

1.6 СИСТЕМИ З РЕЖИМОМ РОЗПОДІЛУ ЧАСУ

При системах 3-го покоління ще залишався різновид пакетної обробки. Багато програмістів сумували за першим поколінням машин, коли вони могли годинами розпоряджатися усією машиною і досить швидко відлагоджувати свої програми.

Бажання скоротити час очікування відповіді, і для того щоб хоч би частково повернути користувачам відчуття безпосередньої взаємодії з комп'ютером, був розроблений інший варіант мультипрограмних систем – системи *розподілу часу*, варіант багатозадачності, при якому в кожного користувача є свій термінал.

Варіант мультипрограмування, який застосовувався в системах розподілу часу, був націлений на створення для кожного окремого користувача ілюзії одноосібного володіння обчислювальною машиною за рахунок періодичного виділення кожній програмі своєї частки процесорного часу. Якщо двадцять користувачів зареєстровані в системі, що працює в режимі розподілу часу, і сімнадцять з них думають, розмовляють або п'ють каву, то центральний процесор (ЦП) по черзі надається трьом користувачам, що працюють в цей час на машині, і забезпечує швидке інтерактивне обслуговування цих користувачів. При цьому він може працювати над великими пакетами у фоновому режимі, коли ЦП не зайнятий іншими завданнями.

Перша серйозна система з режимом розподілу часу **CTSS** (Compatible Time Sharing System – сумісна система розподілу часу) для управління обчислювальними ресурсами комп'ютера IBM 7090 була розроблена на початку 60-х років групою програмістів під керівництвом професора **Фернандо Корбато** в Массачусетському технологічному інституті (МТІ). Головним розробником був математик **Джон Маккарті**.

Після успіху системи CTSS та ж група з МТІ, дослідницька лабораторія Bell Labs і корпорація General Electric у 1965 почали розробку машини, яка повинна була підтримувати одночасно сотні користувачів в режимі розподілу часу. Проектувальники цієї системи, відомої як **MULTICS** (MULTiplexed Information and Computing Service – мультиплексна інформаційна і обчислювальна служба), уявляли собі одну величезну машину, скористатися послугами якої могла кожна людина в районі Бостона. Думка про те, що набагато потужніші машини, чим їх мейнфрейм

GE-645, продаватимуться мільйонами за ціною тисяча доларів і менше за штуку усього лише через тридцять років, здавалася чистісінькою науковою фантастикою.

Існувало багато причин, в яких система MULTICS не захопила весь світ. Не останню роль зіграв той факт, що система була написана мовою високого рівня PL/1, а нормальний компілятор мови PL/1 з'явився лише через декілька років. У результаті, система MULTICS подала багато конструктивних ідей комп'ютерним теоретикам, але перетворити її на серйозний продукт і добитися комерційного успіху виявилось набагато важче, ніж очікувалося. Саме розробники MULTICS почали застосовувати термін *«процес»* в його сучасному значенні в контексті ОС.

Уперше в MULTICS почали застосовувати *віртуальну пам'ять* – метод управління пам'яттю комп'ютера, що дозволяє виконувати програми, що вимагають більше оперативної пам'яті, ніж є в комп'ютері, шляхом автоматичного переміщення частин програми між основною пам'яттю і вторинним сховищем (наприклад, жорстким диском).

Але усі спроби налагодити в системі відносно дружній інтерфейс провалилися. Було вкладено багато грошей, а результат був дещо іншим, ніж хотілося б керівництву з Bell Labs. Проект був закритий. До речі, учасниками проекту значилися **Кен Томпсон** і **Денніс Рітчі**. Попри те, що проект був закритий, вважається, що саме ОС MULTICS дала початок ОС Unix.

До основних досягнень цього періоду в області розробки систем, що працюють в режимі розподілу часу, належать також TSS (Система з розподілом часу), розроблена компанією IBM, і CP/CMS (діалогова моніторна система, IBM), яка потім перетворилася на ОС віртуальних машин (VM). MULTICS, TSS і CP/CMS – усі ці системи мали вже віртуальну пам'ять.

За свій вклад в розробку CTSS і MULTICS в 1990 році Ф. Корбато був удостоєний престижної премії Т'юринга.

Незабаром Bell Labs вибула з проекту, а корпорація General Electric зовсім залишила комп'ютерний бізнес. Проте Массачусетський технологічний інститут проявив завзятість і з часом отримав працюючу систему. Вона була продана як комерційний виріб компанії Honeywell, що купила комп'ютерний бізнес в General

Electric, і встановлена приблизно у вісімдесяти великих компаніях і університетах по всьому світу. Незважаючи на свою нечисленність, користувачі системи були відчайдушно віддані їй. Так, наприклад, компанії General Motors, Ford залишили свої системи в кінці 90-х років, через 30 років після їх виходу. Останній комп'ютер під управлінням MULTICS працював до 2000 року.

Ще одним важливим моментом розвитку за часів 3-го покоління було феноменальне зростання міні-комп'ютерів, починаючи з випуску PDP-1 корпорацією DEC у 1961 році. Ці комп'ютери мали оперативну пам'ять, що складалась усього лише з 4К 18-бітових слів, але коштували вони всього по 120 тисяч доларів (близько 5% від ціни IBM-7094). За цією машиною послідувала ціла серія інших, і як кульмінація – PDP-11.

У 1969 році **Кен Томпсон** (Ken Thompson), один з розробників MULTICS з Bell Labs, знайшов міні-комп'ютер PDP-7, яким ніхто не користувався, і вирішив написати усічену, розраховану на одного користувача, версію системи MULTICS. Пізніше ця система розвинулася в ОС UNIX, що стала популярною в академічному світі і в урядових установах. Системний час реалізації UNIX відлічують з 1 січня 1970 року.

Надалі ще один дослідник лабораторії Bell Labs, **Брайан Керніган**, якимось жартома назвав цю систему **UNICS** (UNiplexed Information and Computing Service – примітивна інформаційна і обчислювальна служба). Прізвисько, дане Керніганом, міцно пристало до нової системи, хоча написання цього слова стало злегка коротше при цій же вимові, перетворившись на **UNIX**.

Робота Томпсона справила на його колег з Bell Labs таке сильне враження, що незабаром до нього приєднався **Денніс Рітчі**, а трохи пізніше і увесь його відділ. Незабаром система UNIX була перенесена на потужніші комп'ютери PDP-11/20,45,70.

У 1974 році Рітчі і Томпсон опублікували статтю про ОС UNIX, яка стала важливою віхою в історії комп'ютерів. За роботу, описану в цій статті, їм пізніше асоціацією з обчислювальної техніки ACM була присуджена премія Т'юрінга.

Експериментальні системи з розподілом часу в 70-х роках перетворилися на успішний комерційний продукт. Зв'язок між комп'ютерними системами США ставав

все інтенсивнішим у міру того, як усе ширше застосування отримували комунікаційні стандарти ТСР/IP Міністерства Оборони. Зв'язок локальною мережею став можливим і недорогим завдяки стандарту Ethernet, розробленому компанією Xerox.

Зросла і кількість проблем, пов'язаних з безпекою, завдяки збільшенню об'ємів інформації, що передається уразливими лініями зв'язку. Величезну увагу почали приділяти шифруванню, у край необхідним стало кодування персональної і захищеної правами власності інформації.

Багатотермінальний режим використовувався не лише в системах розподілу часу, але і в системах пакетної обробки. При цьому не лише оператор, але і усі користувачі одержували можливість формувати свої завдання і управляти їх виконанням зі свого терміналу. Такі операційні системи одержали назву ***систем віддаленого введення завдань***.

Термінальні комплекси могли розташовуватися на великій відстані від процесорних стійок, з'єднуючись з ними за допомогою різних глобальних зв'язків – модемних з'єднань телефонних мереж або виділених каналів. Для підтримки віддаленої роботи терміналів в операційних системах з'явилися спеціальні програмні модулі, що реалізують різні протоколи зв'язку. Такі обчислювальні системи з віддаленими терміналами, зберігаючи централізований характер обробки даних, якоюсь мірою були прообразом сучасних мереж, а відповідне системне програмне забезпечення – прообразом ***мережевих операційних систем***.

Деякі слова хотілося б сказати про **системи реального часу**, що застосовуються для управління різними технічними об'єктами, такими, наприклад, як верстат, супутник і тому подібними. В усіх цих випадках існує гранично допустимий час, впродовж якого має бути виконана та або інша програма, що управляє об'єктом, інакше може статися аварія (наприклад, супутник вийде із зони видимості).

Таким чином, критерієм ефективності для систем реального часу є їх здатність витримувати заздалегідь задані інтервали часу між запуском програми і отриманням результату (керуючої дії). Цей час називається ***часом реакції системи***, а відповідна властивість системи – ***реактивністю***. Для цих систем мультипрограмна суміш є фіксованим набором заздалегідь розроблених програм, а вибір програми на виконання

здійснюється виходячи з поточного стану об'єкту або відповідно до розкладу планових робіт.