

TRƯỜNG ĐẠI HỌC SƯ PHẠM TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TIỂU LUẬN
MÔN HỆ ĐIỀU HÀNH

Giảng viên hướng dẫn: ThS. Trần Đức Tâm

Sinh viên thực hiện:

1. 47.01.104.114 - Trần Đỗ Anh Khoa
2. 47.01.104.084 - Trần Lê Chí Hải
3. 47.01.104.175 - Quan Quốc Quy
4. 47.01.104.074 - Phan Lương Thùy Dương
5. 47.01.104.186 - Nguyễn Duy Tân

TP. HCM, tháng 05/2023

TRƯỜNG ĐẠI HỌC SƯ PHẠM TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

BÀI TIỂU LUẬN
HỌC PHẦN HỆ ĐIỀU HÀNH

ĐỀ TÀI:

- 1/ Tìm hiểu lịch sử ra đời và các giai đoạn phát triển của hệ điều hành Windows**
- 2/ Tìm hiểu về các vấn đề liên quan đến bài toán phân phối tài nguyên CPU cho các tiến trình. Viết chương trình mô phỏng của một chiến lược lập lịch CPU**

Giảng viên hướng dẫn: ThS. Trần Đức Tâm

Sinh viên thực hiện:

- 1/ 47.01.104.114 - Trần Đỗ Anh Khoa
- 2/ 47.01.104.084 - Trần Lê Chí Hải
- 3/ 47.01.104.175 - Quan Quốc Quy
- 4/ 47.01.104.074 - Phan Lương Thùy Dương
- 5/ 47.01.104.186 - Nguyễn Duy Tân

Học phần: Hệ Điều Hành

TP. HCM, tháng 05/2023

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the entire width of the page. There are no margins, text, or other markings present.

MỤC LỤC

MỤC LỤC	ii
LỜI MỞ ĐẦU	1
CHƯƠNG 1. LỊCH SỬ RA ĐỜI VÀ CÁC GIAI ĐOẠN PHÁT TRIỂN CỦA HỆ ĐIỀU HÀNH WINDOWS	2
1.1. Giới thiệu	2
1.2. Khởi đầu của Windows.....	3
1.3. Các thế hệ Windows	3
1.3.1. Hệ điều hành DOS.....	3
1.3.2. Hệ điều hành Windows 1.X	4
1.3.3. Hệ điều hành Windows 2.X	5
1.3.4. Hệ điều hành Windows 3.X	5
1.3.5. Hệ điều hành Windows 9.X	6
1.3.6. Hệ điều hành Windows For Workgroup	7
1.3.7. Hệ điều hành Windows NT.....	7
1.3.8. Hệ điều hành Windows Me	8
1.3.9. Hệ điều hành Windows 2000	8
1.3.10. Hệ điều hành Windows XP	8
1.3.11. Hệ điều hành Windows Vista.....	9
1.3.12. Hệ điều hành Windows 7	10
1.3.13. Hệ điều hành Windows 8/8.1	11
1.3.14. Hệ điều hành Windows 10	11

1.3.15. Hệ điều hành Windows 11	12
CHƯƠNG 2. TÌM HIỂU VỀ CÁC VẤN ĐỀ LIÊN QUAN ĐẾN BÀI TOÁN PHÂN PHỐI TÀI NGUYÊN CPU CHO CÁC TIẾN TRÌNH. VIẾT CHƯƠNG TRÌNH MÔ PHỎNG HOẠT ĐỘNG CỦA MỘT CHIẾN LƯỢC LẬP LỊCH CPU.....	13
2.1. Tìm hiểu về các vấn đề liên quan đến bài toán phân phối tài nguyên CPU cho các tiến trình.....	13
2.1.1. Giới thiệu.....	13
2.1.2. Mục tiêu phân phối tài nguyên	13
2.1.3. Các vấn đề liên quan đến bài toán phân phối tài nguyên CPU	14
2.2. Viết chương trình mô phỏng hoạt động của 1 chiến lược lập lịch CPU.....	25
2.2.1. Viết chương trình mô phỏng hoạt động của chiến lược lập lịch theo công việc ngắn nhất trước (Shortest-job-first (SJF))	25
2.2.2. Hướng dẫn sử dụng chương trình.....	30
TÀI LIỆU THAM KHẢO	33

LỜI MỞ ĐẦU

Ngày nay, với sự bùng nổ của Công nghệ Thông tin, những thiết bị điện tử ngày càng phát triển đa dạng và phổ biến, trong đó có máy tính. Để quản lý, điều hành toàn bộ thành phần (bao gồm cả phần cứng và phần mềm) của máy tính, chúng ta sử dụng hệ điều hành. Hệ điều hành là một chương trình hay một hệ chương trình hoạt động giữa người sử dụng (user) và phần cứng của máy tính. Mục tiêu của hệ điều hành là cung cấp một môi trường để người sử dụng có thể thi hành các chương trình. Nó làm cho máy tính dễ sử dụng hơn, thuận lợi hơn và hiệu quả hơn.

Với bài tiểu luận này, nhóm chúng em tập trung tìm hiểu về lịch sử ra đời và quá trình phát triển của hệ điều hành Windows cũng như tìm hiểu về các bài toán phân phối tài nguyên CPU cho tiến trình. Trong quá trình hoạt động, hệ điều hành Windows phải đối mặt với nhiều thách thức và vấn đề liên quan đến phân phối tài nguyên CPU cho các tiến trình. Bài toán phân phối tài nguyên CPU là một trong những vấn đề quan trọng nhất trong quản lý tài nguyên hệ thống và đảm bảo hiệu suất hoạt động của máy tính. Các vấn đề quan trọng trong bài toán này bao gồm lập lịch CPU, ưu tiên ưu đãi, thời gian chia sẻ, đồng bộ hóa tiến trình và cân bằng tải. Lập lịch CPU liên quan đến việc quyết định xem tiến trình nào được chạy trên CPU và trong khoảng thời gian nào. Các thuật toán lập lịch CPU như FCFS, SJF, SRTF, Round Robin được sử dụng để quản lý việc phân phối thời gian CPU cho các tiến trình.

Để hoàn thành bài tiểu luận này, nhóm chúng em xin chân thành cảm ơn ThS. Trần Đức Tâm, giảng viên giảng dạy học phần Hệ điều hành. Trong quá trình học, thầy đã truyền đạt cho chúng em những kiến thức quý báu, tạo điều kiện, giúp đỡ chúng em hoàn thiện bài tiểu luận này.

CHƯƠNG 1. LỊCH SỬ RA ĐỜI VÀ CÁC GIAI ĐOẠN PHÁT TRIỂN CỦA HỆ ĐIỀU HÀNH WINDOWS

1.1. Giới thiệu

Microsoft Windows (hoặc đơn giản là Windows) là tên của một hệ điều hành dựa trên giao diện người dùng đồ họa được phát triển và được phân phối bởi Microsoft. Nó bao gồm một vài các dòng hệ điều hành, mỗi trong số đó phục vụ một phân nhất định của ngành công nghiệp máy tính.

Hệ điều hành Windows đã trải qua một quãng thời gian phát triển dài, từ khi ra mắt phiên bản đầu tiên vào năm 1985 cho đến ngày nay, đã trải qua nhiều cải tiến và thay đổi kỹ thuật. Với hơn 25 năm giữ vững ưu thế đối với các máy tính cá nhân, Windows đã trở thành một trong những hệ điều hành phổ biến nhất trên thế giới. Mặc dù thị phần của Windows đã từng bị vượt mặt bởi Android, nhưng dần dà nó đã trở lại và giành lại vị trí dẫn đầu.

Windows có nhiều phiên bản khác nhau, bao gồm Windows 10, Windows 8, Windows 7, Windows Vista và Windows XP,... Mỗi phiên bản có các tính năng và cập nhật riêng, nhằm cải thiện hiệu suất, bảo mật và trải nghiệm người dùng. Windows cung cấp một môi trường cho các ứng dụng phổ biến như trình duyệt web, ứng dụng văn phòng, trò chơi và nhiều ứng dụng khác. Nó cũng hỗ trợ các tiêu chuẩn phần cứng phổ biến và có một cộng đồng lớn của nhà phát triển tạo ra các ứng dụng và phần mềm cho nền tảng này.

Phiên bản Windows đầu tiên được giới thiệu vào năm 1985 như một giao diện đồ họa cho MS-DOS, từ đó đến nay, hệ điều hành này đã trải qua nhiều thay đổi và cải tiến kỹ thuật. Với thị phần lên tới 90% trên toàn thế giới, Windows đã chiếm được lòng tin của người dùng trên nhiều loại thiết bị, từ máy tính cá nhân đến điện thoại di động. Dù đã trải qua nhiều biến động về thị phần, Windows vẫn đang dẫn đầu và trở thành hệ điều hành phổ biến nhất trên toàn thế giới.

1.2. Khởi đầu của Windows

Hệ điều hành Windows đã trải qua một hành trình phát triển rất dài. Từ phiên bản đầu tiên được giới thiệu vào năm 1985 với tên gọi Windows 1.0, nó đã trải qua nhiều thay đổi và cải tiến đáng kể.

Phiên bản đầu tiên của Windows được thiết kế để chạy trên nền tảng MS-DOS, với giao diện đồ họa mới mẻ. Tuy nhiên, phiên bản này còn rất đơn giản và khó sử dụng, đặc biệt là khi không có chuột.

Microsoft đã liên tục cải tiến Windows qua các phiên bản tiếp theo, đưa ra các tính năng mới và cải thiện hiệu suất. Tuy nhiên, không phải tất cả các phiên bản đều đạt được sự thành công như mong đợi. Ví dụ, phiên bản Windows Vista đã gặp nhiều vấn đề khi được phát hành vào năm 2006, nhưng Microsoft đã nhanh chóng cải thiện và phát hành phiên bản Windows 7 vào năm 2009.

Cùng với sự phát triển của công nghệ, Windows đã có sự thay đổi đáng kể về giao diện và tính năng. Windows 10, phiên bản mới nhất của hệ điều hành này được phát hành vào năm 2015, đã có nhiều cải tiến về bảo mật và tính năng, giúp nó trở thành một trong những hệ điều hành phổ biến nhất trên thế giới.

Tổng thể, lịch sử phát triển của hệ điều hành Windows là một câu chuyện về sự phát triển và cải tiến liên tục, đưa nó trở thành một trong những hệ điều hành quan trọng nhất trong lịch sử máy tính.

1.3. Các thế hệ Windows

1.3.1. Hệ điều hành DOS

Windows được phát triển từ hệ điều hành ban đầu của Microsoft là DOS, được ra mắt vào năm 1981 với giao diện hoàn toàn bằng văn bản và các lệnh người dùng giản đơn. Sau đó, khi IBM liên hệ với Microsoft để cung cấp hệ điều hành cho các máy tính IBM vào thời điểm ban đầu, Bill Gates và Paul Allen đã mua QDOS từ Seattle Computer Products và điều chỉnh những cần thiết cho hệ thống máy tính mới.

DOS là một tên chung cho hai hệ điều hành khác nhau: PC DOS khi được đóng gói với các máy tính cá nhân IBM và MS-DOS khi được bán dưới dạng một gói riêng bởi Microsoft. DOS không thân thiện với người dùng, yêu cầu người dùng phải nhớ tất cả các lệnh và sử dụng chúng để thực hiện các hoạt động hàng ngày, chẳng hạn như sao chép các file, thay đổi thư mục,... Tuy nhiên, DOS có ưu điểm là tốc độ nhanh và tiêu tốn ít bộ nhớ, đây là hai vấn đề quan trọng khi hầu hết các máy tính chỉ có 640K bộ nhớ. Với việc phát triển tiếp Windows từ DOS, Microsoft đã tạo ra một hệ điều hành thân thiện hơn với người dùng với giao diện đồ họa và các tính năng dễ sử dụng.

1.3.2. Hệ điều hành Windows 1.X

Microsoft đã tin rằng giao diện đồ họa người dùng (GUI) sẽ trở thành xu thế chủ đạo trong công nghệ máy tính cá nhân, với mục tiêu làm cho máy tính dễ sử dụng hơn và tăng cường sự tin tưởng của người dùng. Để thực hiện quan điểm này, Microsoft đã bắt đầu phát triển phiên bản đầu tiên của Windows vào năm 1983, và sản phẩm cuối cùng đã được phát hành vào tháng 11 năm 1985.

Ban đầu, Windows được gọi là Interface Manager, và nó chỉ là một lớp vỏ đồ họa được đặt lên hệ điều hành DOS đang tồn tại. Khác với DOS, một hệ điều hành dựa trên văn bản và sử dụng bàn phím, Windows 1.0 đã hỗ trợ hoạt động bằng chuột, kích vào thả. Tuy nhiên, giao diện của Windows 1.0 còn khá cứng nhắc và không có tính năng xếp chồng cửa sổ như các phiên bản sau này.

Ban đầu, phiên bản đầu tiên của Windows chỉ có một số ít tiện ích. Nó chỉ bao gồm chương trình đồ họa Paint, trình soạn thảo văn bản Write, bộ lịch, notepad và một đồng hồ. Tuy nhiên, Windows 1.0 đã có Control Panel - thành phần được sử dụng để cấu hình các tính năng khác cho môi trường, và MS-DOS Executive - tiền nhiệm của Windows Explorer ngày nay, là bộ quản lý tập tin.

Không ngạc nhiên khi Windows 1.0 không đạt được thành công như mong đợi, vì vào thời điểm đó, không có nhiều nhu cầu cho một giao diện đồ họa người dùng cho các ứng dụng văn bản trên các máy tính cá nhân của IBM. Ngoài ra, phiên

bản đầu tiên của Windows cũng đòi hỏi công suất tính toán cao hơn so với các máy tính thời đại đó.

1.3.3. Hệ điều hành Windows 2.X

Phiên bản thứ hai của Windows được phát hành vào năm 1987, là một phiên bản cải tiến dựa trên Windows 1.0. Phiên bản mới này đã bổ sung tính năng cho phép xếp chồng cửa sổ và điều chỉnh kích thước cửa sổ để dễ dàng chuyển đổi giữa các ứng dụng trên màn hình desktop bằng chuột.

Windows 2.0 đã tích hợp sẵn các ứng dụng Word và Excel của Microsoft, đây là những ứng dụng đồ họa đang cạnh tranh với WordPerfect và Lotus 1-2-3 vào thời điểm đó. Để có thể chạy hợp lệ, các ứng dụng của Microsoft cần một giao diện đồ họa, vì vậy Microsoft đã tích hợp chúng vào Windows.

Tuy nhiên, vào thời điểm này, vẫn chưa có nhiều ứng dụng tương thích với Windows. Một ngoại lệ đáng chú ý là chương trình Aldus PageMaker.

1.3.4. Hệ điều hành Windows 3.X

Windows 3.0, phiên bản đầu tiên được ra mắt vào năm 1990, đã đạt được những thành công thương mại lớn với hơn 2 triệu bản được bán ra chỉ trong vòng vài tháng đầu tiên. Phiên bản này đã có nhiều cải tiến thiết kế so với các phiên bản trước đó, bao gồm Program Manager, Solitaire, hỗ trợ VGA, bộ nhớ ảo và một giao diện người dùng 3D mới mẻ.

Sau đó, Windows 3.1 được ra mắt cùng với các hướng đi tích cực được thấy từ đó. Windows 3.1 lần đầu tiên hiển thị font TrueType, làm cho nó dần trở thành một nền tảng quan trọng cho các máy tính cá nhân. Phiên bản này cũng đã giới thiệu Windows Workgroups, một phiên bản đặc biệt đi kèm với giao thức mạng ngang hàng. Cùng với Windows 3.11, phiên bản này đã được bán cùng với Windows 3.1.

Mặc dù các phiên bản 3.x chưa hoàn toàn được hoàn thiện, nhưng chúng đã được cải thiện rất nhiều và là những bước đi đầu tiên khẳng định đẳng cấp của Microsoft, cho thấy hướng đi đúng đắn của họ trên con đường phát triển.

1.3.5. Hệ điều hành Windows 9.X

Phiên bản Windows 95 của Microsoft là một sự kiện lịch sử trong ngành công nghệ thông tin. Việc phát hành hệ điều hành này đã kết hợp với chiến dịch truyền thông quảng cáo rộng rãi trên các phương tiện truyền thông đại chúng, đã thu hút hàng dài khách hàng đợi từ nửa đêm để mua những bản copy đầu tiên. Cảnh tượng này được coi như việc xếp hàng mua iPhone trong những năm gần đây. Thành công của Windows 95 vào thời điểm đó đã thể hiện sự mong đợi của người dùng sau nhiều năm chờ đợi.

Điều thực sự đặc biệt ở phiên bản này là diện mạo đẹp, khả năng hoạt động tốt và đáp ứng được mong muốn của người dùng sau nhiều năm chờ đợi. Hầu hết các phần của hệ điều hành này đã được viết lại và cải thiện, đặc biệt là giao diện người dùng.

Thanh taskbar lần đầu tiên xuất hiện trên Windows 95, cho phép người dùng dễ dàng điều chỉnh các cửa sổ mở, các nút start menu, các shortcut trên desktop và tên file lên tới 255 ký tự. Doanh số bán hàng mạnh mẽ của phiên bản này đã củng cố sự thống trị của Microsoft trên thị trường hệ điều hành PC.

Sau thành công của Windows 95, phiên bản Windows 98 được ra mắt vào ngày 25 tháng 6 năm 1998, giới thiệu Windows Driver Model, hỗ trợ các thiết bị USB tổng hợp, ACPI, chế độ ngủ đông và các thiết lập đa màn hình. Windows 98 cũng đi kèm với Internet Explorer 4. Tháng Năm 1999, Microsoft ra mắt Windows 98 Second Edition, một bản cập nhật cho Windows 98, bổ sung Internet Explorer 5 và Windows Media Player 6.2 cùng với nhiều cải tiến khác.

Trình duyệt Internet Explorer đã trở thành một phần không thể thiếu của Windows, mặc dù sau này nó đã trở nên lỗi thời và trở thành chủ đề của các meme hài hước. Tuy nhiên, nó vẫn được coi là một huyền thoại trong lòng người dùng máy tính. Các phiên bản Windows 95 và Windows 98 đã đạt được sự thành công vượt trội, giúp Microsoft thống trị thị trường hệ điều hành trong thập niên 1990.

1.3.6. Hệ điều hành Windows For Workgroup

Windows cho các nhóm làm việc (Windows for Workgroups - viết tắt là WFW) là một phiên bản của Windows được phát hành vào năm 1992, nhằm mục đích đầu tiên để kết nối mạng. Ban đầu được phát triển là một add-on của Windows 3.0, tuy nhiên WFW đã được bổ sung thêm các driver và giao thức cần thiết như TCP/IP để hỗ trợ kết nối mạng ngang hàng. Phiên bản này đặc biệt phù hợp với môi trường công ty.

Với WFW, Windows đã được chia thành hai hướng: một hướng dành cho khách hàng, được thiết kế để sử dụng trên các máy tính cá nhân độc lập, gồm Windows 3.1 và phiên bản Windows 95 đang sắp ra mắt; và một hướng dành cho doanh nghiệp, được thiết kế để sử dụng trên các máy tính được kết nối mạng, gồm WFW và phiên bản Windows NT đang sắp ra mắt.

1.3.7. Hệ điều hành Windows NT

Windows NT là một dòng hệ điều hành được Microsoft sản xuất, với phiên bản đầu tiên được phát hành vào tháng 7 năm 1993. Đây là một hệ điều hành vi xử lý độc lập, đa nhiệm và đa người dùng.

Phiên bản đầu tiên trong dòng hệ điều hành này là NT 3.1, được phát hành vào tháng 7 năm 1993, cho phép máy trạm và máy chủ hoạt động. NT 3.5 được phát hành vào tháng 9 năm 1994, sau đó là NT 4.0 vào tháng 6 năm 1996, với giao diện thiết kế mới được lấy cảm hứng từ Windows 95 trên nền tảng NT.

Windows 2000 ra mắt vào ngày 17 tháng 2 năm 2000, là phiên bản kế vị ngay sau Windows NT. Windows 2000 là một sự tiến hóa từ nền tảng NT, và vẫn nhắm đến thị trường doanh nghiệp, đồng thời loại bỏ nhu cầu khởi động lại thường xuyên khi cài đặt phần mềm hoặc thay đổi cấu hình hệ thống. Windows 2000 đã được đón nhận và thành công trong việc hỗ trợ doanh nghiệp, và tất nhiên cũng có thể sử dụng trên các máy tính cá nhân vào thời điểm đó. Khác với Windows NT, phiên bản 2000 có nhiều phiên bản khác nhau từ Professional, Server, Advanced Server, Datacenter Server cho đến Small Business Server. Tất cả các phiên bản này đều hoạt động ổn

định và được thừa kế các tính năng từ Windows 95 và 98, nên có một giao diện đẹp hơn so với thời điểm đó.

1.3.8. Hệ điều hành Windows Me

Microsoft phát hành phiên bản Windows Millennium Edition (Windows Me) vào năm 2000. Tuy nhiên, đây được coi là một lỗi lớn của Microsoft vì đây chỉ là một phiên bản nâng cấp thứ yếu với rất nhiều lỗi thay vì sửa các lỗi trước đó.

Trong phiên bản Windows Me, Microsoft đã nâng cấp các tính năng Internet và đa phương tiện của Windows 98, bổ sung thêm ứng dụng Windows Movie Maker và giới thiệu tính năng System Restore - tất cả đều là các ứng dụng tiện ích. Tuy nhiên, điều đáng chú ý nhất trong Windows Me là sự dễ đổ vỡ và hệ thống dễ bị treo. Điều này đã khiến nhiều khách hàng và doanh nghiệp bỏ qua toàn bộ nâng cấp này.

1.3.9. Hệ điều hành Windows 2000

Windows 2000, được phát hành cùng thời điểm với Windows Me dành cho người dùng cá nhân, là một phiên bản nâng cấp thành công dành cho khối doanh nghiệp của Microsoft. Là sự tiếp nối của Windows NT, Windows 2000 là một bước tiến từ nền tảng NT gốc, vẫn hướng đến thị trường doanh nghiệp.

Khác với NT, Windows 2000 có hai phiên bản (Workstation và Server), và có tới 5 phiên bản khác nhau: Professional, Server, Advanced Server, Datacenter Server và Small Business Server. Tất cả các phiên bản đều kết hợp một cách hài hòa các tính năng từ Windows 95/98, tạo nên một giao diện đẹp mắt và hiện đại.

1.3.10. Hệ điều hành Windows XP

Vào năm 2001, Microsoft đã giới thiệu phiên bản Windows XP, đây là phiên bản đầu tiên mà họ đem sự tin cậy của dòng doanh nghiệp vào thị trường khách hàng, đồng thời mang tính thân thiện vào thị trường doanh nghiệp. Windows XP đã kết hợp tốt nhất giữa các phiên bản Windows 95/98/Me với tính năng thao tác 32-bit của Windows NT/2000 và giao diện người dùng được cải tiến. Về cơ bản, Windows XP có thể được coi là sự kết hợp giữa giao diện của Windows 95/98/Me với nhân hạt

nhân (core) của Windows NT/2000, bỏ qua cơ sở mã DOS đã xuất hiện trong các phiên bản khách hàng trước đó của Windows.

Windows XP đã được phân khúc thị trường với nhiều phiên bản khác nhau, bao gồm XP Home Edition, XP Professional (dành cho người dùng doanh nghiệp), XP Media Center Edition, XP Tablet PC Edition và XP Starter Edition (dành cho người dùng trong các nước đang phát triển). Mặc dù việc phân khúc này đã khiến nhiều người dùng cảm thấy lúng túng, nhưng Microsoft dường như không quá quan tâm đến điều đó.

Từ quan điểm của người dùng, Windows XP là một phiên bản đẹp hơn, nhanh hơn so với các phiên bản trước đó như Windows 95/98 hay Windows 2000. (Nó cũng đáng tin cậy hơn so với hệ điều hành Windows Me không thành công trước đó). Giao diện Luna của nó mang lại trải nghiệm đẹp hơn và thân thiện hơn, tính năng Fast User Switching cho phép chia sẻ máy tính dễ dàng với người dùng khác.

1.3.11. Hệ điều hành Windows Vista

Windows Vista là phiên bản hệ điều hành được phát hành vào năm 2007, với việc phát triển các tính năng của Windows XP và bổ sung thêm tính năng bảo mật và độ tin cậy, cải tiến chức năng truyền thông, và đưa vào giao diện người dùng Aero 3D đẹp mắt.

Tuy nhiên, để sử dụng giao diện nâng cao này, máy tính cần có cấu hình cao, dẫn đến việc Vista không được hỗ trợ nâng cấp từ nhiều máy tính cũ. Giao diện Aero hiển thị các thành phần 3D gần như trong suốt và đường bao cửa sổ kiểu kính, cũng như các biểu tượng thư mục và file hiển thị bằng các thumbnail nội dung của chúng. Khi chuyển giữa các ứng dụng đang mở, Windows sẽ cuộn và xoay vòng để hiển thị theo ngăn xếp 3D, tạo cảm giác uyển chuyển và sâu sắc.

Tuy nhiên, Vista cũng gặp phải một số vấn đề, như tính năng User Account Control gây khó chịu cho người dùng vì yêu cầu quá nhiều xác nhận, cũng như việc nhiều thiết bị ngoại vi cũ không có driver tương thích, và một số chương trình chạy trên XP không hoạt động tốt trên Vista.

Vì những hạn chế và nhược điểm trên, Microsoft đã nhanh chóng phát triển phiên bản kế nhiệm cho Vista là Windows 7, được phát hành vào ngày 22/10/2009. Windows 7 đã được hưởng sự thành công lớn và được sử dụng rộng rãi, và vào ngày 11/4/2017, Windows Vista chính thức được Microsoft cho nghỉ hưu.

1.3.12. Hệ điều hành Windows 7

Windows 7, phiên bản mới nhất dự kiến được phát hành vào tháng 10 năm 2009, được xem như là một bước tiến đáng kể so với Windows Vista. Không giống như khoảng thời gian hai năm ngắn ngủi giữa Windows Vista và phiên bản mới nhất, Windows 7 được coi là một bản nâng cấp đáng kể hơn là chỉ là một gói dịch vụ.

Một số điểm khác biệt chính giữa Windows 7 và Windows Vista là sự tương thích phần cứng và phần mềm. Windows 7 được cải thiện để tương thích tốt hơn với phần cứng cũ và phần mềm cũ, và còn có tính năng Windows XP Mode cho phép chạy các ứng dụng từ thời đại Windows XP trực tiếp trên Windows 7. User Account Control cũng được cải thiện để giảm bớt sự gây khó chịu cho người dùng.

Ngoài ra, giao diện người dùng của Windows 7 cũng có một số thay đổi đáng chú ý. Sidebar đã được loại bỏ và thay thế bằng khả năng đặt các Gadget trực tiếp lên desktop. Windows 7 cũng giới thiệu chế độ Aero Peek mới cho phép người dùng xem "đăng sau" tất cả các cửa sổ đang mở để xem nội dung bên dưới desktop, cũng như các tính năng Aero Snap mới cho phép dễ dàng di chuyển và tối đa hóa các cửa sổ.

Tuy nhiên, thay đổi lớn nhất lại nằm ở taskbar, một dải thanh cố định trên màn hình xuất hiện lần đầu tiên trong Windows 95. Taskbar mới trong Windows 7 cho phép người dùng đính kèm các cửa sổ đang mở và các ứng dụng yêu thích của họ vào taskbar. Bằng cách chuột phải vào một nút trên taskbar, người dùng có thể xem một Jump List các tài liệu gần đây và các hoạt động khác; di chuột qua nút taskbar, người dùng sẽ thấy các ứng dụng đang hoạt động và các thumbnail của tất cả các tài liệu. Có thể nói Windows 7 đã thay đổi cách người dùng thực hiện nhiều công việc, và được đánh giá cao bởi nhiều người dùng.

1.3.13. Hệ điều hành Windows 8/8.1

Windows 8/8.1, được phát hành vào ngày 26/10/2012. Microsoft đã cố gắng tạo ra một hệ điều hành lai (hybrid OS) có thể hoạt động trên cả thiết bị cảm ứng và không cảm ứng để đạt được vị thế trong thế giới di động, bao gồm cả điện thoại và máy tính bảng. Tuy nhiên, hiệu quả mà Windows 8 mang lại không như mong đợi. Nút Start đã bị loại bỏ, giao diện người dùng được gọi là Metro nhưng sau đó phải đổi tên thành Modern do bị kiện kiện bởi một công ty ở châu Âu, và không được đón nhận nồng nhiệt. Tất cả những điều này đã làm cho Windows 8 trở thành một thất bại, không thể thu hút được người dùng từ phiên bản trước đó mà họ đã hài lòng với nó.

Thực tế không rõ ràng, Windows 8/8.1 đã có sự thay đổi trong giao diện người dùng và tên gọi giao diện từ Metro thành Modern, tuy nhiên, đây không được đón nhận tích cực.

1.3.14. Hệ điều hành Windows 10

Vào cuối năm 2015, Windows 10 được ra mắt chính thức tại San Francisco, đây được coi là một phiên bản hoàn hảo với tính năng "tất cả trong một", cho phép các thiết bị cảm ứng và không cảm ứng đều có thể sử dụng chung. Giao diện của Windows 10 đã được thay đổi và được đánh giá là đẹp nhất từ trước đến nay, đặc biệt là tính năng Menu Start đã được nâng cấp mạnh mẽ.

Chức năng của Menu Start được xem là quyền lực nhất của Windows 10, tiếp đó là Settings (Cài đặt) thay thế cho Control Panel (Bảng điều khiển). Cả hai tính năng này hoạt động song song và hỗ trợ lẫn nhau. Phiên bản thử nghiệm Technical Preview của Windows 10 đã nhận được đánh giá cao từ cộng đồng người dùng.

Với các bản cập nhật liên tục từ khi ra mắt, Windows 10 đã cải thiện và thêm nhiều tính năng mới, đến mức có thể nói đây là phiên bản hoàn hảo nhất cho đến thời điểm hiện tại, được sử dụng rộng rãi bởi đa số người dùng ngày nay. Có khả năng Microsoft sẽ tập trung vào việc cải tiến và nâng cấp phiên bản hiện tại thay vì phát hành các phiên bản mới trong tương lai. Tuy nhiên, các chuyên gia phát triển của

Microsoft vẫn đang làm việc trên các phiên bản hệ điều hành lõi mới, vì vậy chúng ta có thể mong đợi nhiều điều thú vị trong tương lai gần.

1.3.15. Hệ điều hành Windows 11

Dường như Windows 10 sẽ là phiên bản cuối cùng trong lịch sử của hệ điều hành Windows, và tương lai sẽ chỉ là các phiên bản cải tiến với tên gọi và số phiên bản khác nhau để đồng bộ hóa tính năng và giao diện.

Tuy nhiên, vào tháng 6/2021, Microsoft đã giới thiệu phiên bản Windows 11 với nhiều cải tiến đáng chú ý. Vào tháng 10/2021, Windows 11 đã được phát hành rộng rãi cho người dùng trên toàn cầu.

Mặc dù vậy, các nhà phát triển của Microsoft vẫn đang nỗ lực làm việc trên các phiên bản mới của hệ điều hành Windows, cho thấy họ không dừng lại ở đây. Do đó, chúng ta có thể mong đợi sẽ có những điều thú vị trong tương lai gần và Windows sẽ tiếp tục phát triển không ngừng.

CHƯƠNG 2. TÌM HIỂU VỀ CÁC VẤN ĐỀ LIÊN QUAN ĐẾN BÀI TOÁN PHÂN PHỐI TÀI NGUYÊN CPU CHO CÁC TIẾN TRÌNH. VIẾT CHƯƠNG TRÌNH MÔ PHỎNG HOẠT ĐỘNG CỦA MỘT CHIẾN LƯỢC LẬP LỊCH CPU

2.1. Tìm hiểu về các vấn đề liên quan đến bài toán phân phối tài nguyên CPU cho các tiến trình.

2.1.1. Giới thiệu

Trong môi trường đa chương, có thể xảy ra tình huống nhiều tiến trình đồng thời sẵn sàng để xử lý. Mục tiêu của các hệ phân chia thời gian (time-sharing) là chuyển đổi CPU qua lại giữa các tiến trình một cách thường xuyên để nhiều người sử dụng có thể tương tác cùng lúc với từng chương trình trong quá trình xử lý.

Để thực hiện được mục tiêu này, hệ điều hành phải lựa chọn tiến trình được xử lý tiếp theo. Bộ điều phối sẽ sử dụng một giải thuật điều phối thích hợp để thực hiện nhiệm vụ này. Một thành phần khác của hệ điều hành cũng tiềm ẩn trong công tác điều phối là bộ phân phối (dispatcher). Bộ phân phối sẽ chịu trách nhiệm chuyển đổi ngữ cảnh và trao CPU cho tiến trình được chọn bởi bộ điều phối để xử lý.

Phân phối tài nguyên CPU cho các tiến trình là một vấn đề quan trọng trong quản lý tài nguyên hệ thống. Khi chạy nhiều tiến trình trên một hệ thống đa nhiệm, việc phân phối công bằng và hiệu quả tài nguyên CPU giữa các tiến trình là cần thiết để đảm bảo sự công bằng và hiệu suất của hệ thống.

2.1.2. Mục tiêu phân phối tài nguyên

Bộ điều phối không cung cấp cơ chế, mà đưa ra các quyết định. Các hệ điều hành xây dựng nhiều chiến lược khác nhau để thực hiện việc điều phối, nhưng tựu chung cần đạt được các mục tiêu sau :

- Độ lợi CPU (Efficiency): Hệ thống phải tận dụng được CPU 100% thời gian.
- Thời gian đáp ứng hợp lý (Response time): Tối thiểu hoá thời gian hồi đáp cho các tương tác của người sử dụng

- Thời gian chờ (Waiting time): Tối thiểu hóa thời gian chờ khi nằm trong hàng đợi.
- Thời gian lưu lại trong hệ thống (Turnaround Time): Tối thiểu hóa thời gian hoàn tất các tác vụ xử lý theo lô.
- Thông lượng tối đa (Throughput): Tối đa hóa số công việc được xử lý trong một đơn vị thời gian.

Tuy nhiên thường không thể thỏa mãn tất cả các mục tiêu kể trên vì bản thân chúng có sự mâu thuẫn với nhau mà chỉ có thể dung hòa chúng ở mức độ nào đó.

2.1.3. Các vấn đề liên quan đến bài toán phân phối tài nguyên CPU

Phân phối tài nguyên CPU cho các tiến trình là một vấn đề quan trọng trong quản lý tài nguyên hệ thống. Khi chạy nhiều tiến trình trên một hệ thống đa nhiệm, việc phân phối công bằng và hiệu quả tài nguyên CPU giữa các tiến trình là cần thiết để đảm bảo sự công bằng và hiệu suất của hệ thống. Dưới đây là một số khía cạnh quan trọng trong việc giải quyết bài toán phân phối tài nguyên CPU.

2.1.3.1. Lập lịch CPU

Lập lịch CPU quyết định thứ tự các tiến trình được thực thi trên CPU. Các thuật toán lập lịch khác nhau như First-Come First-Served (FCFS), Shortest Job First (SJF), Round Robin (RR) và Shortest Remaining Time First (SRTF) có các chiến lược khác nhau trong việc phân phối tài nguyên CPU.

a. Thuật toán SJF (Shortest-job-first)

Nguyên tắc: Đây là một trường hợp đặc biệt của giải thuật điều phối với độ ưu tiên. Trong giải thuật này, độ ưu tiên p được gán cho mỗi tiến trình là nghịch đảo của thời gian xử lý t mà tiến trình yêu cầu: $p = 1/t$. Khi CPU được tự do, nó sẽ được cấp phát cho tiến trình yêu cầu ít thời gian nhất để kết thúc- tiến trình ngắn nhất. Giải thuật này cũng có thể độc quyền hay không độc quyền. Sự chọn lựa xảy ra khi có một tiến trình mới được đưa vào danh sách sẵn sàng trong khi một tiến trình khác đang xử lý. Tiến trình mới có thể sở hữu một yêu cầu thời gian sử dụng CPU cho lần tiếp theo (CPU-burst) ngắn hơn thời gian còn lại mà tiến trình hiện hành cần xử lý. Giải

thuật SJF không độc quyền sẽ dừng hoạt động của tiến trình hiện hành, trong khi giải thuật độc quyền sẽ cho phép tiến trình hiện hành tiếp tục xử lý.

Ví dụ :

Tiến trình	Thời điểm vào RL	Thời gian xử lý
P1	0	6
P2	1	8
P3	2	4
P4	3	2

Sử dụng thuật giải SJF độc quyền, thứ tự cấp phát CPU như sau:

P1	P4	P3	P2
0	6	8	12 20

Sử dụng thuật giải SJF không độc quyền, thứ tự cấp phát CPU như sau:

P1	P4	P1	P3	P2
0	3	5	8	12 20

Ưu điểm:

- Giải thuật được xem là tối ưu, thời gian chờ đợi trung bình giảm
- Tận dụng hết năng lực của CPU

Nhược điểm:

- Cài đặt thuật toán phức tạp, tốn nhiều xử lý cho quá trình quản lý.
- Mặc dù SJF là tối ưu nhưng nó không thể được cài đặt tại cấp định thời CPU ngắn vì không có cách nào để biết chiều dài chu kỳ CPU tiếp theo.
- Giải thuật SJF có thể trung dụng hoặc không trung dụng CPU, dẫn tới giải thuật này có nhiều dị bản khác nhau và sẽ tối ưu hay không tối ưu phụ thuộc vào trung dụng CPU.

b. Thuật toán FCFS (First-Come, First-Served)

Nguyên tắc: của thuật toán FCFS là tiến trình nào đến trước sẽ được thực hiện trước. Các bước thực hiện: Đưa tất cả các tiến trình vào một hàng đợi (queue) theo thứ tự đến của chúng. Bắt đầu thực hiện tiến trình đầu tiên trong hàng đợi. Khi một tiến trình được thực hiện, nó sẽ chạy cho đến khi hoàn thành. Khi tiến trình hiện tại hoàn thành, tiến trình tiếp theo trong hàng đợi sẽ được lựa chọn để thực hiện. Quá trình này tiếp tục cho đến khi tất cả các tiến trình trong hàng đợi được thực hiện hoàn tất.

Ví dụ :

Tiến trình	Thời điểm vào RL	Thời gian xử lý
P1	0	10
P2	1	1
P3	2	2
P4	3	1
P5	4	5

Sử dụng thuật giải FCFS, thứ tự cấp phát CPU như sau:

P1	P2	P3	P4	P5
0	10	11	13	14 19

Ưu điểm của thuật toán FCFS là đơn giản và dễ hiểu. Tuy nhiên, thuật toán FCFS có một số hạn chế như: Thời gian chờ (waiting time) có thể lớn đối với các tiến trình có thời gian thực hiện lâu hơn đến trước, không tối ưu về mặt thời gian chờ, do không xem xét thời gian thực hiện của các tiến trình, có thể xảy ra hiện tượng lập lịch không công bằng (convoy effect) khi một tiến trình dài hạn chặn các tiến trình ngắn hạn phía sau nó.

Thuật toán FCFS là một thuật toán đơn giản và không tối ưu trong việc lập lịch CPU. FCFS thực hiện các tiến trình theo thứ tự đến của chúng, mà không xem xét thời gian thực hiện hoặc độ ưu tiên.

c. Thuật toán RR (Round Robin)

Nguyên tắc : Danh sách sẵn sàng được xử lý như một danh sách vòng, bộ điều phối lần lượt cấp phát cho từng tiến trình trong danh sách một khoảng thời gian sử dụng CPU gọi là quantum. Đây là một giải thuật điều phối không độc quyền : khi một tiến trình sử dụng CPU đến hết thời gian quantum (q) dành cho nó, hệ điều hành thu hồi CPU và cấp cho tiến trình kế tiếp trong danh sách. Nếu tiến trình bị khóa hay kết thúc trước khi sử dụng hết thời gian quantum (q), hệ điều hành cũng lập tức cấp phát CPU cho tiến trình khác. Khi tiến trình tiêu thụ hết thời gian CPU dành cho nó mà chưa hoàn tất, tiến trình được đưa trở lại vào cuối danh sách sẵn sàng để đợi được cấp CPU trong lượt kế tiếp.

Ví dụ:

Tiến trình	Thời điểm vào RL	Thời gian xử lý
P1	0	24
P2	1	3
P3	2	3

Nếu sử dụng $q=4$, thứ tự cấp phát CPU sẽ là :

P1	P2	P3	P1	P1	P1	P1	P1
0	4	7	10	14	18	22	26 30

Thời gian chờ đợi trung bình sẽ là $(0+6+3+5)/3 = 4.66$

Nếu có n tiến trình trong danh sách sẵn sàng và sử dụng quantum q, thì mỗi tiến trình sẽ được cấp phát CPU $1/n$ trong từng khoảng thời gian q. Mỗi tiến trình sẽ không phải đợi quá $(n-1)q$ đơn vị thời gian trước khi nhận được CPU cho lượt kế tiếp.

Ưu điểm:

Các quá trình sẽ được luân phiên cho CPU xử lý nên thời gian chờ đợi sẽ ít.

Đối với các quá trình liên quan đến nhập xuất, IO, người dùng thì rất hiệu quả.

Việc cài đặt không quá phức tạp

Nhược điểm:

Thời gian chờ đợi trung bình dưới chính sách RR thường là quá dài.

Nếu thời gian định mức cho việc xử lý quá lớn thì RR thành FIFO

Nếu thời gian quá ngắn so với thời gian xử lý của một tiến trình trong danh sách hàng đợi thì việc chờ đợi và xử lý luân phiên sẽ nhiều.

Qui tắc là định mức thời gian nên dài hơn 80% chu kỳ CPU.

d. Thuật toán SRTF (Shortest-Remainin-Time-First)

Nguyên tắc: Trong SRTF, quá trình thực thi có thể bị dừng sau một khoảng thời gian nhất định. Khi xuất hiện mọi quy trình, bộ lập lịch ngắn hạn lập lịch cho quy trình với thời gian liên tục còn lại ít nhất trong danh sách các quy trình có sẵn và quy trình đang chạy.

Ví dụ:

Tiến trình	Thời điểm vào	Thời gian xử lý
1	0	9
2	1	4
3	2	9

P1	P2	P2	P1	P3
0	1	2	5	13 22

Thời gian chờ trung bình = $(4+0+11)/3=5$.

Ưu điểm:

- Thời gian chờ đợi ngắn: Với thuật toán SRTF, tiến trình có thời gian xử lý còn lại nhỏ nhất sẽ được ưu tiên thực thi. Điều này dẫn đến thời gian chờ đợi trung bình của các tiến trình được giảm, giúp tăng hiệu suất của hệ thống.
- Thực thi ưu tiên cho các tiến trình ngắn: SRTF ưu tiên thực thi các tiến trình có thời gian xử lý ngắn, giúp hoàn thành nhanh chóng các tiến trình ngắn và giải phóng tài nguyên CPU. Điều này có thể dẫn đến việc cải thiện thời gian đáp ứng của hệ thống.
- Công bằng: SRTF đảm bảo rằng các tiến trình ngắn không bị ưu ái quá mức, mà chỉ được thực thi trong khoảng thời gian cần thiết để hoàn thành. Điều này giúp đảm bảo tính công bằng trong việc phân phối tài nguyên CPU.

Nhược điểm:

- Overhead lập lịch: Thuật toán SRTF đòi hỏi việc kiểm tra thời gian xử lý còn lại của các tiến trình liên tục và thực hiện lập lịch CPU liên tục. Điều này tạo ra một overhead lập lịch, gây tốn thời gian và tài nguyên CPU.
- Độ tin cậy: SRTF dựa vào thông tin về thời gian xử lý còn lại của các tiến trình. Nếu thông tin này không chính xác hoặc bị giả mạo, thuật toán có thể gặp vấn đề trong việc ưu tiên thực thi các tiến trình.
- Thích ứng kém với tiến trình dài: SRTF tập trung vào việc thực thi các tiến trình ngắn để giảm thời gian chờ đợi. Điều này có thể làm cho các tiến trình dài bị lãng phí thời gian, vì chúng có thể bị chờ đợi lâu hơn do sự ưu tiên của các tiến trình ngắn.

2.1.3.2. Ưu tiên tiến trình

Trong phân phối tài nguyên, ưu tiên tiến trình là quá trình ưu tiên cung cấp tài nguyên hệ thống cho một tiến trình hoặc một nhóm tiến trình hơn các tiến trình khác. Điều này có thể được thực hiện để tăng hiệu suất, cải thiện đáp ứng hoặc đảm bảo sự ưu tiên cho các tiến trình quan trọng hơn.

Có một số cách để ưu tiên tiến trình trong phân phối tài nguyên, bao gồm:

Ưu tiên CPU: Một cách phổ biến để ưu tiên tiến trình là ưu tiên phân phối thời gian CPU cho tiến trình quan trọng hơn. Điều này có thể được thực hiện bằng cách sử dụng các thuật toán lập lịch như Round Robin, Priority Scheduling hoặc Completely Fair Scheduler để đảm bảo rằng các tiến trình ưu tiên nhận được sự ưu tiên cao hơn khi cung cấp CPU.

Ưu tiên bộ nhớ: Tiến trình quan trọng có thể được ưu tiên với việc cung cấp ưu tiên cao hơn trong việc sử dụng bộ nhớ hệ thống. Hệ điều hành có thể sử dụng các thuật toán quản lý bộ nhớ như First Fit, Best Fit hoặc Worst Fit để đảm bảo tiến trình ưu tiên nhận được ưu tiên trong việc cấp phát và giải phóng bộ nhớ.

Ưu tiên I/O: Đối với các tiến trình thực hiện nhiều hoạt động I/O, có thể cung cấp ưu tiên cao hơn trong việc sử dụng tài nguyên I/O. Điều này đảm bảo rằng các tiến trình thực hiện nhiều hoạt động đọc/ghi dữ liệu hoặc tương tác với các thiết bị ngoại vi nhận được sự ưu tiên trong việc sử dụng tài nguyên I/O.

Ưu tiên mạng: Trong hệ thống mạng, tiến trình có thể được ưu tiên trong việc sử dụng băng thông mạng hoặc ưu tiên truyền dữ liệu. Điều này có thể đảm bảo rằng các tiến trình quan trọng trong mạng như máy chủ dịch vụ hoặc ứng dụng mạng yêu cầu được ưu tiên trong việc sử dụng tài nguyên.

Ưu tiên tài nguyên khác: Ngoài CPU, bộ nhớ, I/O và mạng, còn có các tài nguyên khác trong hệ thống như đầu vào từ thiết bị ngoại vi, nguồn điện, băng thông mạng và tài nguyên phần cứng khác. Các tiến trình quan trọng có thể được ưu tiên trong việc sử dụng các tài nguyên này để đảm bảo chất lượng dịch vụ và đáp ứng yêu cầu của họ.

Quản lý ưu tiên: Hệ điều hành cung cấp cơ chế quản lý ưu tiên tiến trình, cho phép quản lý và điều chỉnh mức độ ưu tiên của các tiến trình. Qua đó, các tiến trình quan trọng có thể được gán mức độ ưu tiên cao hơn, từ đó nhận được sự ưu tiên trong việc cung cấp tài nguyên hệ thống.

Tài nguyên hạn chế: Trong trường hợp tài nguyên hệ thống hạn chế, quá trình ưu tiên tiến trình trở nên quan trọng hơn bao giờ hết. Hệ điều hành phải xác định các

tiến trình quan trọng và ưu tiên cung cấp tài nguyên cho chúng, nhằm đảm bảo hoạt động ổn định của hệ thống.

Tuy nhiên, quá mức ưu tiên tiến trình cũng có thể gây ra các vấn đề khác như ưu tiên quá mức dẫn đến xung đột tài nguyên, gây ảnh hưởng đến các tiến trình khác hoặc làm suy yếu hiệu suất tổng thể của hệ thống. Do đó, việc ưu tiên tiến trình cần được thực hiện cân nhắc và điều chỉnh sao cho hợp lý, tùy thuộc vào yêu cầu cụ thể của từng hệ thống và ứng dụng.

2.1.3.3. Công bằng phân phối

Đối với một hệ thống đa người dùng, công bằng phân phối tài nguyên CPU là một yếu tố quan trọng. Việc sử dụng các thuật toán lập lịch như Fair Share Scheduling hoặc Weighted Round Robin giúp đảm bảo mỗi tiến trình được cung cấp một lượng tài nguyên CPU tương đối công bằng. Công bằng trong các bài toán phân phối tài nguyên CPU đề cập đến việc đảm bảo sự công bằng và hiệu quả trong việc phân phối thời gian xử lý CPU cho các tiến trình. Mỗi thuật toán có cách tiếp cận khác nhau để đảm bảo công bằng và tối ưu hóa việc sử dụng tài nguyên CPU. Dưới đây là một tóm tắt về công bằng trong các thuật toán phân phối tài nguyên CPU phổ biến:

- **FCFS (First-Come, First-Served):** Đây là thuật toán đơn giản nhất, nơi tiến trình nào đến trước sẽ được xử lý trước. Tuy nhiên, FCFS không đảm bảo công bằng vì nếu một tiến trình có thời gian xử lý lâu đến trước, nó sẽ chiếm lấy tài nguyên CPU trong thời gian dài, gây chậm tiến trình khác.
- **SJF (Shortest Job First):** SJF ưu tiên xử lý các tiến trình có thời gian xử lý ngắn nhất trước. Điều này giúp đảm bảo rằng các tiến trình ngắn sẽ hoàn thành nhanh chóng, tối ưu hóa thời gian chờ và thời gian phản hồi. Tuy nhiên, SJF có thể gây ra hiện tượng "đói" tài nguyên cho các tiến trình dài hơn, vì chúng phải chờ đợi các tiến trình ngắn hoàn thành.
- **SRTF (Shortest Remaining Time First):** SRTF tương tự như SJF, nhưng thay vì xử lý toàn bộ tiến trình ngắn trước, SRTF ưu tiên xử lý tiến trình có thời gian xử lý còn lại ngắn nhất. Điều này giúp đảm bảo công bằng hơn vì các tiến trình có thời gian xử lý ngắn hơn vẫn được ưu tiên, ngay cả khi đã bắt đầu chạy.

- **RR (Round Robin):** RR là một thuật toán lập lịch dựa trên việc chia thời gian CPU thành các đợt nhỏ và chạy lần lượt các tiến trình trong mỗi đợt. Mỗi tiến trình có một đơn vị thời gian để chạy (thường gọi là quantum). Sau khi một tiến trình đã chạy trong một quantum, nó được dừng lại và tiến trình tiếp theo được chạy. Thuật toán RR giúp đảm bảo công bằng vì mỗi tiến trình được cung cấp một lượng tài nguyên CPU nhất định, và không có tiến trình nào chiếm lấy tài nguyên CPU quá lâu.

Tuy cách công bằng được định nghĩa trong từng thuật toán có thể khác nhau, nhưng mục tiêu chung là đảm bảo rằng mỗi tiến trình nhận được một lượng tài nguyên CPU hợp lý và đảm bảo thời gian chờ, thời gian phản hồi và hiệu quả sử dụng tài nguyên CPU tối ưu.

2.1.3.4. Thời gian chia

Đôi khi, thay vì cung cấp toàn bộ tài nguyên CPU cho một tiến trình, hệ điều hành chia nhỏ thời gian chạy CPU thành các đơn vị nhỏ hơn (ví dụ: time slices) và chia sẻ chúng giữa các tiến trình. Điều này cho phép các tiến trình thực thi xen kẽ và tạo cảm giác đồng thời. Trong các bài toán phân phối tài nguyên CPU, thời gian chia (time quantum) được sử dụng trong thuật toán Round Robin (RR). Thời gian chia định nghĩa khoảng thời gian tối đa mà một tiến trình có thể sử dụng CPU liên tục trước khi nó phải nhường CPU cho tiến trình khác.

Thời gian chia trong thuật toán RR có vai trò quan trọng trong việc cân bằng công việc giữa các tiến trình. Khi một tiến trình đang chạy, nếu thời gian chạy của nó chưa hết mà đã hết thời gian chia, nó sẽ bị tạm dừng và chuyển sang tiến trình tiếp theo trong hàng đợi. Tiến trình hiện tại được đưa lại vào cuối hàng đợi và tiến trình tiếp theo được chọn để chạy.

Thời gian chia nên được chọn một cách cân nhắc để đảm bảo sự cân bằng giữa thời gian chờ và hiệu suất của hệ thống. Nếu thời gian chia quá nhỏ, thì việc chuyển đổi giữa các tiến trình sẽ xảy ra quá thường xuyên, dẫn đến độ trễ và tăng thời gian chuyển đổi. Ngược lại, nếu thời gian chia quá lớn, một tiến trình có thể chiếm lấy CPU trong một thời gian dài, gây chậm tiến trình khác.

Thời gian chia thường được định nghĩa như một đơn vị thời gian như millisecond hoặc microsecond. Việc lựa chọn thời gian chia thích hợp phụ thuộc vào yêu cầu cụ thể của hệ thống và đặc điểm của các tiến trình đang chạy.

Tóm lại, thời gian chia trong các bài toán phân phối tài nguyên CPU được sử dụng trong thuật toán Round Robin để quy định khoảng thời gian mà mỗi tiến trình được sử dụng CPU liên tục trước khi nhường cho tiến trình khác.

2.1.3.5. Đo lường hiệu suất

Trong bài toán phân phối tài nguyên CPU, hiệu suất thường được đo lường bằng các thông số sau đây:

- Thời gian chờ trung bình (Average Waiting Time): Đây là thời gian trung bình mà một tiến trình phải chờ đợi để được sử dụng CPU. Thời gian chờ trung bình càng thấp thì tiến trình được phục vụ nhanh chóng hơn.
- Thời gian xoay vòng trung bình (Average Turnaround Time): Đây là thời gian trung bình mà một tiến trình bắt đầu từ khi nó được sử dụng CPU cho đến khi hoàn thành. Thời gian xoay vòng trung bình bao gồm thời gian chờ và thời gian thực thi của tiến trình. Thời gian xoay vòng trung bình càng thấp thì tiến trình được hoàn thành nhanh chóng hơn.
- Thời gian phản hồi trung bình (Average Response Time): Đây là thời gian trung bình mà một tiến trình phải chờ đợi để nhận được phản hồi ban đầu từ hệ thống sau khi yêu cầu sử dụng CPU. Thời gian phản hồi trung bình đo lường khả năng phục vụ ngay lập tức của hệ thống.
- Hiệu suất CPU (CPU Utilization): Đây là tỷ lệ thời gian thực thi của các tiến trình so với tổng thời gian hoạt động của CPU. Hiệu suất CPU càng cao thì tài nguyên CPU được sử dụng hiệu quả hơn.

Các thông số này giúp đánh giá và so sánh hiệu suất của các thuật toán phân phối tài nguyên CPU khác nhau. Mục tiêu là tìm ra thuật toán tối ưu nhất để đảm bảo hiệu suất và hiệu quả của hệ thống.

2.1.3.6. Đa nhiệm và ưu tiên hệ thống

Khi hệ thống máy tính có nhiều tiến trình chạy đồng thời, việc quản lý và phân phối tài nguyên CPU trở thành một yếu tố quan trọng để đảm bảo hoạt động ổn định và hiệu quả của hệ thống. Trong quá trình đa nhiệm, hệ điều hành cần xác định cách ưu tiên tài nguyên CPU giữa các tiến trình theo một quy tắc nhất định.

Đối với các tiến trình hệ thống quan trọng như quản lý tác vụ, giao tiếp mạng, hoặc tương tác người dùng, việc ưu tiên tài nguyên CPU là cực kỳ quan trọng. Điều này đảm bảo rằng các chức năng hệ thống quan trọng được đảm bảo hoạt động một cách mượt mà và không bị gián đoạn. Các tiến trình này thường được gán mức ưu tiên cao hơn, cho phép chúng chiếm một phần lớn tài nguyên CPU khi cần thiết.

Một ví dụ điển hình là quá trình quản lý tác vụ (task scheduler) của hệ điều hành. Quá trình này có nhiệm vụ quản lý việc chuyển đổi giữa các tiến trình, phân phối tài nguyên và lập lịch CPU. Để đảm bảo hiệu suất cao và sự đáng tin cậy của hệ thống, quá trình quản lý tác vụ thường được ưu tiên cao, cho phép nó truy cập và sử dụng tài nguyên CPU một cách ưu tiên so với các tiến trình khác.

Ngoài ra, các tiến trình có liên quan đến giao tiếp mạng hoặc tương tác người dùng cũng được ưu tiên hơn để đảm bảo truyền thông liên tục và phản hồi nhanh chóng. Ví dụ, các tiến trình xử lý yêu cầu mạng hoặc phản hồi người dùng trực tiếp có thể được ưu tiên cao để đảm bảo việc truyền dữ liệu mạng mượt mà hoặc phản hồi người dùng nhanh chóng.

Mức ưu tiên của mỗi tiến trình có thể được xác định thông qua các thuộc tính như độ ưu tiên (priority) hoặc lớp ưu tiên (priority class). Hệ điều hành sẽ sử dụng thông tin này để quyết định cách phân phối tài nguyên CPU giữa các tiến trình.

Một cách thông thường để quản lý ưu tiên hệ thống là sử dụng một hệ thống lớp ưu tiên, trong đó các tiến trình được phân vào các lớp khác nhau dựa trên mức ưu tiên của chúng. Ví dụ, các tiến trình hệ thống quan trọng có thể thuộc vào lớp ưu tiên cao, trong khi các tiến trình ứng dụng thông thường thuộc vào lớp ưu tiên thấp hơn.

Điều này cho phép hệ thống xử lý các tác vụ quan trọng trước, tránh việc bị ảnh hưởng bởi các tiến trình không quan trọng.

Ngoài ra, hệ điều hành cũng có thể sử dụng các thuật toán lập lịch CPU khác nhau để quyết định thời gian xử lý cho từng tiến trình. Ví dụ, giải thuật SRTF (Shortest Remaining Time First) đã được đề cập trước đó có thể được sử dụng để ưu tiên thực thi các tiến trình có thời gian xử lý còn lại ngắn nhất. Điều này đảm bảo rằng các tiến trình ngắn sẽ hoàn thành nhanh chóng, giải phóng tài nguyên CPU cho các tiến trình khác.

Tuy nhiên, việc xác định và quản lý ưu tiên hệ thống không chỉ dựa trên mức ưu tiên của các tiến trình mà còn phụ thuộc vào các yếu tố khác như mức độ sử dụng tài nguyên hiện tại của mỗi tiến trình, tính khẩn cấp của tác vụ, và các quy tắc lập lịch cụ thể được cấu hình trong hệ điều hành. Tóm lại, việc ưu tiên hệ thống trong đa nhiệm là một yếu tố quan trọng để đảm bảo hoạt động ổn định của hệ thống máy tính.

2.2. Viết chương trình mô phỏng hoạt động của 1 chiến lược lập lịch CPU

2.2.1. Viết chương trình mô phỏng hoạt động của chiến lược lập lịch theo công việc ngắn nhất trước (Shortest-job-first (SJF))

```
1. #include <iostream>
2. #include <queue>
3. #include <algorithm>
4. using namespace std;
5.
6. struct Tientrinh {
7.     int id;    // Tên tiến trình
8.     int tgden; // Thời gian đến
9.     int tgxl;  // Thời gian xử lý
10.    int tght;  // Thời gian hoàn thành
11.    int tgxv;  // Thời gian xoay vòng
12.    int tgph;  // Thời gian phản hồi
13. };
14.
15. void LaplichCPU(queue<Tientrinh>& processes, int sltt) {
16.     int tghientai = 0;
17.     int tongtgcho = 0;
```

```

18.     int totaltgxv = 0;
19.     int totaltgph = 0;
20.     int tongtgchay = 0; // Tổng thời gian chạy của tất c
    à các tiến trình
21.
22.     while (!processes.empty()) {
23.         Tientrinh tthientai = processes.front();
24.         processes.pop();
25.         cout << "Quá trình chạy của tiến trình " << tthi
            entai.id << " từ " << tghientai << " đến " << tghientai +
            tthientai.tgxl << endl;
26.         int waitingTime = tghientai - tthientai.tgden;
27.         tongtgcho += waitingTime;
28.         tghientai += tthientai.tgxl;
29.         tthientai.tgxv = tghientai - tthientai.tgden;
30.         totaltgxv += tthientai.tgxv;
31.         tthientai.tgph = tghientai - tthientai.tgden - t
            thientai.tgxl;
32.         totaltgph += tthientai.tgph;
33.         tongtgchay += tthientai.tgxl;
34.     }
35.
36.     double tgchotb = (tongtgcho * 1.0) / sltt;
37.     double averagetgxv = (totaltgxv * 1.0) / sltt;
38.     double averagetgph = (totaltgph * 1.0) / sltt;
39.     double cpus = ((tongtgchay * 1.0) / tghientai) * 100
        ; // Tỷ lệ sử dụng CPU
40.
41.     cout << "Thời gian chờ trung bình: " << tgchotb << e
        ndl;
42.     cout << "Thời gian xoay vòng trung bình: " << averag
        etgxv << endl;
43.     cout << "Thời gian phản hồi trung bình: " << average
        tgph << endl;
44.     cout << "Hiệu suất CPU: " << cpus << "%" << endl;
45. }
46.
47. int main() {
48.     queue<Tientrinh> tientrinh;
49.     int n; // Số lượng tiến trình
50.     cin >> n;
51.     for (int i = 0; i < n; i++) {
52.         Tientrinh t;
53.         t.id = i + 1; // Tăng id tự động
54.         cin >> t.tgden >> t.tgxl;
55.         tientrinh.push(t);
56.     }

```

```

57.
58.     vector<Tientrinh> sxtientrinh;
59.     while (!tientrinh.empty()) {
60.         sxtientrinh.push_back(tientrinh.front());
61.         tientrinh.pop();
62.     }
63.
64.     // Sắp xếp theo thời gian đến và thời gian xử lý tiế
    n trình
65.
66.     sort(sxtientrinh.begin(), sxtientrinh.end(), [](cons
    t Tientrinh& p1, const Tientrinh& p2) {
67.         if (p1.tgxl != p2.tgxl) {
68.             return p1.tgxl < p2.tgxl;
69.         } else {
70.             return p1.tgden < p2.tgden;
71.         }
72.     });
73.
74.     Tientrinh tmp;
75.     int vt = 0;
76.     int ID = -1;
77.     int tg = 9999;
78.     for (int i = 0; i < sxtientrinh.size(); i++) {
79.         if (sxtientrinh[i].tgden < tg) {
80.             ID = sxtientrinh[i].id;
81.             tg = sxtientrinh[i].tgden;
82.             tmp = sxtientrinh[i];
83.             vt = i;
84.         }
85.     }
86.
87.     // Xóa đi tiến trình chạy đầu tiên trong mảng vector
88.
89.     sxtientrinh.erase(sxtientrinh.begin() + vt);
90.
91.     // Đẩy tiến trình đầu tiên vào hàng đợi để chạy
    tientrinh.push(tmp);
92.
93.     for (const auto& tt : sxtientrinh) {
94.         tientrinh.push(tt);
95.     }
96.
97.     LaplichCPU(tientrinh, tientrinh.size());
98.
99.     return 0;
100.}

```


Đoạn code trên là một chương trình mô phỏng việc lập lịch CPU bằng thuật toán SJF (Shortest Job First).

Giải thích chi tiết cho từng phần của chương trình mô phỏng hoạt động của chiến lược lập lịch SJF

Định nghĩa cấu trúc của một tiến trình bao gồm các thuộc tính sau.

ID: Tên tiến trình

Tgden: Thời gian đến của tiến trình.

Tgxl: Thời gian xử lý cần để hoàn thành tiến trình.

Tght: Thời gian hoàn thành (Thời điểm tiến trình hoàn thành).

Tgxv: Thời gian xoay vòng (Thời gian từ lúc tiến trình đến hàng đợi đến khi hoàn thành).

Tgph: Thời gian phản hồi (thời gian từ lúc tiến trình đến hàng đợi đến khi bắt đầu xử lý).

Hàm LaplichCPU thực hiện việc lập lịch CPU và tính toán các thống kê liên quan.

+ Trong vòng lập, tiến trình hiện tại được lấy ra từ hàng đợi, và thông tin về việc chạy tiến trình (thời gian bắt đầu và kết thúc) được in ra màn hình.

+ Thời gian chờ của tiến trình tính bằng hiệu của thời gian hiện tại và thời gian đến của tiến trình.

+ Sau khi xử lý xong một tiến trình, các thống kê như thời gian xoay vòng và thời gian phản hồi được cập nhật.

```

20 ~ while (!processes.empty()) {
21     Tientrinh tthientai = processes.front();
22     processes.pop();
23     cout << "Quá trình chạy của tiến trình " << tthientai.id << " từ " << tghientai.tgden;
24     int waitingTime = tghientai - tthientai.tgden;
25     tongtgcho += waitingTime;
26     tghientai += tthientai.tgxl;
27     tthientai.tgxv = tghientai - tthientai.tgden;
28     totaltgxv += tthientai.tgxv;
29     tthientai.tgph = tghientai - tthientai.tgden - tthientai.tgxl;
30     totaltgph += tthientai.tgph;
31     tongtgchay += tthientai.tgxl;
32 }
33
34 double tgchotb = (tongtgcho * 1.0) / sltt;
35 double averagetgxv = (totaltgxv * 1.0) / sltt;
36 double averagetgph = (totaltgph * 1.0) / sltt;
37 double cpus = ((tongtgchay * 1.0) / tghientai)*100;

```

Cuối cùng, các giá trị trung bình được tính toán và in ra màn hình.

```

39 cout << "Thời gian chờ trung bình: " << tgchotb << endl;
40 cout << "Thời gian xoay vòng trung bình: " << averagetgxv << endl;
41 cout << "Thời gian phản hồi trung bình: " << averagetgph << endl;
42 cout << "Hiệu suất CPU: " << cpus << "%" << endl;

```

Hàm main.

Trong hàm main, một hàng đợi được khởi tạo.

Chúng ta sẽ nhập thời điểm tiến trình đến, và thời gian mà tiến trình thực hiện. Các tiến trình được thêm vào hàng đợi bằng cách sử dụng phương thức “push” của hàng đợi.

```

queue<Tientrinh> tientrinh;
int n; // Số lượng tiến trình
cin >> n;
for (int i = 0; i < n; i++) {
    Tientrinh t;
    t.id=i+1;
    cin>> t.tgden >> t.tgxl;
    tientrinh.push(t);
}

```

Tiến trình được sắp xếp theo thứ tự tăng dần của thời gian đến và thời gian xử lý bằng cách sử dụng hàm “sort”. Sắp xếp theo thời gian mà tiến trình đến và thời gian để xử lý tiến trình

```

59.     while (!tientrinh.empty()) {
60.         sxtientrinh.push_back(tientrinh.front());
61.         tientrinh.pop();
62.     }
63.
64.     // Sắp xếp theo thời gian đến và thời gian xử lý tiể
n trình
65.
66.     sort(sxtientrinh.begin(), sxtientrinh.end(), [](cons
t Tientrinh& p1, const Tientrinh& p2) {
67.         if (p1.tgxl != p2.tgxl) {
68.             return p1.tgxl < p2.tgxl;
69.         } else {
70.             return p1.tgden < p2.tgden;
71.         }
72.     });
73.

```

Tiến trình đã được sắp xếp lại được sắp xếp lại được đẩy trở lại hàng đợi.

Cuối cùng ,hàm “LaplichCPU” được gọi để lập lịch CPU và tính toán các thống kê. Và xuất ra kết quả bao gồm tên của các tiến trình, thời gian hoàn thành, thời gian chờ trung bình và thời gian xoay vòng trung bình, thời gian phản hồi trung bình.

```

LaplichCPU(tientrinh, tientrinh.size());
return 0;

```

2.2.2. Hướng dẫn sử dụng chương trình

- *Nhập dữ liệu:*

Bước 1: Nhập số lượng tiến trình cần chạy (n tiến trình)

Bước 2: Nhập n cặp số(a và b trong đó a là thời điểm tiến trình đến, b là thời gian mà tiến trình đó thực hiện), tên tiến trình sẽ được đánh số từ 1 đến n.

Bước 3: Thực hiện chương trình (nếu dùng bằng CodePad++ của Upcoder thì người dùng phải chuyển sang ngôn ngữ c++ 14 để chương trình có thể chạy được)

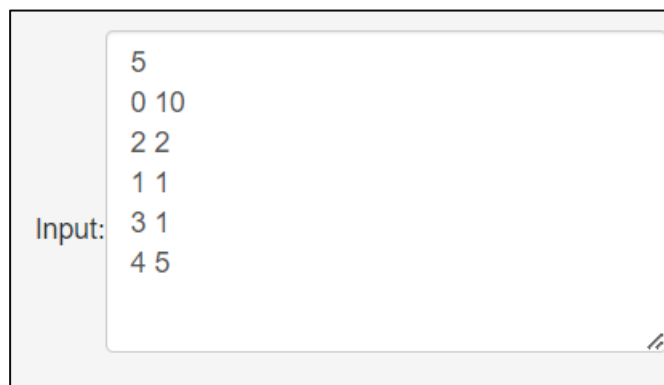
- *Dữ liệu xuất ra:*

Chương trình sẽ xuất ra n dòng đầu là thứ tự thực hiện các tiến trình và khoảng thời gian mà tiến trình thực hiện.

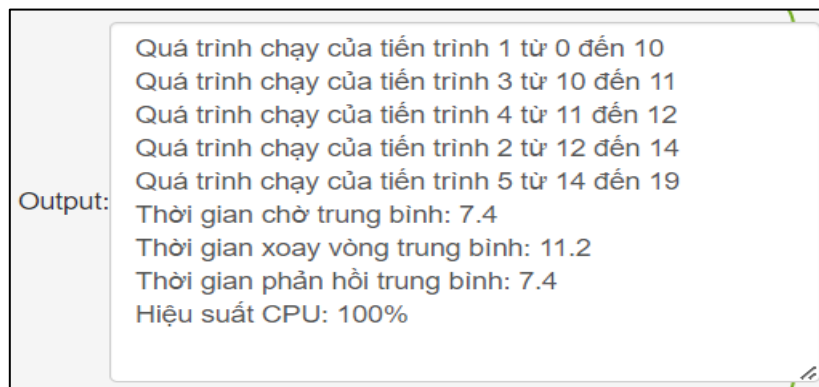
Các dòng tiếp theo chương trình sẽ xuất ra thời gian chờ trung bình, thời gian xoay vòng trung bình, thời gian phản hồi trung bình của các tiến trình và hiệu suất của CPU khi thực hiện các tiến trình.

- *Các test case*

Test case 1:

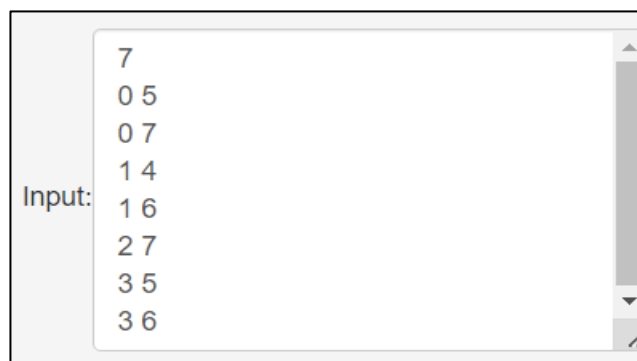


```
5
0 10
2 2
1 1
3 1
4 5
```



```
Quá trình chạy của tiến trình 1 từ 0 đến 10
Quá trình chạy của tiến trình 3 từ 10 đến 11
Quá trình chạy của tiến trình 4 từ 11 đến 12
Quá trình chạy của tiến trình 2 từ 12 đến 14
Quá trình chạy của tiến trình 5 từ 14 đến 19
Thời gian chờ trung bình: 7.4
Thời gian xoay vòng trung bình: 11.2
Thời gian phản hồi trung bình: 7.4
Hiệu suất CPU: 100%
```

Test case 2:



```
7
0 5
0 7
1 4
1 6
2 7
3 5
3 6
```

Output:	Quá trình chạy của tiến trình 1 từ 0 đến 5 Quá trình chạy của tiến trình 3 từ 5 đến 9 Quá trình chạy của tiến trình 6 từ 9 đến 14 Quá trình chạy của tiến trình 4 từ 14 đến 20 Quá trình chạy của tiến trình 7 từ 20 đến 26 Quá trình chạy của tiến trình 2 từ 26 đến 33 Quá trình chạy của tiến trình 5 từ 33 đến 40 Thời gian chờ trung bình: 13.8571 Thời gian xoay vòng trung bình: 19.5714 Thời gian phản hồi trung bình: 13.8571 Hiệu suất CPU: 100%
---------	--

Test case 3:

Input:	6 0 3 0 9 1 2 1 9 1 5 2 7
--------	---

Output:	Quá trình chạy của tiến trình 1 từ 0 đến 3 Quá trình chạy của tiến trình 3 từ 3 đến 5 Quá trình chạy của tiến trình 5 từ 5 đến 10 Quá trình chạy của tiến trình 6 từ 10 đến 17 Quá trình chạy của tiến trình 2 từ 17 đến 26 Quá trình chạy của tiến trình 4 từ 26 đến 35 Thời gian chờ trung bình: 9.33333 Thời gian xoay vòng trung bình: 15.1667 Thời gian phản hồi trung bình: 9.33333 Hiệu suất CPU: 100%
---------	--

Như vậy, nhóm đã trình bày xong chương trình mô phỏng thuật toán FJS để lập lịch CPU bằng ngôn ngữ C++. Dưới đây là đường link đến course code của nhóm:

Link code: <https://github.com/trandoanhkhoa/Hedieuhanh>

TÀI LIỆU THAM KHẢO

Tiếng Việt:

- [1] Nguyễn Trần Hậu, Hệ Điều Hành - Cơ Sở Lý Thuyết và Bài Tập, Đại học Khoa học Tự nhiên, 2014
- [2] Lê Trung Thông, Hệ Điều Hành và Quản Trị Mạng, Đại học Khoa học Tự nhiên, 2016
- [3] Nguyễn Kim Anh, Hệ Điều Hành, Đại học Khoa học Tự nhiên, 2015
- [4] Trần Minh Triết, Hệ Điều Hành - Nguyên Lý và Bài Tập, Nhà xuất bản Khoa học và Kỹ thuật, 2017

Tiếng Anh:

- [5] Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Operating System Concepts, 2018
- [6] Andrew S. Tanenbaum, Modern Operating Systems, 2014

Website:

- [7] <https://quantrimang.com/lang-cong-nghe/lang-lich-su-ve-he-dieu-hanh-windows-cua-microsoft-59098>
- [8] <https://phongvu.vn/cong-nghe/tat-tan-tat-ve-download-windows-10/>
- [9] <https://viblo.asia/p/tien-trinh-trong-he-dieu-hanh-phan-3-3Q75wg6Q5Wb>
- [10] <http://diendan.congdongcviet.com/threads/t24142::cpu-scheduler-bo-lap-lich-cho-cpu-mon-ly-thuyet-hdh.cpp>