якого можна задавати власні правила розмежування доступу до ресурсів, а не як суб'єкт, що успадковує права доступу користувача (а якщо ні, то, слід говорити про довіру до користувача, а не до процесу).

Для формулювання задач антивірусного захисту приведемо укрупнену класифікацію вірусних атак:

1. "Шкідливі програми" (трояни і т.п.). Окремі програми, які виконують ті або інші деструктивні/несанкціоновані дії.

2. "Віруси". Програми, що звичайно не мають власного модуля, що виконується, і "живучі" (як правило, зараження здійснюється по засобом їх приєднання до виконуваного файлу) усередині іншого файлового об'єкта або частини фізичного носія.

3. "Хробаки". Різновид 1,2,4, що використовує мережеві можливості для зараження.

4. "Макро-Віруси" (скриптові віруси) - програми, для виконання яких потрібне певне середовище виконання (командний інтерпретатор, віртуальна машина й т.п.). У цю ж групу можемо віднести й офісні застосування, що дозволяють створювати й підключати макроси.

Наведена класифікація підтверджує, що проблеми антивірусного захисту у великій мері зводяться до питання апріорної довіри до процесів, як наслідок, основу їх розв'язку становить реалізація довірчого контролю доступу до ресурсів.

Висновки.

1. Віднесення задач антивірусного захисту до класу самостійних задач захисту інформації, у загальному випадку, необгрунтовано – це лише деяка підмножина задач захисту інформації від НСД.
2. Як наслідок, сказаного вище – механізмами захисту інформації від НСД, насамперед, механізмами контролю доступу до ресурсів, може бути здійснений ефективний антивірусний захист.

Антивіруси:

5.3 МЕРЕЖЕВИЙ ФІЛЬТР

Мережевий фільтр (екран) призначений для заборони чи дозволу користування певними сервісами зовнішньої мережі комп'ютерів чи їх групі, а також навпаки. Перш за все мережеві фільтри поділяються на персональні та міжмережні. Персональний мережевий фільтр призначений для захисту одного комп'ютера, в той час коли міжмережні використовують для цілої робочої групи чи підмережі. Мережеві фільтри бувають програмні, а також апаратні (тільки міжмережні). Апаратні мережеві фільтри підключають у вигляді окремого пристрою до мережі.

Функції мережевих фільтрів:

- Фільтрація пакетів. У цьому випадку виконуються зовсім прості функції (фактично як у спеціалізованого маршрутизатора), які полягають у перегляді заголовка кожного пакету та перевірки IP адреси та порта на правильність. Наприклад, пакет, який передається в Інтернет, але має локальну адресу призначення, блокується. Ця функція присутня у всіх сучасних мережевих фільтрах і відрізняється високою швидкістю та відсутністю необхідності діалогу з користувачем.

- Проксі сервер. Це друга загальновідома функція. Різниця між проксі сервером та фільтрацією пакетів полягає в тому, що проксі сервер вимагає, щоб всі сеанси зв'язку встановлювались через нього, а не напряму. Отримавши дані, проксі сервер відсилає їх далі від свого імені, а отримавши відповідь, пересилає її потрібному комп'ютеру внутрішньої мережі. Як можна здогадатись, продуктивність не є такою ж високою, як при фільтрації пакетів, але достатньою, оскільки відбувається тільки заміна заголовка пакета. Головною перевагою проксі сервера є те, що він приховує справжні адреси ваших комп'ютерів, замінюючи їх на свою. Це унеможливорює атаку на конкретний комп'ютер в підмережі.

- Проксі сервер програм. Це третя функція. Цей різновид проксі сервера відрізняється "розумінням" протоколів програм, що ведуть передачу даних. Хороший приклад такого сервера –поштовий сервер. Проксі сервер програм може проводити ідентифікацію користувачів, а не тільки покладатись на IP адресу, в деяких випадках можливе виявлення передавання вірусів та іншого шкідливого коду. Це все звичайно потребує набагато більших обчислювальних потужностей, а також в більшості випадків вимагає значного перенастроювання робочих станцій, через що втрачається прозорість роботи мережевого фільтру.

- Кешування даних. Це не є традиційною функцією мережевих фільтрів, але на даний час стає надзвичайно популярною властивістю. Ідея полягає у тому, що, оскільки всі дані проходять через мережевий фільтр, він може зберігати найбільш популярну інформацію і при наступному звертанні за нею видати її зі свого кешу.

- Статистика та повідомлення. Важливою властивістю мережевого фільтру є ведення історії всіх мережевих з'єднань, а також вивід повідомлень про атаки на мережу чи комп'ютер. Історія з'єднань допомагає правильно настроїти мережевий фільтр, щоб комп'ютер був одночасно захищений від нападів і відкритий для доступу авторизованим користувачам.

- Керування. Для персональних мережевих фільтрів основна характеристика – зручність їхнього налаштування. Керування міжмережними фільтрами здебільшого відбувається дистанційно (використовуючи HTML інтерфейс чи інший), що потребує впевненості в надійності авторизації та каналу зв'язку.