

CHƯƠNG 1. GIỚI THIỆU

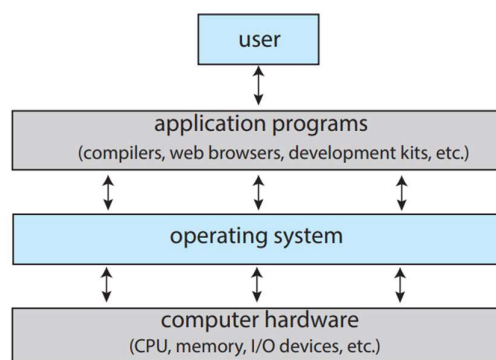
1.1. Khái niệm hệ điều hành

Vào những năm 1960, định nghĩa về hệ điều hành là phần mềm điều khiển phần cứng. Nhưng bối cảnh của hệ thống máy tính đã phát triển đáng kể kể từ đó, đòi hỏi một định nghĩa phong phú hơn.

Phần cứng ngày nay thực thi rất nhiều ứng dụng phần mềm. Để tăng cường sử dụng phần cứng, các ứng dụng được thiết kế để thực thi đồng thời. Nếu các ứng dụng này không được lập trình cẩn thận, chúng có thể gây nhiễu lẫn nhau. Từ đó, một lớp phần mềm được gọi là hệ điều hành sẽ tách các ứng dụng khỏi phần cứng và cung cấp các dịch vụ cho phép mỗi ứng dụng thực thi một cách an toàn và hiệu quả.

Hệ điều hành là phần mềm cho phép các ứng dụng tương tác với phần cứng của máy tính. Phần mềm chứa các thành phần lõi của hệ điều hành được gọi là nhân (kernel). Hệ điều hành có thể được tìm thấy trong các thiết bị khác nhau, từ điện thoại di động và ô tô đến máy tính cá nhân và máy tính lớn. Trong hầu hết các hệ thống máy tính, người dùng yêu cầu máy tính thực hiện một hành động (ví dụ: thực thi một ứng dụng hoặc in tài liệu) và hệ điều hành sẽ quản lý phần mềm và phần cứng để tạo ra kết quả mong muốn.

Hệ điều hành chủ yếu là trình quản lý tài nguyên - chúng quản lý phần cứng, bao gồm bộ xử lý, bộ nhớ, thiết bị đầu vào/đầu ra và thiết bị liên lạc. Nó cũng quản lý các ứng dụng và các phần mềm trừu tượng khác.



Hình 1-1. Các thành phần chính của hệ thống máy tính.

1.2. Lịch sử hệ điều hành

1.2.1. Giai đoạn 1940 – 1950

Các hệ điều hành đã phát triển trong 60 năm qua và trải qua nhiều giai đoạn hoặc thế hệ khác nhau tương ứng với nhiều thập kỷ. Vào những năm 1940, những chiếc máy tính kỹ thuật số điện tử đầu tiên không có hệ điều hành. Máy móc thời đó còn thô sơ đến mức các lập trình viên thường nhập các chương trình ngôn ngữ máy của họ từng bit một trên các hàng công tắc cơ học. Những năm sau các lập trình viên nhập chương trình ngôn ngữ máy của họ vào các thẻ đục lỗ. Sau đó, sử dụng hợp ngữ - sử dụng các chữ viết tắt giống như tiếng Anh để thể hiện các hoạt động cơ bản của máy tính - để tăng tốc quá trình lập trình.

Phòng thí nghiệm nghiên cứu General Motors đã phát triển hệ điều hành đầu tiên vào đầu những năm 1950 cho máy tính IBM 701 của mình. Các hệ điều hành những năm 1950 thường chỉ thực hiện một công việc tại một thời điểm. (Một công việc bao gồm các lệnh chương trình tương ứng với một nhiệm vụ tính toán cụ thể). Các công việc thường được thực hiện mà không cần người dùng nhập vào trong nhiều phút, nhiều giờ hoặc nhiều ngày. Những máy tính đầu tiên này được gọi là hệ thống xử lý hàng loạt đơn luồng, vì các chương trình và dữ liệu được gửi theo nhóm hoặc theo đợt bằng cách tải chúng liên tiếp vào băng hoặc đĩa. Bộ xử lý đọc các câu lệnh của công việc và thiết lập công việc tiếp theo. Khi công việc hiện tại kết thúc, trình đọc các câu lệnh công việc tiếp theo. Mặc dù hệ điều hành của những năm 1950 đã giảm thời gian chuyển đổi giữa các công việc nhưng các lập trình viên phải kiểm soát trực tiếp các tài nguyên hệ thống như bộ nhớ và thiết bị đầu vào/đầu ra. Đây là công việc chậm chạp, khó khăn và tẻ nhạt. Hơn nữa, những hệ thống ban đầu này yêu cầu toàn bộ chương trình phải được tải vào bộ nhớ để chương trình có thể chạy.

1.2.2. Giai đoạn những năm 1960

Các hệ thống của những năm 1960 cũng là hệ thống xử lý hàng loạt, nhưng chúng sử dụng tài nguyên máy tính hiệu quả hơn bằng cách chạy nhiều công việc cùng một lúc. Hệ thống bao gồm nhiều thiết bị ngoại vi như đầu đọc thẻ, máy đục lỗ thẻ, máy in, ổ băng từ và ổ đĩa. Bất kỳ một công việc nào cũng hiếm khi sử dụng được toàn bộ tài nguyên của hệ thống. Một công việc thông thường sẽ sử dụng bộ xử lý trong một khoảng thời gian nhất định trước khi thực hiện thao tác vào/ ra (I/O) trên một trong các thiết bị ngoại vi của hệ thống. Tại thời điểm này, bộ xử lý sẽ không hoạt động trong khi công việc chờ thao tác I/O hoàn tất. Các hệ thống của những năm 1960 đã cải thiện việc sử dụng tài nguyên bằng cách cho phép một công việc sử dụng bộ xử lý trong khi các công

việc khác sử dụng thiết bị ngoại vi. Trên thực tế, việc chạy hỗn hợp nhiều công việc khác nhau - một số công việc chủ yếu sử dụng bộ xử lý (được gọi là công việc liên quan đến bộ xử lý hoặc công việc liên quan đến tính toán) và một số công việc chủ yếu sử dụng các thiết bị ngoại vi (được gọi là công việc liên quan đến I/O) - có vẻ là cách tốt nhất tối ưu hóa việc sử dụng tài nguyên. Với những quan sát này, các nhà thiết kế hệ điều hành đã phát triển các hệ thống đa chương trình để quản lý nhiều công việc cùng một lúc. Trong môi trường đa chương trình, hệ điều hành nhanh chóng chuyển bộ xử lý từ công việc này sang công việc khác, duy trì hoạt động một số công việc trong khi vẫn duy trì sử dụng các thiết bị ngoại vi. Mức độ đa chương trình của hệ thống cho biết có thể quản lý bao nhiêu công việc cùng một lúc. Do đó, hệ điều hành đã phát triển từ việc quản lý một công việc sang quản lý nhiều công việc cùng một lúc.

Trong các hệ thống máy tính đa chương trình, chia sẻ tài nguyên là một trong những mục tiêu chính. Khi tài nguyên được chia sẻ giữa một tập hợp công việc, mỗi công việc duy trì quyền kiểm soát độc quyền đối với các tài nguyên cụ thể được phân bổ cho nó, một công việc có thể chờ một tài nguyên không bao giờ có sẵn. Nếu điều này xảy ra, công việc đó sẽ không hoàn thành nhiệm vụ, lúc này hệ thống yêu cầu người dùng khởi động lại nó, dẫn đến mất tất cả kết quả mà công việc đã hoàn thành trước đó.

Thông thường trong những năm 1960, người dùng không có mặt tại cơ sở máy tính khi công việc của họ được thực hiện. Các công việc được gửi trên thẻ đục lỗ hoặc băng máy tính và chờ cho đến khi người vận hành hệ thống có thể tải chúng vào máy tính để thực thi. Thông thường, công việc của người dùng sẽ mất hàng giờ hoặc thậm chí vài ngày trước khi được xử lý. Một lỗi nhỏ nhất trong chương trình, thậm chí là thiếu dấu chấm hoặc dấu phẩy, sẽ "đánh bom" công việc, lúc đó người dùng (thường nản lòng) sẽ sửa lỗi, gửi lại công việc và một lần nữa phải đợi hàng giờ hoặc hàng ngày để thực hiện lần thử tiếp theo. Việc phát triển phần mềm trong môi trường đó diễn ra rất chậm chạp.

Năm 1964, IBM công bố dòng máy tính System/360 ("360" dùng để chỉ tất cả các điểm trên la bàn để biểu thị khả năng ứng dụng phổ quát). Các mẫu máy tính 360 khác nhau được thiết kế phần cứng tương thích với hệ điều hành OS/360. Qua nhiều năm, IBM đã phát triển kiến trúc 360 của mình thành dòng 370 và gần hơn là dòng 390 và zSeries.

Các hệ điều hành tiên tiến hơn đã được phát triển để phục vụ nhiều người dùng tương tác cùng một lúc. Người dùng tương tác giao tiếp với công việc của họ trong quá trình thực hiện. Vào những năm 1960, người dùng tương tác với máy tính thông qua "thiết bị đầu cuối cầm" (tức là các thiết bị cung cấp giao diện người dùng nhưng không có nguồn điện bộ xử lý) trực tuyến (tức là được gắn trực tiếp vào máy tính thông qua

kết nối đang hoạt động). Vì người dùng hiện diện và tương tác với nó nên hệ thống máy tính cần phản hồi nhanh chóng các yêu cầu của người dùng; nếu không, năng suất của người dùng có thể bị ảnh hưởng. Tăng năng suất đã trở thành mục tiêu quan trọng của máy tính vì nguồn nhân lực cực kỳ đắt đỏ so với tài nguyên máy tính. Hệ thống chia sẻ thời gian được phát triển để hỗ trợ người dùng tương tác đồng thời.

Nhiều hệ thống chia sẻ thời gian của những năm 1960 là hệ thống đa chế độ hỗ trợ xử lý hàng loạt cũng như các ứng dụng thời gian thực (như hệ thống điều khiển quy trình công nghiệp). Các hệ thống thời gian thực cố gắng cung cấp phản hồi trong một khoảng thời gian giới hạn nhất định. Ví dụ, một phép đo từ một nhà máy lọc dầu cho thấy nhiệt độ quá cao có thể cần được chú ý ngay lập tức để ngăn chặn một vụ nổ. Các tài nguyên của hệ thống thời gian thực thường không được sử dụng đúng mức - điều quan trọng hơn là các hệ thống như vậy phải phản hồi nhanh hơn là sử dụng tài nguyên của chúng một cách hiệu quả. Phục vụ cả công việc hàng loạt và thời gian thực có nghĩa là hệ điều hành phải phân biệt giữa các loại người dùng và cung cấp cho mỗi loại mức độ dịch vụ phù hợp. Các công việc xử lý hàng loạt có thể gặp phải độ trễ hợp lý, trong khi các ứng dụng tương tác yêu cầu mức dịch vụ cao hơn và hệ thống thời gian thực yêu cầu mức dịch vụ cực cao.

Những hệ điều hành chia sẻ thời gian trong giai đoạn này bao gồm CTSS (Hệ thống chia sẻ thời gian tương thích) do MIT phát triển, TSS (Hệ thống chia sẻ thời gian) do IBM phát triển, hệ thống Multics được phát triển tại MIT, GE và Phòng thí nghiệm Bell là hệ thống kế thừa của CTSS và CP/CMS (Chương trình điều khiển/Hệ thống giám sát hội thoại) - mà cuối cùng đã phát triển thành hệ điều hành VM (Máy ảo) của IBM - được phát triển bởi Trung tâm khoa học Cambridge của IBM.

Sau khi được tải vào bộ nhớ chính, các chương trình có thể được thực thi nhanh chóng; tuy nhiên, bộ nhớ chính quá đắt để chứa số lượng lớn chương trình cùng một lúc. Trước những năm 1960, công việc phần lớn được tải vào bộ nhớ bằng thẻ đục lỗ hoặc băng, một công việc tẻ nhạt và tốn thời gian, trong thời gian đó hệ thống sẽ rồi, không làm gì. Các hệ điều hành của những năm 1960 đã kết hợp các thiết bị giúp giảm thời gian nhàn rỗi của hệ thống bằng cách lưu trữ lượng lớn dữ liệu trên các phương tiện lưu trữ rẻ tiền như băng từ, đĩa từ. Mặc dù đĩa cứng cho phép truy cập tương đối nhanh hơn vào các chương trình và dữ liệu so với băng, nhưng chúng chậm hơn đáng kể so với bộ nhớ chính. Vấn đề đặt ra trong giai đoạn này là tối ưu hóa hiệu suất ổ đĩa, làm sao hệ điều hành có thể quản lý các yêu cầu đầu vào/đầu ra của ổ đĩa để cải thiện hiệu suất.

1.2.3. Giai đoạn những năm 1970

Các hệ thống của những năm 1970 chủ yếu là các hệ thống đa chương trình đa chế độ hỗ trợ xử lý hàng loạt, chia sẻ thời gian và các ứng dụng thời gian thực. Máy tính cá nhân đang ở giai đoạn sơ khai, được phát triển liên tục nhờ công nghệ vi xử lý. Các hệ thống chia sẻ thời gian thử nghiệm của những năm 1960 đã phát triển thành các sản phẩm thương mại vững chắc vào những năm 1970. Truyền thông giữa các hệ thống máy tính trên khắp nước Mỹ tăng lên khi các tiêu chuẩn truyền thông TCP/IP của Bộ Quốc phòng được sử dụng rộng rãi - đặc biệt là trong môi trường điện toán quân sự và đại học. Truyền thông trong mạng cục bộ (LAN) được thực hiện một cách thiết thực và tiết kiệm nhờ tiêu chuẩn Ethernet được phát triển tại Trung tâm Nghiên cứu Palo Alto (PARC) của Xerox.

Các vấn đề về bảo mật gia tăng khi khối lượng thông tin truyền qua các đường dây liên lạc dễ bị tấn công ngày càng tăng. Mã hóa nhận được nhiều sự chú ý - việc mã hóa dữ liệu độc quyền hoặc riêng tư trở nên cần thiết để ngay cả khi dữ liệu bị xâm phạm, nó cũng không có giá trị đối với bất kỳ ai ngoài người nhận dự định. Trong những năm 1970, hệ điều hành đã phát triển bao gồm khả năng kết nối mạng và bảo mật, đồng thời tiếp tục cải thiện hiệu suất để đáp ứng nhu cầu thương mại.

Cuộc cách mạng máy tính cá nhân bắt đầu vào cuối những năm 1970 với những hệ thống như Apple II và bùng nổ vào những năm 1980.

1.2.4. Giai đoạn những năm 1980

Những năm 1980 là thập kỷ của máy tính cá nhân và máy trạm. Công nghệ vi xử lý đã phát triển đến mức có thể tạo được những máy tính để bàn cao cấp được gọi là máy trạm có sức mạnh tương đương với những máy tính lớn của một thập kỷ trước. Máy tính cá nhân IBM phát hành năm 1981 và máy tính cá nhân Apple Macintosh phát hành năm 1984 đã giúp các cá nhân và doanh nghiệp nhỏ có thể sở hữu máy tính chuyên dụng của riêng mình. Các phương tiện truyền thông có thể được sử dụng để truyền dữ liệu nhanh chóng và tiết kiệm giữa các hệ thống. Thay vì đưa dữ liệu đến một trung tâm lắp đặt máy tính quy mô lớn để xử lý, máy tính được phân phối đến những nơi cần đến dữ liệu. Các phần mềm như chương trình bảng tính, bộ xử lý văn bản, gói cơ sở dữ liệu và gói đồ họa đã giúp thúc đẩy cuộc cách mạng máy tính cá nhân bằng cách tạo ra nhu cầu từ các doanh nghiệp có thể sử dụng các sản phẩm này để tăng năng suất của họ.

Máy tính cá nhân tỏ ra tương đối dễ học và sử dụng, một phần là do giao diện đồ họa người dùng (GUI) sử dụng các ký hiệu đồ họa như cửa sổ, biểu tượng và menu để hỗ trợ người dùng tương tác với chương trình. Trung tâm Nghiên cứu Palo Alto (PARC) của Xerox đã phát triển chuột và GUI; Việc Apple phát hành máy tính cá nhân

Macintosh vào năm 1984 đã giúp phổ biến việc sử dụng chúng. Trong máy tính Macintosh, GUI được nhúng vào hệ điều hành để tất cả các ứng dụng đều có giao diện tương tự nhau. Khi đã quen với GUI Macintosh, người dùng có thể học cách sử dụng các ứng dụng mới nhanh hơn.

Khi chi phí công nghệ giảm, việc truyền thông tin giữa các máy tính trong mạng máy tính trở nên kinh tế và thiết thực hơn. Thư điện tử, truyền tập tin và các ứng dụng truy cập cơ sở dữ liệu từ xa ngày càng phát triển. Điện toán phân tán (tức là sử dụng nhiều máy tính độc lập để thực hiện một tác vụ chung) trở nên phổ biến theo mô hình máy khách/máy chủ. Khách hàng là máy tính người dùng yêu cầu các dịch vụ khác nhau; máy chủ là các máy tính thực hiện các dịch vụ được yêu cầu. Máy chủ thường được dành riêng cho một loại nhiệm vụ, chẳng hạn như hiển thị đồ họa, quản lý cơ sở dữ liệu hoặc phục vụ các trang Web.

Lĩnh vực công nghệ phần mềm tiếp tục phát triển, một lực đẩy lớn đến từ chính phủ Hoa Kỳ nhằm cung cấp sự kiểm soát chặt chẽ hơn các dự án phần mềm của Bộ Quốc phòng. Một số mục tiêu của sáng kiến này bao gồm khả năng sử dụng lại mã đã được hiện thực hóa và xây dựng sớm các nguyên mẫu để các nhà phát triển và người dùng có thể đề xuất sửa đổi sớm trong quá trình thiết kế phần mềm.

1.2.5. Giai đoạn những năm 1990

Hiệu suất phần cứng tiếp tục được cải thiện theo cấp số nhân trong những năm 1990. Đến cuối thập kỷ này, một máy tính cá nhân thông thường có thể thực thi hàng trăm triệu lệnh mỗi giây và lưu trữ hơn một gigabyte thông tin trên đĩa cứng; một số siêu máy tính có thể thực hiện hơn một nghìn tỷ phép tính mỗi giây. Bộ xử lý và bộ nhớ rẻ cho phép người dùng thực thi các chương trình lớn, phức tạp trên máy tính cá nhân và cho phép các công ty vừa và nhỏ sử dụng các máy này cho cơ sở dữ liệu mở rộng và các công việc xử lý từng được giao cho các hệ thống máy tính lớn. Chi phí công nghệ giảm cũng dẫn đến sự gia tăng số lượng máy tính gia đình, được sử dụng cho cả công việc và giải trí.

Vào những năm 1990, sự ra đời của World Wide Web đã dẫn đến sự bùng nổ của điện toán phân tán. Ban đầu, hệ điều hành thực hiện việc quản lý tài nguyên riêng biệt bên trong một máy tính. Với sự ra đời của World Wide Web và kết nối Internet ngày càng nhanh, điện toán phân tán đã trở nên phổ biến trong các máy tính cá nhân. Người dùng có thể yêu cầu dữ liệu được lưu trữ ở các vị trí ở xa hoặc yêu cầu các chương trình chạy trên các bộ xử lý ở xa. Các tổ chức lớn có thể sử dụng bộ đa xử lý phân tán (tức là mạng máy tính chứa nhiều bộ xử lý) để mở rộng quy mô tài nguyên và tăng hiệu quả.

Tuy nhiên, các ứng dụng phân tán vẫn còn hạn chế do giao tiếp qua mạng diễn ra ở tốc độ tương đối chậm so với tốc độ xử lý nội bộ của từng máy tính.

Khi nhu cầu kết nối Internet tăng lên, hệ điều hành hỗ trợ các tác vụ mạng đã trở thành tiêu chuẩn. Người dùng ở nhà và trong các tổ chức tăng năng suất bằng cách truy cập tài nguyên trên mạng máy tính. Tuy nhiên, kết nối ngày càng tăng dẫn đến sự gia tăng các mối đe dọa bảo mật máy tính. Các nhà thiết kế hệ điều hành đã phát triển các kỹ thuật để bảo vệ máy tính khỏi những cuộc tấn công độc hại này. Các mối đe dọa bảo mật ngày càng tinh vi hơn tiếp tục thách thức khả năng chống lại các cuộc tấn công như vậy của ngành công nghiệp máy tính.

Tập đoàn Microsoft trở nên thống trị vào những năm 1990. Năm 1981, Microsoft phát hành phiên bản đầu tiên của hệ điều hành DOS cho máy tính cá nhân IBM. Vào giữa những năm 1980, Microsoft đã phát triển hệ điều hành Windows, một giao diện đồ họa người dùng được xây dựng dựa trên hệ điều hành DOS. Microsoft phát hành Windows 3.0 vào năm 1990; phiên bản mới này có giao diện thân thiện với người dùng và chức năng phong phú. Hệ điều hành Windows trở nên vô cùng phổ biến sau khi Windows 3.1 được phát hành vào năm 1993, với những phiên bản kế nhiệm là Windows 95 và Windows 98, gần như đã chiếm lĩnh thị trường hệ điều hành dành cho máy tính để bàn vào cuối những năm 90. Các hệ điều hành này, vay mượn từ nhiều khái niệm (chẳng hạn như biểu tượng, menu và cửa sổ) được phổ biến bởi các hệ điều hành Macintosh đời đầu, cho phép người dùng điều hướng nhiều ứng dụng đồng thời một cách dễ dàng. Microsoft cũng tham gia vào thị trường hệ điều hành doanh nghiệp với bản phát hành Windows NT năm 1993, Windows NT nhanh chóng trở thành hệ điều hành được lựa chọn cho các máy trạm của công ty.

1.2.6. Giai đoạn những năm 2000 đến nay

Trong giai đoạn này, phần mềm trung gian (middleware), là phần mềm liên kết hai ứng dụng riêng biệt (thường qua mạng), đã trở nên quan trọng khi các ứng dụng được đẩy lên trên World Wide Web và khách hàng sử dụng chúng thông qua kết nối Internet tốc độ cao. Middleware phổ biến trong các ứng dụng Web, trong đó máy chủ Web server phải tạo ra nội dung để đáp ứng yêu cầu của người dùng với sự trợ giúp của cơ sở dữ liệu. Phần mềm trung gian hoạt động như một người chuyển phát nhanh để truyền các thông điệp giữa máy chủ Web và cơ sở dữ liệu, đơn giản hóa việc giao tiếp giữa nhiều kiến trúc khác nhau. Các dịch vụ web bao gồm một tập hợp các tiêu chuẩn liên quan có thể cho phép hai ứng dụng máy tính bất kỳ giao tiếp và trao đổi dữ liệu qua Internet. Một dịch vụ Web giao tiếp qua mạng để cung cấp một tập hợp các hoạt động cụ thể mà các ứng dụng khác có thể gọi ra. Dữ liệu được truyền qua lại bằng các giao thức chuẩn

như HTTP, cùng giao thức được sử dụng để truyền các trang Web thông thường. Các dịch vụ web hoạt động bằng cách sử dụng các tiêu chuẩn mở, dựa trên văn bản cho phép các thành phần được viết bằng các ngôn ngữ khác nhau và trên các nền tảng khác nhau có thể giao tiếp với nhau. Chúng là những phần mềm có sẵn để sử dụng trên Internet.

Các dịch vụ web sẽ giúp thúc đẩy sự chuyển dịch sang điện toán phân tán thực sự. Công cụ tìm kiếm Google cũng có thể được tích hợp với chức năng khác thông qua API Web của Google (www.google.com/apis), kết nối với chỉ mục của Google về các trang Web sử dụng dịch vụ Web.

Kiến trúc mạng và đa bộ xử lý đang tạo ra nhiều cơ hội cho nghiên cứu và phát triển các kỹ thuật thiết kế phần cứng và phần mềm mới. Các ngôn ngữ lập trình tuần tự chỉ định một phép tính tại một thời điểm hiện được bổ sung bằng các ngôn ngữ lập trình đồng thời, chẳng hạn như Java, cho phép tính toán song song; trong Java, các đơn vị tính toán song song được chỉ định thông qua các luồng.

Ngày càng có nhiều hệ thống thể hiện tính song song lớn; chúng có số lượng lớn bộ xử lý để có thể thực hiện song song nhiều phần tính toán độc lập. Về mặt khái niệm, điều này khác biệt đáng kể so với tính toán tuần tự trong 60 năm qua; có những vấn đề quan trọng và đầy thách thức trong việc phát triển phần mềm phù hợp để giải quyết vấn đề song song như vậy.

Các hệ điều hành đang chuẩn hóa giao diện người dùng và ứng dụng để chúng dễ sử dụng hơn và hỗ trợ nhiều chương trình hơn. Microsoft đã hợp nhất các dòng hệ điều hành Windows dành cho người tiêu dùng và chuyên nghiệp vào Windows XP. Trong hệ điều hành tiếp theo của nó (tên mã Longhorn).

Microsoft có kế hoạch tích hợp các định dạng của các loại tệp khác nhau. Ví dụ: điều này sẽ cho phép người dùng tìm kiếm hệ thống của họ cho tất cả các tệp (tài liệu, bảng tính, e-mail, v.v.) có chứa các từ khóa nhất định. Longhorn cũng sẽ bao gồm giao diện người dùng 3D nâng cao, bảo mật được cải thiện và hỗ trợ cho việc ghi DVD. Các hệ điều hành nguồn mở, chẳng hạn như Linux, sẽ được sử dụng rộng rãi hơn và sẽ sử dụng các giao diện lập trình ứng dụng (API) tiêu chuẩn như Giao diện hệ điều hành di động (POSIX) để cải thiện khả năng tương thích với các hệ điều hành dựa trên UNIX khác. chẳng hạn như điện thoại di động và PDA, sẽ trở nên phổ biến hơn khi các thiết bị di động được trang bị bộ vi xử lý ngày càng mạnh mẽ.

Ngày nay, những thiết bị này được sử dụng cho các chức năng như e-mail, duyệt Web và chụp ảnh kỹ thuật số. Các ứng dụng sử dụng nhiều tài nguyên, chẳng hạn như video chuyển động đầy đủ, sẽ phát triển nhanh chóng trên các thiết bị này. Do tài nguyên

thiết bị di động bị giới hạn bởi kích thước nhỏ của thiết bị nên điện toán phân tán sẽ đóng vai trò thậm chí còn lớn hơn vì PDA và điện thoại di động sẽ yêu cầu lượng dữ liệu và sức mạnh xử lý ngày càng tăng từ các máy tính từ xa.

1.3. Các thành phần của hệ điều hành

Người dùng tương tác với hệ điều hành thông qua một hoặc nhiều ứng dụng đặc biệt gọi là shell hoặc trình thông dịch lệnh. Hầu hết các shell ngày nay được cài đặt dưới dạng giao diện console cho phép người dùng ra lệnh từ bàn phím hoặc dưới dạng GUI cho phép người dùng trỏ và nhấp, kéo và thả biểu tượng để yêu cầu dịch vụ từ hệ điều hành (ví dụ: để mở một ứng dụng). Ví dụ: Microsoft Windows cung cấp GUI qua đó người dùng có thể ra lệnh; Ngoài ra, người dùng có thể mở cửa sổ nhắc lệnh chấp nhận các lệnh đã nhập.

Phần mềm chứa các thành phần lõi của hệ điều hành được gọi là nhân (kernel). Các thành phần nhân của hệ điều hành điển hình bao gồm:

- Bộ lập lịch tiến trình, xác định thời điểm và thời gian thực hiện một tiến trình trên bộ xử lý.
- Trình quản lý bộ nhớ, xác định thời điểm và cách thức phân bổ bộ nhớ cho các tiến trình và phải làm gì khi bộ nhớ chính đầy.
- Trình quản lý I/O, phục vụ các yêu cầu vào/ra tương ứng từ và tới các thiết bị phần cứng.
- Trình quản lý giao tiếp liên tiến trình (IPC), cho phép các tiến trình giao tiếp với nhau.
- Trình quản lý hệ thống tệp, tổ chức các bộ sưu tập dữ liệu được đặt tên trên các thiết bị lưu trữ và cung cấp giao diện để truy cập dữ liệu trên các thiết bị đó.

1.4. Mục tiêu của hệ điều hành

Người dùng mong muốn những đặc điểm nhất định của hệ điều hành, chẳng hạn như:

- Hiệu quả (efficiency)
- Mạnh mẽ (robustness)
- Khả năng mở rộng tài nguyên (scalability)
- Khả năng mở rộng công nghệ (extensibility)
- Tính di động (portability)

- An toàn (security)
- Tính tương tác (interactivity)
- Khả năng sử dụng (usability)

Một hệ điều hành hiệu quả phải đạt được thông lượng cao và thời gian quay vòng trung bình thấp. Thông lượng đo lường công việc mà bộ xử lý có thể hoàn thành trong một khoảng thời gian nhất định. Hãy nhớ lại rằng một vai trò của hệ điều hành là cung cấp dịch vụ cho nhiều ứng dụng. Một hệ điều hành hiệu quả sẽ giảm thiểu thời gian cung cấp các dịch vụ này.

Một hệ điều hành mạnh mẽ phải có khả năng chịu lỗi và đáng tin cậy - hệ thống sẽ không bị lỗi do lỗi phần cứng hoặc ứng dụng riêng biệt và nếu bị lỗi, nó sẽ hoạt động một cách phù hợp (tức là bằng cách giảm thiểu mất mát công việc và bằng cách ngăn ngừa hư hỏng phần cứng của hệ thống). Một hệ điều hành như vậy sẽ cung cấp dịch vụ cho từng ứng dụng trên phần cứng đó.

Một hệ điều hành có khả năng mở rộng tài nguyên có thể sử dụng các tài nguyên khi chúng được thêm vào. Nếu một hệ điều hành không có khả năng mở rộng thì nó sẽ nhanh chóng đạt đến điểm mà các tài nguyên bổ sung sẽ không được sử dụng hết. Một hệ điều hành có khả năng mở rộng tài nguyên có thể dễ dàng điều chỉnh mức độ đa chương trình của nó. Khả năng mở rộng là một thuộc tính đặc biệt quan trọng của hệ thống đa bộ xử lý - khi có nhiều bộ xử lý được thêm vào hệ thống, lý tưởng nhất là khả năng xử lý sẽ tăng tỷ lệ thuận với số lượng tiến trình, tuy nhiên, trên thực tế, điều đó không xảy ra.

Một hệ điều hành có thể mở rộng công nghệ sẽ thích ứng tốt với các công nghệ mới và cung cấp khả năng mở rộng hệ điều hành để thực hiện các nhiệm vụ ngoài thiết kế ban đầu của nó.

Một hệ điều hành di động được thiết kế sao cho có thể hoạt động trên nhiều cấu hình phần cứng. Tính di động của ứng dụng cũng rất quan trọng vì việc phát triển ứng dụng rất tốn kém, do đó, cùng một ứng dụng phải chạy trên nhiều cấu hình phần cứng khác nhau để giảm chi phí phát triển. Hệ điều hành đóng vai trò quan trọng để đạt được tính di động này.

Hệ điều hành an toàn ngăn người dùng và phần mềm truy cập các dịch vụ và tài nguyên mà không được phép.

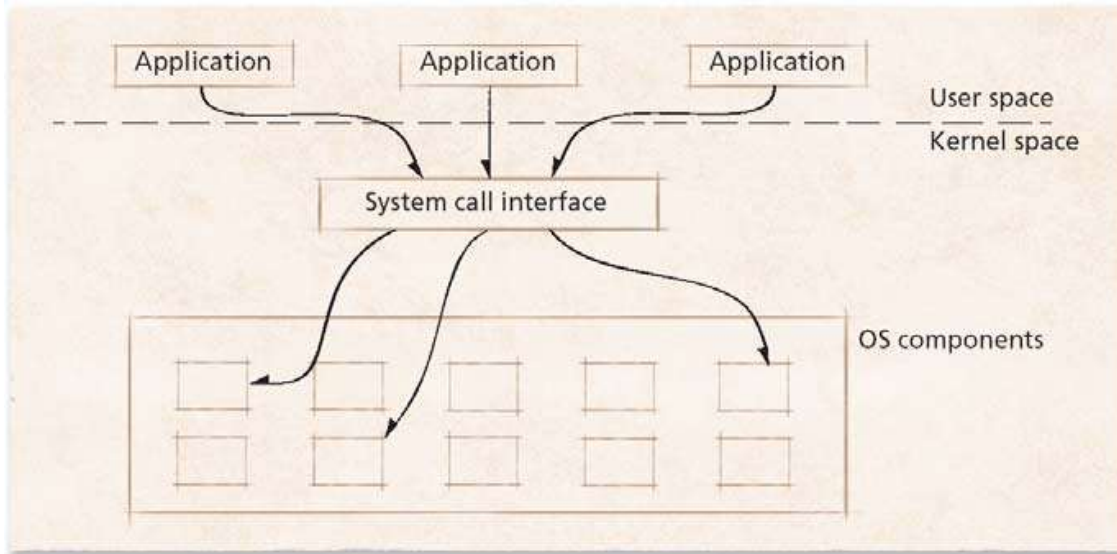
Hệ điều hành tương tác cho phép các ứng dụng phản hồi nhanh chóng với các hành động hoặc sự kiện của người dùng. Một hệ điều hành có thể sử dụng được là một hệ

điều hành có tiềm năng phục vụ một lượng lớn người dùng. Các hệ điều hành này thường cung cấp giao diện người dùng dễ sử dụng. Các hệ điều hành như Linux, Windows XP và MacOS X được đặc trưng là các hệ điều hành có thể sử dụng được vì mỗi hệ điều hành đều hỗ trợ một bộ ứng dụng lớn và cung cấp giao diện người dùng tiêu chuẩn. Nhiều hệ điều hành thử nghiệm và học thuật không hỗ trợ một số lượng lớn ứng dụng hoặc cung cấp giao diện thân thiện với người dùng và do đó không được coi là có thể sử dụng được.

1.5. Cơ bản về ứng dụng

Khi Máy tính cá nhân IBM (thường được gọi đơn giản là "PC") xuất hiện vào năm 1981, nó ngay lập tức tạo ra một ngành công nghiệp phần mềm khổng lồ trong đó các nhà cung cấp phần mềm độc lập có thể tiếp thị các gói dành cho PC IBM chạy theo hệ điều hành MS-DOS. Hệ điều hành giúp các nhà phát triển phần mềm ứng dụng không phải đối mặt với những chi tiết lộn xộn khi thao tác phần cứng máy tính để quản lý bộ nhớ, thực hiện đầu vào/đầu ra, xử lý các đường truyền thông, v.v. Hệ điều hành cung cấp một loạt các lệnh gọi giao diện lập trình ứng dụng (API) mà các lập trình viên ứng dụng sử dụng để thực hiện các thao tác phần cứng và các hoạt động khác. API cung cấp các lệnh gọi hệ thống mà qua đó chương trình người dùng sẽ hướng dẫn hệ điều hành thực hiện công việc; nhà phát triển ứng dụng chỉ cần biết nên gọi những thủ tục nào để hoàn thành các nhiệm vụ cụ thể (Hình 1-2). Trong Hình 1-2, vùng phía trên đường nét đứt là không gian người dùng, biểu thị các thành phần phần mềm không phải của hệ điều hành và không thể truy cập trực tiếp vào tài nguyên vật lý của hệ thống. Vùng bên dưới đường nét đứt là không gian nhân, biểu thị các thành phần của hệ điều hành và có quyền truy cập không hạn chế vào tài nguyên hệ thống. Nếu một ứng dụng cố gắng sử dụng sai tài nguyên hệ thống hoặc sử dụng các tài nguyên chưa được cấp thì hệ điều hành phải can thiệp để ngăn ứng dụng đó làm hỏng hệ thống hoặc can thiệp vào các ứng dụng khác của người dùng.

Môi trường phát triển ứng dụng do MS-DOS tạo ra đã khuyến khích sự phát triển của hàng chục nghìn gói phần mềm ứng dụng. Điều này lại khuyến khích người dùng mua PC IBM và các thiết bị tương thích. Windows có thể có cơ sở ứng dụng gồm hàng trăm nghìn ứng dụng.



Hình 1-2. Tương tác giữa ứng dụng và hệ điều hành.

Khi cơ sở ứng dụng (tức là sự kết hợp giữa phần cứng và môi trường hệ điều hành trong đó ứng dụng được phát triển) được thiết lập rộng rãi, việc yêu cầu người dùng và nhà phát triển phần mềm chuyển đổi sang môi trường phát triển ứng dụng hoàn toàn mới được cung cấp bởi một nền tảng phát triển ứng dụng hoàn toàn mới sẽ trở nên vô cùng khó khăn. Vì vậy, rất có thể các kiến trúc mới phát triển trong vài năm tới sẽ nỗ lực hết sức để hỗ trợ các cơ sở ứng dụng chính hiện có.

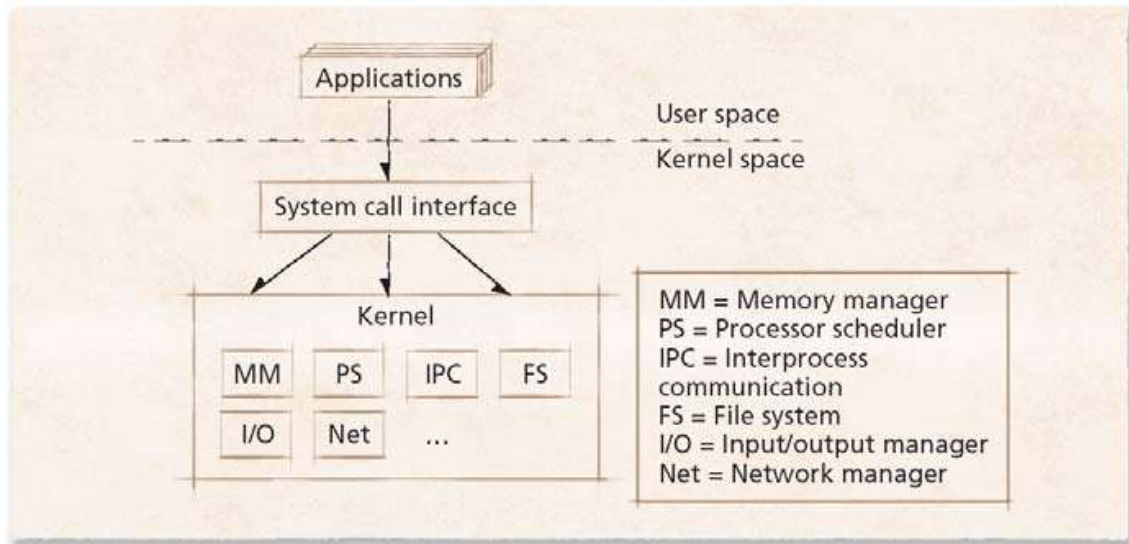
1.6. Kiến trúc hệ điều hành

Các hệ điều hành ngày nay có xu hướng phức tạp vì chúng cung cấp nhiều dịch vụ và hỗ trợ nhiều loại tài nguyên phần cứng và phần mềm. Kiến trúc hệ điều hành có thể giúp các nhà thiết kế quản lý sự phức tạp này bằng cách tổ chức các thành phần hệ điều hành và chỉ định đặc quyền mà mỗi thành phần thực thi. Trong thiết kế nguyên khối, mọi thành phần của hệ điều hành đều được chứa trong kernel; trong thiết kế vi nhân, chỉ bao gồm các thành phần thiết yếu.

1.6.1. Kiến trúc nguyên khối

Hệ điều hành nguyên khối là kiến trúc hệ điều hành sớm nhất và phổ biến nhất. Mọi thành phần của hệ điều hành đều được chứa trong kernel và có thể giao tiếp trực tiếp với bất kỳ thành phần nào khác (tức là chỉ bằng cách sử dụng lệnh gọi hàm). Nhân thường thực thi với quyền truy cập không hạn chế vào hệ thống máy tính (Hình 1-3). OS/360, VMS và Linux là các hệ điều hành nguyên khối. Giao tiếp trực tiếp giữa các thành phần làm cho hệ điều hành nguyên khối đạt hiệu quả cao. Bởi vì các nhân nguyên khối nhóm các thành phần lại với nhau, tuy nhiên rất khó để tách biệt nguồn lỗi. Hơn

nữa, vì tất cả các mã đều thực thi với quyền truy cập không hạn chế vào hệ thống, nên các hệ thống có nhân nguyên khối đặc biệt dễ bị hư hại do mã sai hoặc mã độc.



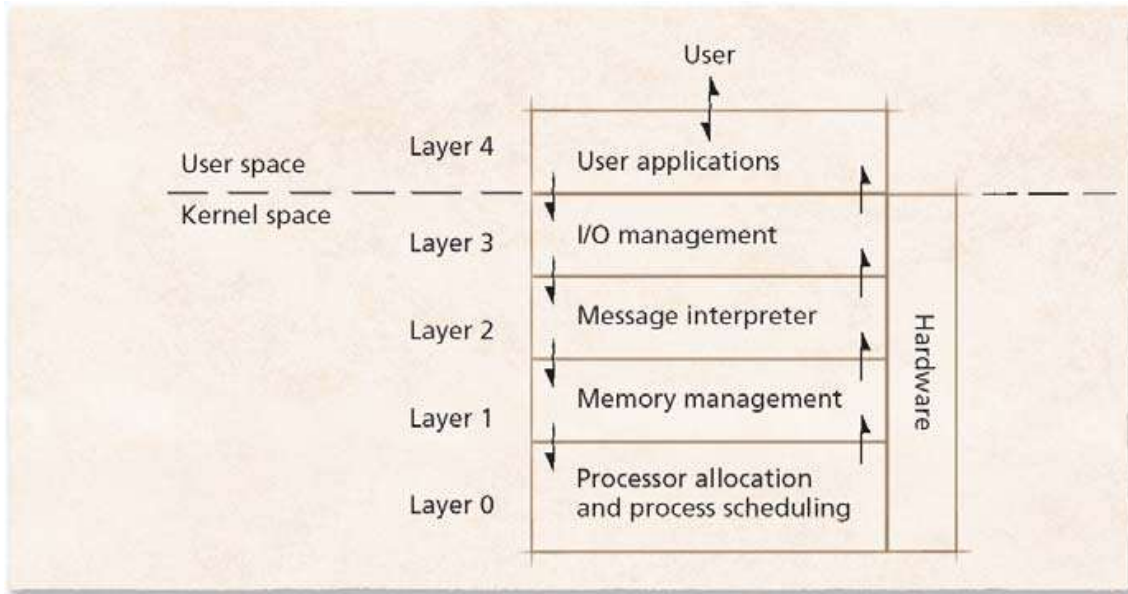
Hình 1-3. Kiến trúc nhân hệ điều hành nguyên khối.

1.6.2. Kiến trúc phân lớp

Khi các hệ điều hành trở nên lớn hơn và phức tạp hơn, các thiết kế nguyên khối thuần túy trở nên khó sử dụng. Cách tiếp cận theo lớp đối với hệ điều hành cố gắng giải quyết vấn đề này bằng cách nhóm các thành phần thực hiện các chức năng tương tự thành các lớp. Mỗi lớp giao tiếp độc quyền với những lớp ngay bên trên và bên dưới nó. Các lớp cấp thấp hơn cung cấp dịch vụ cho các lớp cấp cao hơn bằng cách sử dụng giao diện ẩn việc triển khai chúng.

Hệ điều hành phân lớp có tính mô-đun cao hơn hệ điều hành nguyên khối vì việc triển khai từng lớp có thể được sửa đổi mà không yêu cầu bất kỳ sửa đổi nào đối với các lớp khác. Một hệ thống mô-đun có thành phần độc lập có thể được tái sử dụng trong toàn bộ hệ thống. Mỗi thành phần ẩn cách nó thực hiện công việc của mình và trình bày một giao diện chuẩn mà các thành phần khác có thể sử dụng để yêu cầu các dịch vụ của nó. Tính mô-đun áp đặt cấu trúc và tính nhất quán trên hệ điều hành—thường đơn giản hóa việc xác thực, gỡ lỗi và sửa đổi. Tuy nhiên, theo cách tiếp cận theo lớp, yêu cầu của quy trình người dùng có thể cần phải đi qua nhiều lớp trước khi được phục vụ. Bởi vì các phương thức bổ sung phải được gọi để truyền dữ liệu từ lớp này sang lớp tiếp theo nên hiệu suất sẽ giảm so với hạt nhân nguyên khối, có thể chỉ yêu cầu một lệnh gọi duy nhất để phục vụ một yêu cầu tương tự. Ngoài ra, vì tất cả các lớp đều có quyền truy cập không hạn chế vào hệ thống, các hạt nhân phân lớp cũng dễ bị hư hại do lỗi hoặc mã độc. Hệ điều hành THE là một ví dụ ban đầu về hệ điều hành phân lớp (Hình 1-4). Nhiều

hệ điều hành ngày nay, bao gồm Windows và Linux, cài đặt một số cấp độ phân lớp.

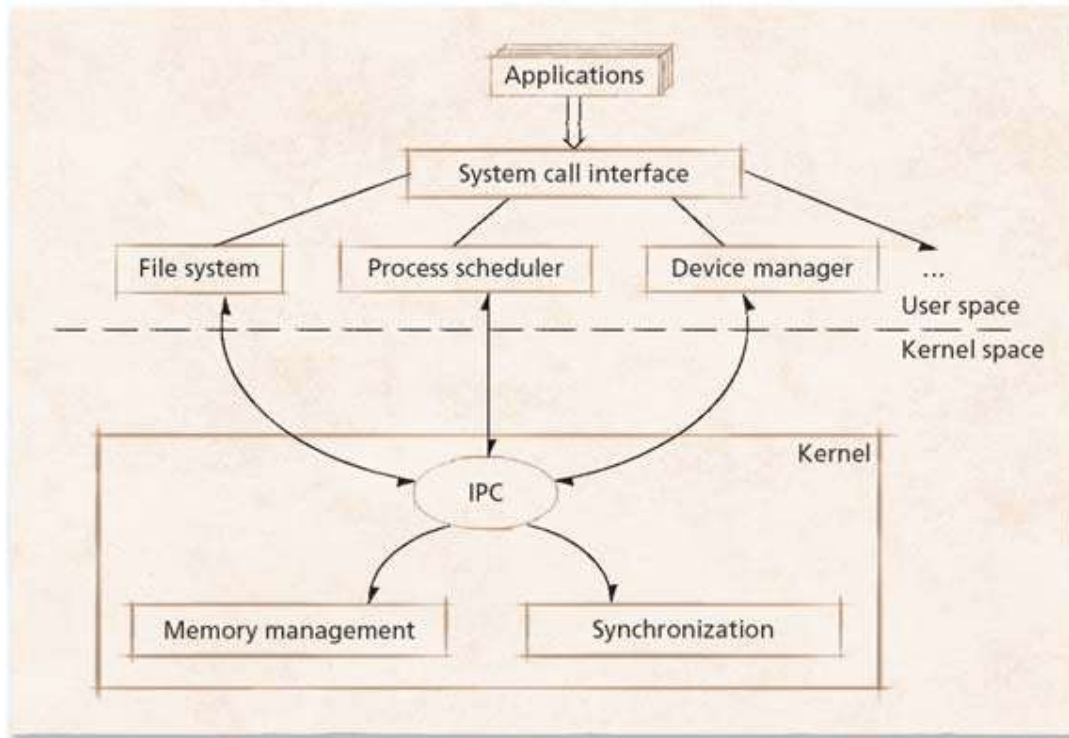


Hình 1-4. Các lớp của hệ điều hành THE.

1.6.3. Kiến trúc vi nhân

Kiến trúc hệ điều hành vi nhân chỉ cung cấp một số lượng nhỏ các dịch vụ nhằm cố gắng giữ cho hạt nhân nhỏ và có khả năng mở rộng. Các dịch vụ này thường bao gồm quản lý bộ nhớ cấp thấp, liên lạc giữa các tiến trình và đồng bộ hóa tiến trình cơ bản để cho phép các tiến trình hợp tác. Trong các thiết kế vi nhân, hầu hết các thành phần của hệ điều hành - chẳng hạn như quản lý tiến trình, kết nối mạng, tương tác hệ thống tệp và quản lý thiết bị - thực thi bên ngoài nhân với mức đặc quyền thấp hơn (Hình 1-5). Vi nhân thể hiện mức độ mô-đun hóa cao, khiến chúng có khả năng mở rộng tài nguyên, di động và có thể mở rộng công nghệ. Hơn nữa, vì vi nhân không dựa vào từng thành phần để thực thi nên một hoặc nhiều thành phần có thể bị lỗi mà không khiến hệ điều hành bị lỗi. Tuy nhiên, tính mô-đun như vậy phải trả giá bằng việc tăng mức độ giao tiếp giữa các mô-đun, điều này có thể làm giảm hiệu suất hệ thống.

Mặc dù rất ít hệ điều hành phổ biến ngày nay hoàn toàn áp dụng thiết kế vi nhân, nhưng Linux và Windows XP chẳng hạn, có chứa các thành phần mô-đun.



Hình 1-5. Kiến trúc hệ điều hành vi nhân.

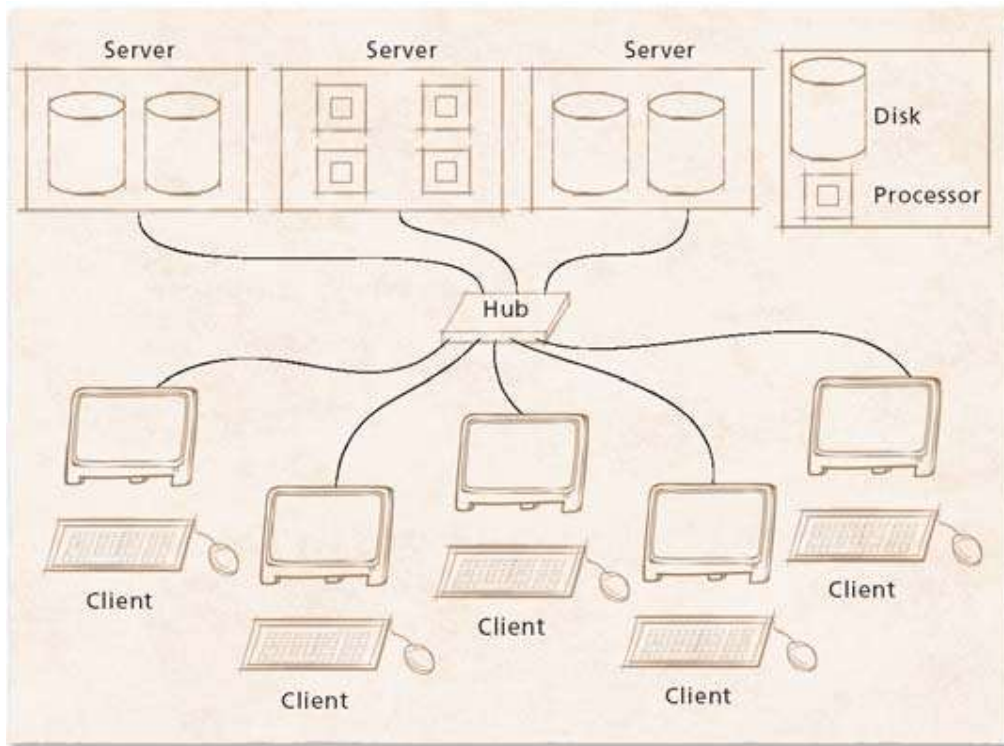
1.6.4. Hệ điều hành mạng và phân tán

Những tiến bộ trong công nghệ viễn thông đã ảnh hưởng sâu sắc đến hệ điều hành. Hệ điều hành mạng cho phép các tiến trình của nó truy cập vào các tài nguyên (ví dụ: các tệp) nằm trên các máy tính độc lập khác trên mạng. Cấu trúc của nhiều hệ điều hành mạng và phân tán thường dựa trên mô hình client/server chủ (Hình 1-6). Client trong mạng yêu cầu tài nguyên - chẳng hạn như tệp và thời gian bộ xử lý - thông qua giao thức mạng thích hợp. Các server đáp ứng các tài nguyên thích hợp. Trong các mạng như vậy, người thiết kế hệ điều hành phải xem xét cẩn thận cách quản lý dữ liệu và liên lạc giữa các máy tính.

Một số hệ điều hành có tính "kết nối mạng" nhiều hơn những hệ điều hành khác. Trong môi trường nối mạng, một tiến trình có thể được thực thi trên máy tính mà nó được tạo ra hoặc trên một máy tính khác trên mạng. Trong một số hệ điều hành mạng, người dùng có thể chỉ định chính xác nơi các tiến trình của họ chạy; nói cách khác, hệ điều hành xác định nơi các tiến trình được thực thi. Ví dụ: hệ thống có thể xác định rằng một tiến trình có thể được thực thi hiệu quả hơn trên máy tính đang tải nhẹ. Hệ thống tệp được nối mạng là một thành phần quan trọng của hệ điều hành mạng. Ở mức thấp nhất, người dùng lấy tài nguyên trên máy khác bằng cách kết nối rõ ràng với máy đó và truy xuất tệp. Hệ thống tệp mạng cấp cao hơn cho phép người dùng truy cập các tệp từ

xa như thể chúng ở trên hệ thống cục bộ. Ví dụ về hệ thống tệp mạng trong Hệ thống tệp mạng (NFS) của Sun và hệ thống tệp Andrew và Coda của CMU.

Hệ điều hành phân tán là một hệ điều hành duy nhất quản lý tài nguyên trên nhiều hệ thống máy tính. Các hệ thống phân tán tạo ra cảm giác các máy tính như là một máy tính duy nhất, do đó một tiến trình có thể truy cập tất cả các tài nguyên của hệ thống bất kể vị trí của tiến trình đó trong mạng máy tính của hệ thống phân tán. Các hệ điều hành phân tán thường khó cài đặt và yêu cầu các thuật toán phức tạp để cho phép các tiến trình giao tiếp và chia sẻ dữ liệu. Ví dụ về hệ điều hành phân tán là hệ điều hành Chord của MIT và hệ điều hành Amoeba của Đại học Vrije (VU) ở Amsterdam.



Hình 1-6. Mô hình hệ điều hành mạng.

1.7. Câu hỏi ôn tập

Câu 1.1. Hệ điều hành chỉ quản lý phần cứng?

SAI. Hệ điều hành quản lý các ứng dụng và phần mềm trừu tượng khác, chẳng hạn như máy ảo.

Câu 1.2. Mục đích chính của hệ điều hành là gì?

Mục đích chính của hệ điều hành là cho phép các ứng dụng tương tác với phần cứng của máy tính và quản lý tài nguyên phần cứng và phần mềm của hệ thống.

Câu 1.3. Tại sao ngôn ngữ assembly được phát triển?

Hợp ngữ được phát triển để tăng tốc quá trình lập trình. Chúng cho phép các lập trình viên sử dụng các lệnh dưới dạng viết tắt giống như tiếng Anh để dễ dàng làm việc hơn so với các lệnh bằng ngôn ngữ máy.

Câu 1.4. Điều gì đã hạn chế quy mô và khả năng của các chương trình trong những năm 1950?

Toàn bộ chương trình phải được tải vào bộ nhớ để thực thi. Bởi vì bộ nhớ tương đối đắt tiền nên dung lượng bộ nhớ khả dụng cho những máy tính đó rất nhỏ.

Câu 1.5. Tính toán tương tác và sự cải thiện về thời gian xử lý của nó đã tăng năng suất của lập trình viên như thế nào?

Khoảng thời gian từ khi hoàn thành công việc đến khi trả kết quả đã giảm từ hàng giờ hoặc hàng ngày xuống còn vài phút hoặc thậm chí vài giây. Điều này cho phép các lập trình viên nhập, biên dịch và chỉnh sửa chương trình một cách tương tác cho đến khi lỗi cú pháp được loại bỏ.

Câu 1.6. TSS, Multics và CP/CMS đã kết hợp khái niệm mới nào? Tại sao nó lại hữu ích cho các lập trình viên?

TSS, Multics và CP/CMS đều được tích hợp bộ nhớ ảo. Bộ nhớ ảo cho phép các ứng dụng truy cập vào nhiều bộ nhớ hơn mức có sẵn trên hệ thống. Điều này cho phép các lập trình viên phát triển các ứng dụng lớn hơn, mạnh mẽ hơn. Ngoài ra, hệ thống bộ nhớ ảo loại bỏ phần lớn gánh nặng quản lý bộ nhớ của người lập trình.

Câu 1.7. Những phát triển nào trong những năm 1970 đã cải thiện khả năng giao tiếp giữa các hệ thống máy tính?

Các tiêu chuẩn TCP/IP của DoD được sử dụng rộng rãi trong truyền thông mạng - chủ yếu trong môi trường máy tính của trường đại học và quân đội. Ngoài ra, PARC của Xerox đã phát triển tiêu chuẩn Ethernet, làm cho mạng cục bộ (LAN) tốc độ tương đối cao trở nên thiết thực và tiết kiệm.

Câu 1.8. Vấn đề mới nào đã được đưa ra do sự gia tăng giao tiếp giữa các máy tính? Vấn đề này đã được giải quyết như thế nào?

Giao tiếp giữa các máy tính gây ra các vấn đề về bảo mật vì dữ liệu được gửi qua các đường truyền dễ bị tổn thương. Mã hóa được sử dụng để làm cho dữ liệu không thể đọc được đối với bất kỳ ai khác ngoài người nhận.

Câu 1.9. Khía cạnh nào của máy tính cá nhân, được Apple Macintosh phổ biến, khiến

chúng đặc biệt dễ học và sử dụng?

Giao diện người dùng đồ họa (GUI) tạo điều kiện thuận lợi cho việc sử dụng máy tính cá nhân bằng cách cung cấp giao diện thống nhất, dễ sử dụng cho mọi ứng dụng. Điều này cho phép người dùng tìm hiểu các ứng dụng mới nhanh hơn.

Câu 1.10. Những công nghệ nào có thể được sử dụng để thu hẹp khoảng cách giữa các hệ điều hành khác nhau? Làm thế nào những công nghệ này có thể thực thi cùng một ứng dụng trên nhiều nền tảng?

Máy ảo và trình giả lập hệ điều hành thu hẹp khoảng cách giữa các hệ điều hành khác nhau. Các ứng dụng có thể được viết một lần để sử dụng chức năng của máy ảo hoặc trình mô phỏng. Các máy ảo hoặc trình mô phỏng có thể được triển khai để ẩn phần trình bày của nền tảng cơ bản khỏi các ứng dụng.

Câu 1.11. Những thành phần nào của phần hệ điều hành thực hiện từng hoạt động sau đây?

- a. Ghi vào đĩa.
- b. Xác định quá trình nào sẽ chạy tiếp theo.
- c. Xác định vị trí trong bộ nhớ mà một tiến trình mới sẽ được đặt.
- d. Tổ chức các tập tin trên đĩa.
- e. Cho phép một tiến trình gửi dữ liệu đến một tiến trình khác.

a) I/O manager; b) processor scheduler; c) memory manager; d) file system manager; e) interprocess communication (IPC) manager.

Câu 1.12. Tại sao việc cho phép người dùng thực hiện các thao tác đọc hoặc ghi vào bất kỳ vùng đĩa nào theo ý muốn lại nguy hiểm?

Điều này nguy hiểm vì người dùng có thể vô tình hoặc cố ý ghi đè lên dữ liệu quan trọng (chẳng hạn như tệp hệ điều hành) hoặc đọc thông tin nhạy cảm (chẳng hạn như tài liệu bí mật) mà không được phép.

Câu 1.13. Mục tiêu nào của hệ điều hành tương ứng với từng đặc điểm sau?

- a. Người dùng không thể truy cập dịch vụ hoặc thông tin mà không có sự cho phép thích hợp.
- b. Hệ điều hành chạy trên nhiều cấu hình phần cứng khác nhau.
- c. Hiệu suất hệ thống tăng đều đặn khi bổ sung thêm bộ nhớ và bộ xử lý.
- d. Hệ điều hành hỗ trợ các thiết bị không có sẵn tại thời điểm thiết kế.

e. Lỗi phần cứng không nhất thiết khiến hệ thống bị lỗi

a) security; b) portability; c) scalability; d) extensibility; e) robustness.

Câu 1.14. Hỗ trợ trình điều khiển thiết bị đóng góp như thế nào vào khả năng mở rộng của hệ điều hành?

Trình điều khiển thiết bị cho phép các nhà phát triển thêm hỗ trợ cho phần cứng không tồn tại khi hệ điều hành được thiết kế. Với mỗi loại thiết bị mới được thêm vào hệ thống, trình điều khiển thiết bị tương ứng phải được cài đặt.

Câu 1.15. Đặc điểm xác định của một hệ điều hành nguyên khối là gì?

Trong hệ điều hành nguyên khối, mọi thành phần của hệ điều hành đều được chứa trong kernel.

Câu 1.16. Tại sao hệ điều hành nguyên khối có xu hướng hoạt động hiệu quả? Điểm yếu chính của hạt nhân nguyên khối là gì?

Nhân nguyên khối có xu hướng hoạt động hiệu quả vì có ít cuộc gọi chuyển từ không gian người dùng sang không gian nhân. Vì tất cả mã hệ điều hành trong nhân nguyên khối đều hoạt động với quyền truy cập không hạn chế vào phần cứng và phần mềm của máy tính nên các hệ thống này đặc biệt dễ bị hư hỏng do mã sai.

Câu 1.17. Hệ điều hành phân lớp có tính mô đun hóa cao hơn hệ điều hành nguyên khối như thế nào?

Trong các hệ điều hành phân lớp, việc cài đặt và giao diện là riêng cho mỗi lớp. Điều này cho phép mỗi lớp được kiểm tra và gỡ lỗi riêng biệt. Nó cũng cho phép các nhà thiết kế thay đổi cách cài đặt từng lớp mà không cần sửa đổi các lớp khác.

Câu 1.18. Tại sao hệ điều hành phân lớp có xu hướng kém hiệu quả hơn hệ điều hành nguyên khối?

Trong các hệ điều hành phân lớp, một số cuộc gọi có thể được yêu cầu để liên lạc giữa các lớp, trong khi chi phí này không tồn tại trong các hạt nhân nguyên khối.

Câu 1.19. Sự khác biệt giữa kiến trúc phân lớp thuần túy và kiến trúc vi nhân là gì?

Kiến trúc phân lớp cho phép giao tiếp độc quyền giữa các thành phần hệ điều hành trong các lớp liên kề. Kiến trúc vi nhân cho phép giao tiếp giữa tất cả các thành phần của hệ điều hành thông qua vi nhân.

Câu 1.20. Làm thế nào để vi nhân thúc đẩy tính di động?

Vi nhân không phụ thuộc vào một nền tảng phần cứng cụ thể; hỗ trợ cho phần cứng

mới có thể được cung cấp bằng cách tải một mô-đun mới.

Câu 1.21. Sự khác biệt chính giữa hệ điều hành mạng và phân tán là gì?

Một hệ điều hành mạng điều khiển một máy tính nhưng hợp tác với các máy tính khác trên mạng. Trong hệ điều hành phân tán, một hệ điều hành điều khiển nhiều máy tính trong mạng.

Câu 1.22. Ưu điểm chính của hệ điều hành phân tán là gì? Thách thức chính của việc thiết kế một cái là gì?

Ưu điểm chính là các quy trình không cần biết vị trí của các tài nguyên mà chúng sử dụng, điều này giúp đơn giản hóa việc lập trình ứng dụng. Điều này khiến người lập trình hệ thống phải trả giá, người phải thực hiện các thuật toán phức tạp để cho phép các quy trình giao tiếp và chia sẻ dữ liệu giữa nhiều máy tính, tạo ra ảo tưởng rằng chỉ có một máy tính lớn hơn.