3.1 ВИМОГИ, ЩО ПРЕД'ЯВЛЯЮТЬСЯ ДО ОПЕРАЦІЙНОЇ СИСТЕМИ

Операційна система є серцевиною програмного забезпечення, вона створює середовище для виконання додатків і багато в чому визначає, які корисні для користувача властивості матимуть ці додатки. У зв'язку з цим розглянемо вимоги, яким повинна задовольняти сучасна ОС.

Очевидно, що головною вимогою, що пред'являється до операційної системи, є здатність виконання основних функцій: ефективного управління ресурсами і забезпечення зручного інтерфейсу для користувача і прикладних програм. Сучасна ОС, як правило, повинна реалізовувати мультипрограмну обробку, віртуальну пам'ять, свопінг, підтримувати багатовіконний інтерфейс, а також виконувати багато інших, абсолютно необхідних функцій.

Окрім цих вимог функціональної повноти до операційних систем пред'являються не менш важливі експлуатаційні і ринкові вимоги, які наведені нижче.

Розширюваність. Код ОС має бути написаний так, щоб можна було легко внести доповнення і зміни, якщо це знадобиться, і не порушити цілісність системи.

Переносимість. Код ОС повинен легко переноситися з процесора одного типу на процесор іншого типу і з апаратної платформи одного типу на апаратну платформу іншого типу.

Надійність і відмовостійкість. Система має бути захищена як від внутрішніх, так і від зовнішніх помилок, збоїв і відмов. Її дії мають бути завжди передбачуваними, а додатки не зможуть завдавати шкоди ОС. Надійність і відмовостійкість ОС передусім визначаються архітектурними рішеннями, покладеними в її основу, а також якістю її реалізації. Крім того, важливо, чи включає ОС програмну підтримку апаратних засобів забезпечення відмовостійкості, таких, наприклад, як дискові масиви або джерела безперебійного живлення.

Сумісність. ОС повинна мати засоби для виконання прикладних програм, написаних для інших операційних систем. Крім того, призначений для користувача інтерфейс має бути сумісний з існуючими системами і стандартами.

Захист інформації і безпека. ОС повинна мати засоби захисту ресурсів одних користувачів від інших.

Продуктивність. Система повинна мати настільки хорошу швидкодію і час реакції, наскільки це дозволяє апаратна платформа. На продуктивність ОС впливає багато чинників, серед яких основними є архітектура ОС, різноманіття функцій, якість програмування коду, можливість виконання ОС на високопродуктивній (багатопроцесорній) платформі тощо.

Постійне зростання вимог до ОС призводить не лише до удосконалення їх архітектури, але і до нових способів їх організації. Як вже відзначалося вище, для задоволення вимог, що пред'являються до сучасної ОС, велике значення має її структурна побудова. Операційні системи пройшли тривалий шлях розвитку від монолітних систем до добре структурованих модульних систем, здатних до розвитку, розширення і легкого перенесення на нові платформи. В експериментальних і комерційних ОС були випробувані найрізноманітніші підходи і структурні елементи, більшість з яких можна об'єднати в наступні категорії:

1. Архітектура ядра.
2. Множинні прикладні середовища.
3. Концепція віртуальних машин.
4. Мережеві і розподілені ОС.
5. Симетрична багатопроцесорність.
6. Багатопоточність.

Розглянемо детальніше деякі з цих вимог і структурну побудову сучасних операційних систем.

3.1.1 Розширюваність

Тоді як апаратна частина комп'ютера застаріває за декілька років, корисне життя операційних систем може вимірюватися десятиліттями. Тому операційні системи завжди еволюційно змінюються з часом, і ці зміни значиміші, ніж зміни апаратних засобів. Зміни ОС є набуттям нею нових властивостей. Наприклад, підтримка нових пристроїв, таких як CD-ROM, можливість зв'язку з мережами нового типу, підтримка багатообіцяючих технологій, таких як графічний інтерфейс користувача або об'єктно-орієнтоване програмне оточення, використання більш ніж одного процесора. Тому збереження цілісності коду, які б зміни не вносилися в операційну систему, є головною метою розробки.

Розширюваність може досягатися за рахунок модульної структури ОС, при якій програми будуються з набору окремих модулів, що взаємодіють тільки через функціональний інтерфейс. Нові компоненти можуть бути додані в операційну систему модульним шляхом, вони виконують свою роботу, використовуючи інтерфейси, підтримувані існуючими компонентами.

Використання об'єктів для представлення системних ресурсів також покращує розширюваність системи. **Об'єкти** – це абстрактні типи даних, над якими можна виконувати тільки ті дії, які передбачені спеціальним набором об'єктних функцій. Об'єкти дозволяють однаково управляти системними ресурсами. Додавання нових об'єктів не руйнує існуючі об'єкти і не вимагає змін існуючого коду.

Прекрасні можливості для розширення надає підхід до структуризації ОС за типом клієнт-сервер з використанням технології мікроядра. Відповідно до цього підходу ОС будується як сукупність привілейованої програми, що управляє, і набору непривілейованих послуг-серверів. Основна частина ОС може залишатися незмінною в той час, як можуть бути додані нові сервери або удосконалені старі.

3.1.2 Переносимість

В ідеалі код ОС повинен легко переноситися з процесора одного типу на процесор іншого типу, і з апаратної платформи одного типу на апаратну платформу іншого типу. Переносимі ОС мають декілька варіантів реалізації для різних платформ, таку властивість ОС називають також багатоплатформністю (кросплатформністю, мультиплатформністю).

Вимога переносимості коду тісно пов'язана також з розширюваністю. Розширюваність дозволяє покращувати операційну систему, тоді як переносимість дає можливість переміщати усю систему на машину, що базується на іншому процесорі або апаратній платформі, роблячи при цьому по можливості невеликі зміни в коді. Написання переносимої ОС аналогічно написанню будь- якого переносимого коду. Для цього треба дотримуватись деяких правил.

По-перше, велика частина коду має бути написана мовою, яка є на усіх машинах, куди треба переносити систему. Це означає, що код має бути написаний на мові високого рівня, переважно стандартизованою, наприклад, на мові С. Програма,

написана на асемблері, не є переносимою, якщо тільки переносити її на машину, що має командну сумісність з машиною, з якої переносять додаток.

По-друге, слід врахувати, в яке фізичне оточення програма має бути перенесена. Різна апаратура вимагає різних рішень при створенні ОС. Наприклад, ОС, побудована на 32-бітових адресах, не може бути перенесена на машину з 16-бітовими адресами (хіба що з величезними труднощами).

По-третє, важливо мінімізувати або, якщо можливо, виключити ті частини коду, які безпосередньо взаємодіють з апаратними засобами.

По-четверте, якщо код, залежний від апаратури, не може бути повністю виключений, то він має бути ізольований в декількох модулях, що добре локалізуються.

Для легкого перенесення ОС при її розробці мають бути дотримані такі вимоги:

1. Переносима мова високого рівня. Більшість переносимих ОС написані на мові С (стандарт ANSI X3.159-1989). Розробники вибирають мову С тому, що вона стандартизована, і тому, що С-компілятори широко доступні. Асемблер використовується тільки для тих частин системи, які повинні безпосередньо взаємодіяти з апаратурою (наприклад, обробник переривань) або для частин, які вимагають максимальної швидкості (наприклад, цілочисельна арифметика підвищеної точності). Проте непереносний код має бути ретельно ізольований усередині тих компонентів, де він використовується.

2. Ізоляція процесора. Деякі низькорівневі частини ОС повинні мати доступ до структур даних і регістрів, залежних від процесора. Проте код, який робить це, повинен міститися в невеликих модулях, які можуть бути замінені аналогічними модулями для інших процесорів.

3.1.3 Сумісність

Існує декілька «довгоживучих» популярних операційних систем (різновиди UNIX, MS-DOS, Windows, OS/2), для яких напрацьована широка номенклатура додатків. Деякі з них користуються широкою популярністю. Тому для користувача, що переходить з однієї ОС або на іншу апаратну платформу, дуже приваблива можливість запуску в новому середовищі звичних додатків. Якщо ОС має засоби для виконання прикладних програм, написаних для ОС, то про неї говорять, що вона має сумісність з

цими ОС. Поняття сумісності включає також підтримку інтерфейсів користувача інших ОС.

Слід розділяти питання двійкової сумісності і сумісності на рівні початкових текстів додатків. Двійкова сумісність досягається в тому випадку, коли можна взяти виконувану програму і запустити її на виконання на іншій ОС. Для цього потрібні: сумісність на рівні команд процесора, сумісність на рівні системних викликів і навіть на рівні бібліотечних викликів, якщо вони є динамічно зв'язуваними.

Сумісність на рівні початкових текстів вимагає наявності відповідного компілятора в складі програмного забезпечення, а також сумісності на рівні бібліотек і системних викликів. При цьому потрібна перекомпіляція наявних початкових текстів в новий виконуваний модуль.

Сьогодні таку сумісність забезпечує стандартизація розробки ПО:

- наявність стандарту на мови програмування;
- наявність стандарту на інтерфейс операційних систем.

Роботи щодо стандартизації інтерфейсу ОС відбуваються в рамках проекту POSIX (Portable Operating System Interfice – Переносимий Інтерфейс ОС). Найважливішим стандартом є POSIX 1003.1, який описує набір бібліотечних процедур (відкриття файлу, створення нового процесу і тому подібне), які можуть бути реалізовані в системі. Цей процес стандартизації триває і в наші дні. Останньою модифікацією стандарту є базова специфікація Open Group/IEEE. Ці стандарти відображають традиційний набір засобів, реалізований в UNIX- сумісних системах.

Так, функції бібліотеки підсистеми Win32 API виконуються в режимі користувача і в режимі ядра. Наприклад, до виходу Windows NT 4.0 (1996 р.) віконна і графічна системи працювали в режимі користувача як частину процесу підсистеми Win32. Потім для підвищення продуктивності реалізацію цих підсистем було перенесено в режим ядра.

Підсистема POSIX працює в режимі користувача і реалізує набір функцій, визначених стандартом POSIX 1003.1. Оскільки додатки, написані для однієї підсистеми, не можуть використати функції інших підсистем, в POSIX- програмах не можна користуватися засобами Win32 API (зокрема, графічними і мережевими

функціями), що знижує значущість цієї підсистеми. Підсистема POSIX не є обов'язковим компонентом Windows XP.

Чи має нова ОС двійкову сумісність або сумісність початкових текстів з існуючими системами, залежить від багатьох чинників. Найголовніший з них – архітектура процесора, на якому працює нова ОС. Якщо процесор, на який переноситься ОС, використовує той же набір команд (можливо з деякими доповненями) і той же діапазон адрес, тоді двійкова сумісність може бути досягнута досить просто.

Набагато складніше досягти двійкової сумісності між процесорами, заснованими на різній архітектурі. Для того щоб один комп'ютер виконував програми іншого (наприклад, DOS-програму на Mac), цей комп'ютер повинен працювати з машинними командами, які йому спочатку незрозумілі.

Виходом в таких випадках є використання так званих прикладних середовищ. Основну частину програми, як правило, складають виклики бібліотечних функцій. Прикладне середовище імітує бібліотечні функції, використовуючи заздалегідь написану бібліотеку функцій аналогічного призначення, а інші команди емулює кожну окремо за допомогою програми- емулятора.

3.1.4 Надійність, захист інформації і безпека

У жовтні 1988 року в США сталася подія, названа фахівцями найбільшим порушенням безпеки американських комп'ютерних систем з тих, що коли-небудь траплялися. 23-річний студент випускного курсу Корнельского університету Роберт Таппан Морріс запустив в комп'ютерній мережі ARPANET програму, представляючу собою різновид комп'ютерних вірусів – мережевих черв'яків. В результаті атаки був повністю або частково заблокований ряд загальнонаціональних комп'ютерних мереж. У результаті вірус уразив більше 6200 комп'ютерних систем по всій Америці. Загальний збиток від цієї атаки був оцінений фахівцями мінімум в 100 мільйонів доларів. Р. Морріс був виключений з університету з правом повторного вступу через рік, і засуджений судом до штрафу в 270 тис. доларів і трьох місяців ув'язнення.

Важливість вирішення проблеми інформаційної безпеки нині загальновизнана, підтвердженням чому служать гучні процеси про порушення цілісності систем.

Проблема забезпечення безпеки носить комплексний характер, для її вирішення потрібне поєднання законодавчих, організаційних і програмно-технічних заходів.

Технічні засоби реалізуються програмним і апаратним забезпеченням і розв'язують різні задачі з захисту і можуть бути вбудовані в операційні системи, або можуть бути реалізовані у вигляді окремих продуктів.

Система має бути захищена як від внутрішніх, так і від зовнішніх помилок, збоїв і відмов. Її дії мають бути завжди передбачуваними, а додатки не повинні мати можливості завдавати шкоди ОС. Надійність і відмовостійкість ОС передусім визначаються архітектурними рішеннями, покладеними в її основу, а також якістю її реалізації (відлагодженого коду). Крім того, важливо, чи включає ОС програмну підтримку апаратних засобів забезпечення відмовостійкості, таких, наприклад, як дискові масиви або джерела безперебійного живлення.

Із зростанням популярності систем розподілу часу виникла проблема захисту інформації. Це пов'язано зі збільшеною цінністю інформації, що обробляється комп'ютерами, а також з підвищеним рівнем загроз, існуючих при передачі даних мережами, особливо публічними, таким як Інтернет. Багато операційних систем мають сьогодні розвинені засоби захисту інформації, засновані на шифруванні даних, аутентифікації і авторизації. У комп'ютери і ОС були вбудовані деякі інструменти загального призначення, що підтримують різні механізми захисту і забезпечують безпеку, які умовно можна поділити на дві категорії:

1. Контроль над доступом. Пов'язаний з регулюванням доступу користувача до системи в цілому, до її підсистем і даних, а також до різних ресурсів.
2. Контроль над переміщенням інформації. Регулювання потоку даних усередині системи і при їх доставці користувачеві.

Правила безпеки визначають такі властивості, як захист ресурсів одного користувача від інших і встановлення квот на ресурси для запобігання захоплення одним користувачем усіх системних ресурсів (таких як пам'ять).

Забезпечення захисту інформації від несанкціонованого доступу є обов'язковою функцією операційних систем. У більшості популярних систем гарантується міра безпеки даних, що відповідає рівню С2 системи стандартів США (Windows NT, окремі реалізації Unix та ін.).

Основи стандартів в області безпеки були закладені в документі «Критерії оцінки надійних комп'ютерних систем». Цей документ, виданий в США в 1983 році національним центром комп'ютерної безпеки (NCSC – National Computer Security Center), часто називають Помаранчевою Книгою (за кольором обкладинки).

Відповідно до вимог Помаранчевої Книги безпечною вважається така система, яка «за допомогою спеціальних механізмів захисту контролює доступ до інформації таким чином, що особи, які мають тільки відповідні повноваження, або процеси, виконуються від їх імені, можуть отримати доступ на читання, запис, створення або видалення інформації».

Ієрархія рівнів безпеки, наведена в Помаранчевій Книзі, позначає нижчий рівень безпеки як D, а вищий – як A. Вважається, що такі ОС, як MS-DOS, Mac OS, OS/2, мають рівень захищеності D.

Основними властивостями, характерними для С-систем, є наявність підсистеми обліку подій, пов'язаних з безпекою, і вибірковий контроль доступу. Рівень С ділиться на 2 підрівні: рівень С1, що забезпечує захист даних від помилок користувачів, але не від дій зловмисників, і більш строгий рівень С2. На рівні С2 мають бути присутніми засоби секретного входу, що забезпечують ідентифікацію користувачів шляхом введення унікального імені і пароля перед тим, як їм буде дозволений доступ до системи. Виборковий контроль доступу, потрібний на цьому рівні, дозволяє власникові ресурсу визначити, хто має доступ до ресурсу і що він може з ним робити. Власник робить це шляхом надання прав доступу користувачеві або групі користувачів. Засоби обліку і спостереження (auditing) – забезпечують можливість виявити і зафіксувати важливі події, пов'язані з безпекою, або будь-які спроби створити, отримати доступ або видалити системні ресурси. Захист пам'яті – на цьому рівні система не захищена від помилок користувача, але поведінка його може бути проконтрольована за записами в журналі, залишеними засобами спостереження.