

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
VIỆN KINH TẾ



BÁO CÁO BÀI TẬP LỚN
HỆ ĐIỀU HÀNH LINUX VÀ MS-DOS
Bộ Môn: Hệ Điều Hành

Đề tài: Tìm hiểu chung về 2 họ hệ điều hành (OS): MS-Dos và Linux trên máy tính cá nhân PC. So sánh về cấu trúc hệ thống, về các hàm Shell (lời gọi hệ thống), về quản lý bộ nhớ, về quản lý file của 2 hệ điều hành trên và cho biết các đặc điểm của 2 hệ thống.

Thành viên: Trần Minh Hiếu (Trưởng nhóm) - MSV:
Nguyễn Tấn Dũng - MSV: B21DVCN049
Trần Đức An
Đỗ Chí Công - MSV: B21DVCN031
Vũ Ngọc Duy

HÀ NỘI - 2023

Mục lục

Giới thiệu	3
1 Khái quát lịch sử	3
1.1 MS-DOS	3
1.2 Linux	3
2 Thống kê số lượng sử dụng hệ điều hành	4
2.1 Linux	4
2.2 MS-DOS	4
3 Mục đích sử dụng	4
3.1 MS-DOS (Microsoft Disk Operating System)	4
3.2 Linux	4
Nội dung chính	6
1 I. Cấu trúc hệ điều hành	6
1.1 Linux	6
1.2 MS-DOS	8
1.3 So sánh	9
2 Quản lý tiến trình	10
2.1 Quản lý tiến trình (Linux)	10
2.1.1 Tiến trình	10
2.1.2 Điều khiển và giám sát tiến trình	11
2.1.3 Lập kế hoạch các tiến trình	12
2.2 Nguồn tham khảo	13
2.2.1 Tạo và xóa bỏ các quá trình của user và của hệ thống	13
2.2.2 Ngừng, hủy bỏ một quá trình	13
2.2.3 Cơ chế đồng bộ các quá trình	14
3 Các hàm Shell	15
3.1 Linux	15
3.1.1 Shell là gì?	15
3.1.2 Các loại Shells	15
3.1.3 Hàm trong Shell	16
3.2 MS-DOS	18
4 Quản lý bộ nhớ	19
4.1 Linux	19
4.1.1 Phân chương cố định	19
4.1.2 Phân chương động	19
4.1.3 Phân trang	20
4.2 Phân đoạn	21
4.3 Bộ nhớ ảo	22
5 Quản lý File	22
5.1 Linux	22
5.1.1 Các khái niệm	22
5.1.2 Các phương pháp truy cập file	23

5.2	MS-DOS	24
5.2.1	Các khái niệm	24
KẾT LUẬN		26
1	Mục đích sử dụng	26
1.1	Linux	26
1.2	MS-DOS	27
2	SO SÁNH	27
2.1	Linux	27

Giới thiệu

1 Khái quát lịch sử

1.1 MS-DOS

- Xuất Hiện và Phát Triển:
 - MS-DOS, do Microsoft phát triển, xuất hiện vào năm 1981.
 - Được thiết kế cho máy tính cá nhân và máy tính tương thích IBM PC.
- Thời Kỳ Phổ Biến: Trong thập kỷ 1980 và đầu thập kỷ 1990, MS-DOS là hệ điều hành phổ biến nhất trên máy tính cá nhân.
- Giao Diện và Tính Năng:
 - Giao diện dòng lệnh (Command Line Interface - CLI).
 - Hỗ trợ cơ bản cho các ứng dụng và trò chơi.
- Giới Hạn và Chuyển Đổi:
 - Hạn chế về đồ họa và đa nhiệm.
 - Microsoft chuyển hướng sang Windows để cải thiện giao diện đồ họa và khả năng đa nhiệm

1.2 Linux

- Ngày xuất hiện và phát triển
 - Linux được sáng tạo bởi Linus Torvalds vào năm 1991.
 - Mã nguồn mở, cho phép cộng đồng tham gia phát triển.
- Tính Năng và Mục Đích Sử Dụng:
 - Giao diện dòng lệnh và đồ họa, với nền tảng mạnh mẽ.
 - Được sử dụng rộng rãi trên máy chủ, máy tính cá nhân, và thiết bị nhúng.
- Mã Nguồn Mở và Linh Hoạt:
 - Mã nguồn mở giúp thúc đẩy sự đổi mới và sửa lỗi liên tục.
 - Linh hoạt và tùy chỉnh cao, phù hợp với nhiều môi trường và mục đích sử dụng.
- Phổ Biến Ngày Nay:
 - Ngày nay, Linux là một trong những hệ điều hành phổ biến nhất trên thế giới, được sử dụng trong nhiều môi trường từ máy tính cá nhân đến máy chủ và điện toán đám mây.
 - Android: Phiên bản Android chạy trên hạt nhân Linux, làm cho Linux hiện diện mạnh mẽ trong lĩnh vực di động

→ Tổng kết: MS-DOS và Linux đều đóng vai trò quan trọng trong lịch sử hệ điều hành, từ giai đoạn đầu của máy tính cá nhân đến sự phổ biến và ổn định ngày nay. MS-DOS đã mở đường cho sự phổ biến của máy tính cá nhân, trong khi Linux đại diện cho sự mở nguồn và linh hoạt trong thế giới công nghiệp và máy chủ.

2 Thống kê số lượng sử dụng hệ điều hành

2.1 Linux

Linux đã trở thành một trong những hệ điều hành phổ biến nhất trên thế giới, đặc biệt trong các môi trường máy chủ và hệ thống lớn.

Trên máy chủ web, Linux chiếm một tỷ lệ lớn, đặc biệt là trong các trang web lớn và các dịch vụ điện toán đám mây.

Linux được sử dụng rộng rãi trong các thiết bị nhúng, điện thoại thông minh (qua Android), và các thiết bị IoT (Internet of Things).

2.2 MS-DOS

MS-DOS không còn được sử dụng phổ biến trên máy tính cá nhân và máy tính xách tay.

Hệ điều hành này đã chuyển hướng sang Windows từ cuối thập kỷ 1980, và người dùng máy tính cá nhân chủ yếu sử dụng các phiên bản Windows mới hơn. MS-DOS vẫn tồn tại trong một số ứng dụng nhúng và thiết bị cổ điển, nhưng không có thống kê chính xác về số lượng sử dụng.

Vui lòng kiểm tra nguồn tin tức và thống kê hiện tại để có thông tin chi tiết và cập nhật về sự phổ biến của các hệ điều hành này tính đến thời điểm hiện tại.

3 Mục đích sử dụng

3.1 MS-DOS (Microsoft Disk Operating System)

- Máy Tính Cá Nhân:
 - MS-DOS xuất hiện trong giai đoạn đầu của máy tính cá nhân, đặc biệt là trên các máy tính tương thích IBM PC.
 - Dùng cho mục đích văn phòng, xử lý văn bản, và quản lý tệp tin.
- Trò Chơi và Giải Trí: MS-DOS là nền tảng cho nhiều trò chơi kinh điển như Prince of Persia, Doom, và Civilization. Giải trí thông qua giao diện dòng lệnh và trải nghiệm đơn giản.
- Hỗ Trợ Ứng Dụng Doanh Nghiệp: Sử dụng trong môi trường doanh nghiệp để thực hiện các tác vụ quản lý cơ bản

3.2 Linux

- Máy Chủ và Hệ Thống Lớn:
 - Linux là lựa chọn phổ biến cho máy chủ web, máy chủ ứng dụng, và hệ thống lớn do tính ổn định và bảo mật cao.
 - Sử dụng trong các trung tâm dữ liệu và các dịch vụ điện toán đám mây.
- Phát Triển Phần Mềm và Mã Nguồn Mở:

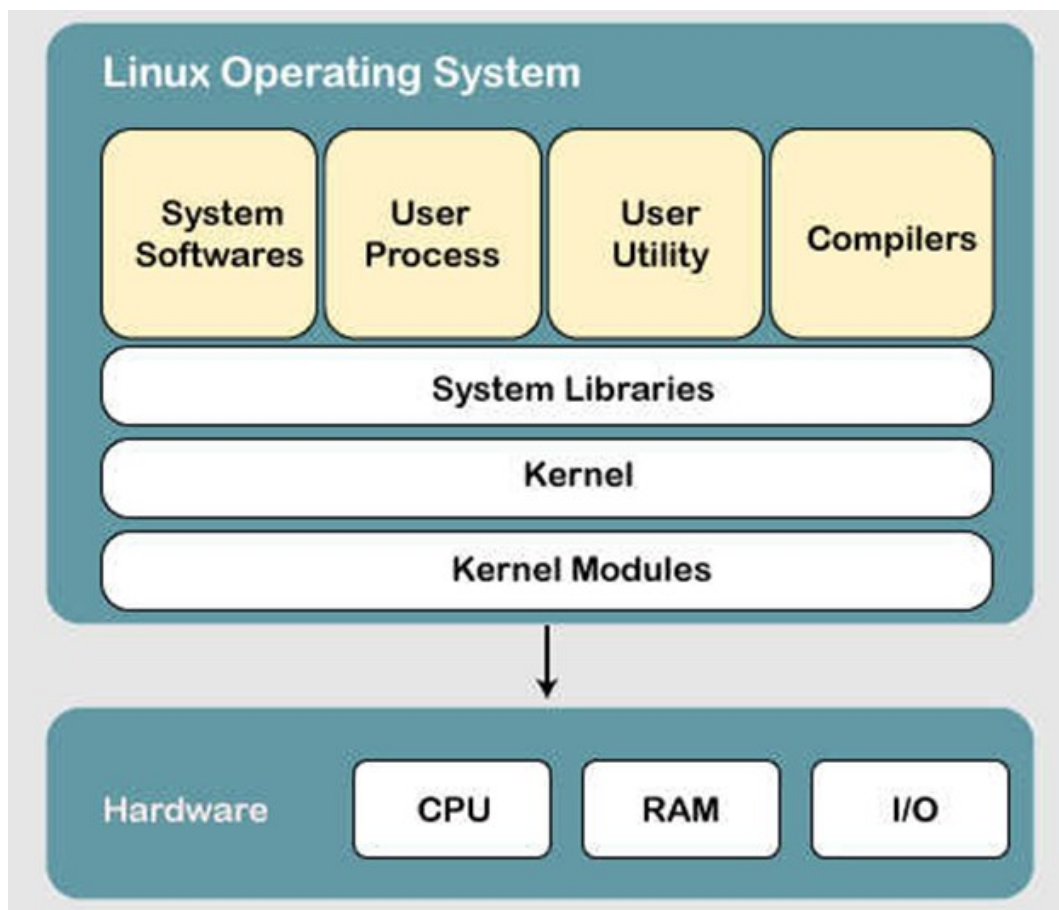
- Linux là một nền tảng phổ biến cho phát triển phần mềm và hỗ trợ mô hình mã nguồn mở.
 - Người phát triển sử dụng Linux để xây dựng các ứng dụng và hệ thống mới.
- Hệ Thống Nhúng và IoT:
 - Linux được tích hợp trong các thiết bị nhúng và hệ thống IoT.
 - Linh hoạt và tùy chỉnh, phù hợp với nhiều loại thiết bị.
- Máy Tính Cá Nhân và Người Dùng Cuối:
 - Ngày nay, các phiên bản Linux dành cho người dùng cá nhân như Ubuntu, Fedora đang ngày càng phổ biến.
 - Cung cấp trải nghiệm đồ họa và chức năng tương tự như các hệ điều hành khác.
- Hệ Thống An Toàn: Sử dụng trong các hệ thống an ninh mạng và các ứng dụng yêu cầu bảo mật cao.

Nội dung chính

1 I. Cấu trúc hệ điều hành

1.1 Linux

Ngoài nắm rõ hệ điều hành Linux là gì chúng ta cũng nên biết về cấu trúc của hệ điều hành này. Linux là hệ điều hành có cấu trúc phân lớp. Các thành phần được kết hợp với nhau và hình thành một hệ thống hoạt động đầy đủ nhất.



Cấu trúc hệ điều hành Linux

Cấu trúc hệ điều hành linux bao gồm:

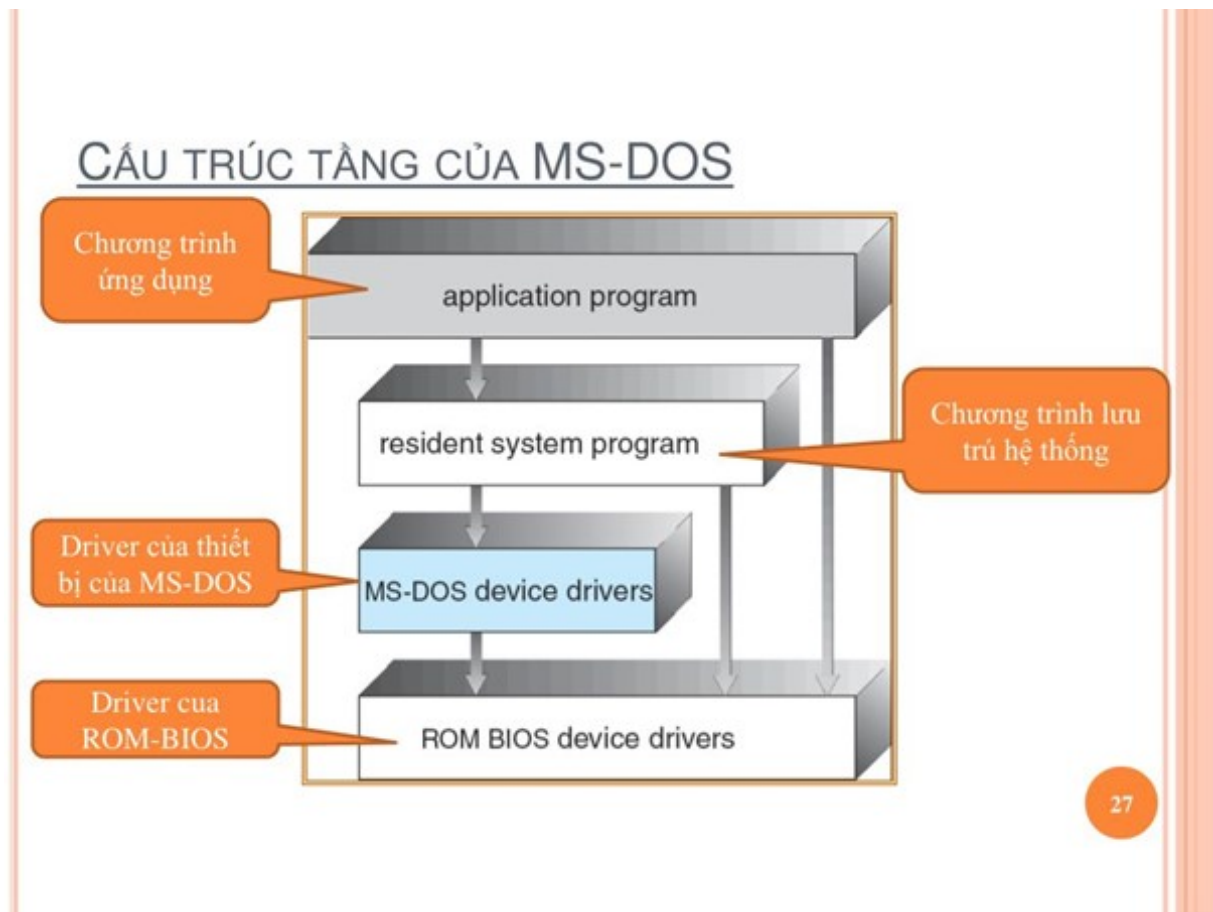
- Kernel (Nhân): Phần quan trọng nhất trong hệ điều hành Linux có vai trò quản lý tài nguyên trong phần cứng như: Bộ vi xử lý, bộ nhớ, thiết bị lưu trữ, thiết bị ngoại vi, định vị. Qua đó các phần mềm có thể truy cập và dùng. Nhân linux gồm 4 chức năng :
 - Quản lý thiết bị: Một hệ thống có nhiều thiết bị được kết nối với nó như CPU, thiết bị nhớ, card âm thanh, card đồ họa,... Một nhân lưu trữ tất cả dữ liệu liên quan đến tất cả các thiết bị trong trình điều khiển thiết bị (nếu không có nhân này sẽ không thể để điều khiển các thiết bị). Do đó kernel biết thiết bị có thể làm gì và thao tác như thế nào để mang lại hiệu suất tốt nhất. Nó cũng quản lý giao tiếp giữa tất cả các thiết bị. Kernel có một số quy tắc nhất định mà tất cả các thiết bị phải tuân theo.
 - Quản lý bộ nhớ: Một chức năng khác mà kernel phải quản lý là quản lý bộ nhớ. Kernel theo dõi bộ nhớ đã sử dụng và chưa sử dụng và đảm bảo rằng các tiến trình không được thao tác dữ liệu của nhau bằng địa chỉ bộ nhớ ảo.

- Quản lý quy trình: Trong quy trình, nhân quản lý chỉ định đủ thời gian và ưu tiên cho các quy trình trước khi xử lý CPU cho các quy trình khác. Nó cũng xử lý thông tin bảo mật và quyền sở hữu.
- Xử lý lệnh gọi hệ thống: Xử lý lệnh gọi hệ thống có nghĩa là một lập trình viên có thể viết một truy vấn hoặc yêu cầu hạt nhân thực hiện một tác vụ
- Thư viện hệ thống(System Libraries)
 - Thư viện hệ thống là các chương trình đặc biệt giúp truy cập các tính năng của hạt nhân. Một hạt nhân phải được kích hoạt để thực hiện một tác vụ và việc kích hoạt này được thực hiện bởi các ứng dụng. Nhưng các ứng dụng phải biết cách đặt lệnh gọi hệ thống vì mỗi hạt nhân có một nhóm lệnh gọi hệ thống khác nhau.
 - Các lập trình viên đã phát triển một thư viện tiêu chuẩn của các thủ tục để giao tiếp với các hạt nhân. Mỗi hệ điều hành hỗ trợ các tiêu chuẩn này, và sau đó các tiêu chuẩn này được chuyển sang các lệnh gọi hệ thống cho hệ điều hành đó.
 - Thư viện hệ thống nổi tiếng nhất dành cho Linux là Glibc(Thư viện GNU C).
- Công cụ hệ thống(System Tools)
 - Hệ điều hành Linux có một tập hợp các công cụ tiện ích, thường là các lệnh đơn giản. Nó là một phần mềm mà dự án GNU đã viết và xuất bản theo giấy phép nguồn mở của họ để phần mềm được cung cấp miễn phí cho tất cả người dùng
 - Với sự trợ giúp của các lệnh, bạn có thể truy cập tệp của mình, chỉnh sửa và thao tác dữ liệu trong thư mục hoặc tệp của bạn, thay đổi vị trí của tệp hoặc bất cứ thứ gì
- Giao diện người dùng
 - Linux có nhiều giao diện người dùng, bao gồm cả CLI (thông qua terminal) và GUI (thông qua các môi trường desktop như GNOME hoặc KDE). Bash là một trong những shell phổ biến nhất trên Linux
 - Shell: Là nơi chứa các dòng lệnh và cung cấp cho người dùng giao diện để nhập các dòng lệnh yêu cầu hệ thống thực hiện. Hiện có khá nhiều loại Shell nhưng bash shell là phổ biến nhất.
- hệ tệp : Linux hỗ trợ nhiều hệ thống tệp như ext4, XFS, Btrfs. Dữ liệu được tổ chức thành các thư mục và tệp, với mỗi thư mục và tệp có một đường dẫn duy nhất từ các thư mục gốc(/).
- Bộ khởi động : Linux sử dụng trình khởi động như GRUB để quản lý quá trình khởi động. Trình khởi động này chịu trách nhiệm cho việc chọn kernel cần khởi động và truyền các tham số cần thiết cho kernel.
- Phần mềm quản lý gói: Hệ thống Linux thường sử dụng quản lý gói như APT(Debian, Ubuntu) hoặc YUM(Red Hat, CentOS) để cài đặt và quản lý phần mềm
- Phân quyền: Linux sử dụng hệ thống phân quyền mạnh mẽ dựa trên nguyên tắc “tất cả mọi thứ là tệp”. Người quản trị có thể quyết định quyền truy cập của người dùng và nhóm đối với các tệp và thư mục

→ Cấu trúc của hệ điều hành Linux rất linh hoạt và có nhiều phần tương tác để cung cấp một môi trường hệ thống mạnh mẽ cho người sử dụng và quản trị viên

1.2 MS-DOS

Hệ điều hành MS-DOS là một hệ điều hành đơn nhiệm và không có khả năng đa người dùng.



Cấu trúc tầng của MS-DOS

Cấu trúc chính của MS-DOS:

- Kernel: MS-DOS sử dụng kernel đơn giản để quản lý tài nguyên hệ thống. Kernel thường được tải lên từ file 'IO.SYS' hoặc 'MSDOS.SYS' khi hệ thống khởi động
- Giao diện người dùng: Giao diện chính của MS-DOS là dòng lệnh (Command-Line Interface-CLI). Người dùng sử dụng bàn phím để nhập các lệnh và tương tác với hệ thống
- Hệ thống tệp(File System): MS-DOS sử dụng hệ thống tệp FAT. Sắp xếp dữ liệu thành các cluster và một bảng FAT giữ thông tin về vị trí của các cluster trên đĩa
- Bộ tải hệ thống : MS-DOS sử dụng chương trình bộ tải thường là 'IBMBIO.COM' hoặc 'IO.SYS' để nạp kernel vào bộ nhớ khi hệ thống khởi động
- Shell: MS-DOS có một shell chính 'COMMAND.COM' để xử lý các lệnh người dùng và quản lý các tác vụ hệ thống cơ bản
- Utilities: để thực hiện các công việc như sao lưu, phân vùng ổ đĩa, định dạng ổ đĩa, ...
- Interrupts và BIOS Calls: để thực hiện các chức năng cơ bản như đọc / ghi vào đĩa, hiển thị ký tự trên màn hình, ... Và để MS-DOS tương tác với phần cứng

→ Cấu trúc MS-DOS khá đơn giản so với các hệ điều hành hiện đại. Chủ yếu tập trung vào việc cung cấp môi trường dòng lệnh đơn giản và quản lý tài nguyên cơ bản của máy tính cá nhân

1.3 So sánh

1. Kiến trúc hạ tầng

- Linux:
 - Hệ điều hành đa người dùng và đa nhiệm
 - Kiến trúc được xây dựng dựa trên UNIX
 - Hỗ trợ nhiều người sử dụng và quá trình chạy đồng thời
- Ms-Dos:
 - Hệ điều hành đơn nhiệm đơn người dùng
 - Thiết kế đơn giản, chỉ chạy một ứng dụng tại một thời điểm

2. Giao diện người dùng

- Linux:
 - Cung cấp giao diện dòng lệnh (CLI) và giao diện đồ họa (GUI)
 - Giao diện đồ họa thường được thực hiện bởi các môi trường như GNOME, KDE
- Ms-Dos:
 - Chủ yếu sử dụng giao diện dòng lệnh Command Prompt

3. Quản lý tệp và thư mục

- Linux:
 - Sử dụng hệ thống quản lý tệp hệ thống phân cấp
 - Mọi thứ là tệp hoặc thư mục, tổ chức trong cây thư mục duy nhất
- Ms-Dos:
 - Sử dụng hệ thống tệp FAT (FAT 16 hoặc FAT 32)
 - Không có khái niệm cây thư mục

4. Định dạng tệp và thư mục

- Linux:
 - Hỗ trợ nhiều định dạng tệp như ext4, xfs, ...
- Ms-Dos:
 - Sử dụng hệ thống tệp FAT (FAT 16 hoặc FAT 32) hoặc NTFS

5. Bảo mật

- Linux:
 - Cung cấp hệ thống bảo mật mạnh mẽ với quyền người dùng cấp cao và cấp thấp
 - Cơ chế kiểm soát quyền truy cập chi tiết
- Ms-Dos:
 - Bảo mật kém hơn so với Linux
 - Hỗ trợ hạn chế về quản lý quyền hạn người dùng

6. Đa nhiệm và ổ đĩa

- Linux:

- Hỗ trợ đa nhiệm và có khả năng làm việc với nhiều loại ổ đĩa và hệ thống tệp khác nhau
- Ms-Dos:
 - Chủ yếu hỗ trợ ổ đĩa FAT và có hạn chế trong việc thực hiện đa nhiệm

7. Hệ thống tệp

- Linux:
 - Sử dụng hệ thống tệp ext4 là phổ biến, nhưng hỗ trợ nhiều hệ thống tệp khác nhau
- Ms-Dos:
 - Sử dụng hệ thống tệp FAT16 hoặc FAT32, và hỗ trợ NTFS từ các phiên bản Windows mới hơn

8. Kernel

- Linux:
 - Mã nguồn mở và có khả năng tùy chỉnh cao
- Ms-Dos:
 - Đóng và có hạn chế tính tùy chỉnh

→ Linux và Ms-Dos có những đặc điểm riêng biệt về kiến trúc hệ thống, giao diện người dùng, quản lý tệp và thư mục, bảo mật, đa nhiệm và ổ đĩa. Sự khác biệt này phản ánh sự tiến triển của công nghệ và mục đích và mục đích sử dụng của từng hệ điều hành

2 Quản lý tiến trình

2.1 Quản lý tiến trình (Linux)

2.1.1 Tiến trình

- Tiến trình tiền cảnh
 - Tiến trình tiền cảnh là một tiến trình đang chạy ở chế độ tiền cảnh, tức là nó đang chiếm quyền sử dụng giao diện của người dùng của hệ điều hành. Điều này nghĩa là một tiến trình tiền cảnh đang chạy, người dùng không thể nhập bất kỳ lệnh nào khác vào hệ điều hành đến khi tiến trình đó hoàn thành.
vd: ls-R/
 - Trong ví dụ, lệnh sẽ liệt kê hết các thư mục và tệp trong thư mục gốc. Lệnh này mất một lúc phải hoàn thành, vì vậy trong thời gian đó, người dùng không thể nhập bất kỳ lệnh nào khác
- Tiến trình hậu cảnh
 - Tiến trình hậu cảnh là một tiến trình đang chạy ở chế độ hậu cảnh, tức là nó đang chiếm quyền sử dụng giao diện người dùng (UI) của hệ điều hành. Điều này có nghĩa là khi một tiến trình hậu cảnh đang chạy, người dùng vẫn có thể nhập bất kỳ lệnh nào khác vào hệ điều hành.
 - Để chuyển một tiến trình từ chế độ tiền cảnh sang chế độ hậu cảnh, bạn có thể sử dụng ký tự & khi bạn khởi động tiến trình.
vd: ls -R / &
 - Lệnh trên đang ở chế độ hậu cảnh. Người dùng vẫn có thể nhập các lệnh vào hệ điều hành.

2.1.2 Điều khiển và giám sát tiến trình

- Chế độ hiện (foreground) và chế độ ngầm (background)
 - Chế độ hiện (foreground): Mặc định các tiến trình thực thi tuần tự, tiến trình này thực hiện xong rồi mới đến tiến trình khác.
vd: \$ emacs
 - Chế độ ngầm (background): Cho phép thực tiến trình cha và tiến trình con chạy song song vd: \$ emacs &
- Sử dụng ps để lấy thông tin trạng thái của tiến trình
Lệnh ps: Hiển thị các tiến trình
vd:

```
$ ps
  PID TTY          TIME CMD
 2803 pts/1    00:00:00 bash
 2965 pts/1    00:00:00 ps
$ ps aux
USER      PID  %CPU  %MEM    VSZ   RSS  TTY  STAT  START  TIME  COMMAND
root         1   0.1   0.1   1104   460  ?    S     15:26  0:03  init[3]
...
ttanh    951   0.0   0.3   1728   996  pts/0  S     16:09  0:00  bash
ttanh    953   0.0   1.9   6860  4916  pts/0  S     16:09  0:00  emacs
ttanh    966   0.0   0.3   2704  1000  pts/0  R     16:23  0:00  ps aux
...
```

- Phát tín hiệu cho một chương trình đang chạy

Sử dụng lệnh kill hủy một tiến trình

Lệnh kill để dừng một tiến trình nào đó

vd: kill PID

Với PID là PID của tiến trình nào đó

Sử dụng lệnh killall hủy một tiến trình

Dùng để hủy tiến trình chạy bằng tên

vd: killall signal_demo.pl

Chạy một tiến trình ở hậu cảnh hoặc tiền cảnh

Khi một tiến trình không sử dụng ký hiệu “&”, tiến trình đó sẽ chạy ở tiền cảnh. Điều này có nghĩa là shell sẽ chặn và đợi tiến trình kết thúc trước khi trả lại quyền điều khiển

Khi thêm “&” vào sau lệnh, tiến trình sẽ chạy ở chế độ hậu cảnh. Shell không chặn và tiếp tục chạy các lệnh khác mà không cần đợi tiến trình kết thúc.

Tạm dừng tiến trình

Khi nhấn Ctrl+Z khi tiến trình đang chạy ở chế độ tiền cảnh. Tiến trình đó được đưa vào hậu cảnh.

Lệnh “jobs” để hiển thị trạng thái của các tiến trình đang chạy ở chế độ hậu cảnh.

vd:

[1] Stopped man ln

```
[2] -Stopped tail
[3] +Stopped ls-R/
```

Dấu '-' và '+' để chỉ ra tiến trình đang chạy ở foreground

Đánh thức tiến trình Để đánh thức một tiến trình nào đó ở hậu cảnh, người dùng sử dụng lệnh "bg"

vd: bg%<job-number>

Để chuyển một tiến trình từ hậu cảnh sang chạy trên tiền cảnh, người dùng sử dụng lệnh 'fg'

vd: fg 3

- Giao tiếp giữa các tiến trình Đôi khi các tiến trình cần trao đổi thông tin cho nhau để xử lý. Một cơ chế được sử dụng khá phổ biến trên Linux là pipe(đường ống). Cơ chế đường ống giữa hai tiến trình cho phép định hướng lại đầu ra của tiến trình thứ nhất trở thành đầu vào của tiến trình thứ hai
Cơ chế đường ống được thiết lập bằng cách sử dụng ký tự: |
\$ cmd1 | cmd2
vd:

```
- $ls -l | more #affiche page par page
- $ls -l | tee log.txt #duplique la sortie
- ps aux | grep conky | grep -v grep | awk
  '{print $2}' | xargs kill
- youtube-dl $1 -q -o - | ffmpeg -i - $2
```

2.1.3 Lập kế hoạch các tiến trình

Sử dụng lệnh at

Lệnh 'at' sử dụng để lên lịch thực hiện các công việc một lần trong tương lai vd: at 20:40

Để xem dung lượng đĩa sử dụng trong toàn bộ file, thư mục của hệ thống sẽ được gọi vào lúc 8:40 p.m

Lệnh 'atrm' sử dụng để hủy một công việc đã lên lịch dựa trên số thứ tự của công việc đó

vd: atrm 1

Câu lệnh trên sẽ hủy công việc có số thứ tự là 1

Sử dụng lệnh crontab

Tiện ích **Cron** cho phép sắp xếp một câu lệnh để thực thi theo một lịch trình định kỳ. Ví dụ. bạn có thể sử dụng cron để xóa các tệp cũ trong /tmp hằng ngày hoặc chạy một tiến trình mỗi ngày hoặc mỗi tuần.

vd:

```
minute hour day month weekday username command
```

* Xóa các tệp cũ trong thư mục /tmp hằng ngày

```
0 0 * * * root find /tmp -mtime +7 -delete
```

Công việc này sẽ chạy vào lúc 0 giờ sáng hằng ngày và xóa tất cả các tệp trong thư mục /tmp có tuổi thọ hơn 7 ngày.

2.2 Nguồn tham khảo

2.2.1 Tạo và xóa bỏ các quá trình của user và của hệ thống

MS-DOS chỉ thị dòng lệnh bằng cách thực hiện trên màn hình. Mỗi lệnh bao gồm một tập và các chỉ thị. Để yêu cầu MS-DOS thi hành một lệnh, người dùng gõ lệnh và nhấn Enter. Nếu người dùng muốn gõ lại một lệnh, hãy nhấn ESC là có thể bắt đầu lại.

MS-DOS có những phím chỉnh để lặp lại hoặc thay đổi một lệnh:

- F1: Hiển thị lệnh trước đó (mỗi lượt ra 3 ký tự)
- F3: Hiển thị toàn bộ lệnh trước đó (một lượt)

Cách thức MS-DOS đáp ứng một lệnh

- MS-DOS sẽ thông báo khi nào lệnh được hoàn tất hay có lỗi khi gõ lệnh
- MS-DOS có sẵn những chỉ dẫn hỗ trợ cho mọi lệnh để tra cứu. Chỉ cần gõ tên lệnh theo sau là '/' hoặc gõ HELP theo sau là một lệnh

2.2.2 Ngừng, hủy bỏ một quá trình

Gọi thực hiện một chương trình

- ***Gọi thực hiện một chương trình***
Có 4 cách gọi thực hiện chương trình
- Nếu một chương trình được liệt kê trong nhóm có mặt trong danh sách thì chọn một chương trình
- Gọi thực hiện 1 chương trình bằng cách chọn tập tin hay một tập tin liên kết với chương trình đó
- Dùng lệnh RUN từ menu file và viết tập tin chương trình
- Làm việc tại dấu nhắc đợi lệnh
- ***Chuyển đổi giữa các chương trình***

Chạy nhiều chương trình

B1: Chạy chương trình đầu tiên bằng cách đúp chuột

B2: Bấm CTRL + ESC để trở lại MS-DOS Shell. Chương trình vừa gọi xuất hiện trên Active Task list

B3: Chạy chương trình 2 bằng cách đúp chuột

Thêm 1 chương trình và Active task list

B1: Chọn tệp tin chương trình muốn thêm và Active Task list

B2: Giữ phím SHIFT, rồi bấm đúp chuột

Chuyển tới chương khác, từ MS-DOS Shell

B1: Bấm đúp chuột chương trình trên Active Task list

B2: Nhấn ENTER

Chuyển từ MS-DOS Shell từ bất cứ chương nào

Nhấn CTRL+ESC

Kết thúc một chương trình

- Để rời bỏ một chương trình
- Chuyển sang MS-DOS Shell
- Từ Active Task list chọn chương trình muốn rời bỏ
- Từ menu file, chọn delete hoặc nhấn DEL

2.2.3 Cơ chế đồng bộ các quá trình

Nếu người dùng có tập tin thường dùng cùng 1 chương trình nào đó. Để tiết kiệm thời gian bằng cách kết các tập tin với chương trình. Mỗi lần mở tập tin được kết, chương trình sẽ khởi động, kèm theo nạp các tập tin đó

- *Kết các tập tin với một chương trình*

B1: Chọn thư mục chứa chương trình muốn kết với một kiểu tập tin

B2: Từ danh sách, chọn tên tệp tin của chương trình ấy

B3: Từ menu file, chọn Associate, hộp thoại Associate file sẽ xuất hiện

B4: Trong hộp Extensions, gõ tên mở rộng của tệp tin muốn kết hợp chương trình

B3: Nhấn OK

- *Chạy một tập tin đã được kết nối với một chương trình khác*

B1: Từ menu file, chọn RUN, hộp thoại Run xuất hiện.

B2: Gõ tên đường dẫn và tên tập tin của chương trình mới

B3: Nhấn OK

- *Gỡ bỏ một liên kết giữa một chương trình và một kiểu tập tin*

B1: Chọn tập tin có liên kết muốn bỏ

B2: Từ menu file, chọn Associate, hộp thoại Associate xuất hiện. Tên chương trình được hiển thị trong hộp thoại văn bản.

B3: Nhấn BACKSPACE để xóa tên chương trình.

B4: Nhấn OK

3 Các hàm Shell

3.1 Linux

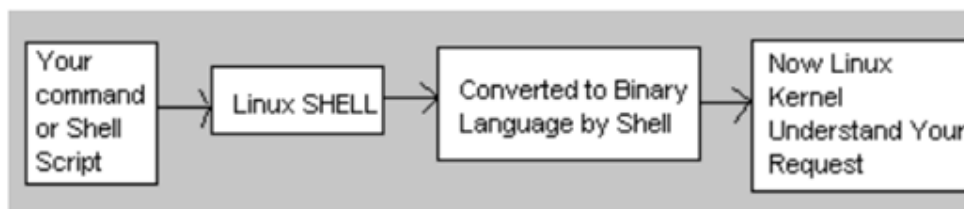
3.1.1 Shell là gì?

Máy tính hiểu ngôn ngữ của 0 và 1 được gọi là ngôn ngữ nhị phân. Trong những ngày đầu của việc tính toán, các chỉ thị được cung cấp bằng ngôn ngữ nhị phân, điều này khó khăn đối với chúng ta để đọc và viết. Do đó, trong hệ điều hành có một chương trình đặc biệt gọi là Shell. Shell chấp nhận các chỉ thị hoặc lệnh của bạn bằng tiếng Anh và dịch chúng thành ngôn ngữ nhị phân của máy tính.

Shell là trình thông dịch lệnh của Linux

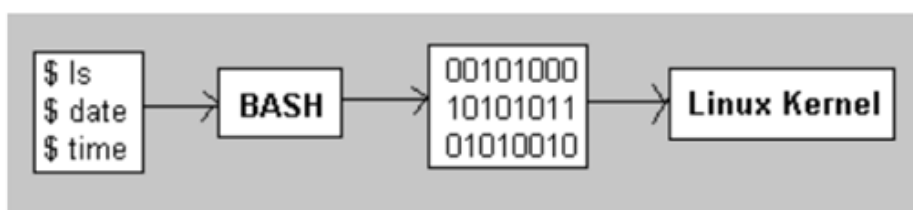
- Thường tương tác với người dùng theo từng câu lệnh
- Shell đọc lệnh từ bàn phím hoặc file
- Nhờ hạt nhân Linux thực hiện lệnh

This is what Shell Does for US



Linux Shell Script Tutorial

You type Your command and shell convert it as



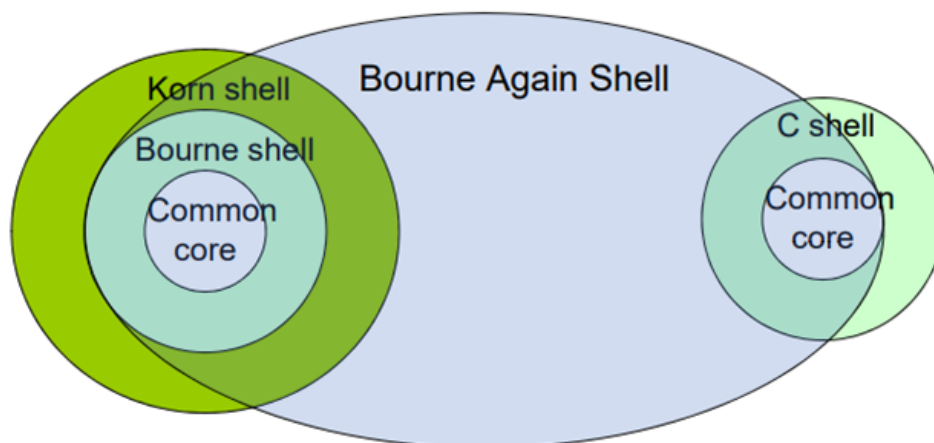
3.1.2 Các loại Shells

Các loại Shells đều có chung các chức năng cơ bản như: multitasking, piping và để sử dụng.

Có 4 Shells phổ biến nhất:

- The Bourne shell (sh)
- The Korn shell (ksh)
- The C shell (csh)
- Bourne Again shell (bash)

Mối liên hệ giữa các loại Shells



Để in ra đường dẫn của shell hiện tại đang được sử dụng: `$ echo $SHELL`

3.1.3 Hàm trong Shell

Cấu trúc hàm:

<pre>ten_ham() { <lệnh 1> <lệnh 2> ... <lệnh n> }</pre>	<p>Ví dụ:</p> <pre>func() { echo "Function foo is executing" } echo "script starting" func echo "script ended" exit 0</pre>
	<pre>script starting Function foo is executing script ending</pre>

Kết quả

Biến cục bộ (có hiệu lực bên trong hàm): Khai báo từ khóa `local`

Biến toàn cục: Khai báo không dùng từ khóa `local`

- Ví dụ sử dụng biến cục bộ và toàn cục:

```
sample text="global variable"
foo() {
    local sample_text="local variable"
    echo "Function foo is executing"
    echo $sample_text
}
echo "script starting"
echo $sample_text
foo
echo "script ended"
echo $sample_text

exit 0
```

script starting
global variable **Kết quả**
Function foo is executing
local variable
script ended
global variable

Các lệnh của shell

Lệnh: (lệnh rỗng) tương đương với true

vd: Kiểm tra nếu file fred tồn tại thì không làm gì cả

```
rm -f fred
if [ -f fred ]; then
:
else
    echo file fred does not exist
fi
exit 0
```

Lệnh exec: Dùng để gọi một lệnh khác và gọi exit khi kết thúc lệnh

vd:

```
echo "Try to execute mc program"
exec mc
echo "you can not see this message !"
```

Lệnh set: Dùng để gán giá trị cho các biến môi trường \$1,\$2,.. bằng các loại bỏ các khoảng trắng không cần thiết và đặt nội dung truyền vào các biến tham số

vd:

```
$ set This is parameter
$ echo $1
This
$ echo $3
parameter
```

Ví dụ

Lệnh set
vd:

```
echo Current date is $(date)
set $(date)
echo The month is $2
echo The year is $6
exit 0
```

Kết quả kết xuất

```
$/set_use.sh
Current date Fri March 13 16:06:16 EST 2001
The month is March
The year is 2001
```

3.2 MS-DOS

Dưới đây là một số hàm shell cơ bản của MS-DOS

Lệnh CD :Sử dụng để thay đổi thư mục làm việc
vd: C:\folder_name

Lệnh DIR :Sử dụng để hiển thị thư mục và tập tin trong thư mục hiện tại
vd: DIR

Lệnh DEL :Xóa các tập tin
vd: DEL file_name

Lệnh REN :Để đổi tên tập tin vd: REN old_file_name new_file_name

Lệnh RD: Sử dụng để xóa thư mục vd: RD folder_name

4 Quản lý bộ nhớ

4.1 Linux

4.1.1 Phân chương cố định

Trong hệ thống Linux, quản lý bộ nhớ có thể được chia thành các khu vực cố định để phục vụ các mục đích cụ thể. Dưới đây là một số khu vực quản lý bộ nhớ cố định trong Linux:

- User Space và Kernel Space: Hệ thống Linux chia bộ nhớ thành hai không gian chính: User Space và Kernel Space. User Space là nơi các ứng dụng và tiến trình người dùng chạy, trong khi Kernel Space là nơi hạt nhân (kernel) chạy.
- Stack: Mỗi tiến trình có một stack riêng. Stack được sử dụng để lưu trữ biến cục bộ và thông tin về các hàm.
- Heap: Đây là khu vực bộ nhớ được cấp phát động cho các biến và cấu trúc dữ liệu khi chương trình đang chạy.
- Text (Code): Khu vực này chứa mã máy của chương trình được thực thi. Nó là bất biến và được chia sẻ giữa các tiến trình để tối ưu hóa bộ nhớ.
- Data (Initialized và Uninitialized): Chứa dữ liệu đã khởi tạo và chưa khởi tạo. Dữ liệu đã khởi tạo chứa giá trị khởi tạo trước khi chương trình chạy, còn dữ liệu chưa khởi tạo có giá trị mặc định.

4.1.2 Phân chương động

Trong hệ thống Linux, quản lý bộ nhớ có thể được phân chia thành các phần chương động để đáp ứng các yêu cầu cụ thể của các tiến trình và ứng dụng. Dưới đây là một số khái niệm và phân chương động quan trọng:

- Heap (Đống):
 - Là một khu vực trong bộ nhớ được sử dụng để cấp phát động bởi các hàm như 'malloc', 'calloc', 'realloc', và 'free'.
 - Chứa dữ liệu không được xác định trước, và các tiến trình có thể cấp phát và giải phóng bộ nhớ từ Heap khi cần.
- Stack (Ngăn Xếp):
 - Dùng để quản lý các biến cục bộ và các hàm gọi.
 - Các biến và giá trị của hàm được lưu trữ trên Stack.
- BSS (Block Started by Symbol):
 - Là một phần của bộ nhớ chứa các biến chưa được khởi tạo (zero-initialized data) trong chương trình.
 - Dữ liệu trong BSS được đặt thành giá trị mặc định (thường là 0).
- Data Segment (DSEG):
 - Chứa các biến đã được khởi tạo và có giá trị cụ thể.
 - Có thể được chia thành một phần có thể chia sẻ (Shared Data Segment) và một phần không thể chia sẻ.

- Text Segment(TSEG)
 - + Chứa mã máy của chương trình và thông thường là duy nhất (read-only).
 - Các hàm và câu lệnh máy được lưu trữ ở đây.
- Memory-Mapped Files:
 - Cho phép các tiến trình truy cập vào các tệp tin dưới dạng bộ nhớ.
 - Sử dụng để tối ưu hóa việc đọc và ghi dữ liệu từ và đến tệp tin
- Memory-Mapped I/O: Cho phép bộ nhớ của chương trình được ánh xạ vào không gian địa chỉ, giúp tiếp cận dữ liệu tương tự như các tệp tin thông thường.
- Cấp Phát Động (Dynamic Allocation):
 - Cấp phát động bằng các hàm như 'malloc', 'calloc, và 'realloc'.
 - Cung cấp linh hoạt trong việc quản lý và sử dụng bộ nhớ.
- Shared Memory: Cho phép nhiều tiến trình chia sẻ một phần của bộ nhớ, giúp họ truy cập và cập nhật dữ liệu chung.
- Virtual Memory:
 - Hệ thống quản lý bộ nhớ ảo, cung cấp ảo hóa bộ nhớ để mỗi tiến trình có cảm giác mình đang sử dụng một phần không gian bộ nhớ riêng biệt.
 - Các phần trên đây thường được quản lý và đổi bởi kernel của hệ thống, và sự hiểu biết về cách bộ nhớ được tổ chức giúp người phát triển và quản trị hệ thống tối ưu hóa hiệu suất và tài nguyên.

4.1.3 Phân trang

Phân trang là một kỹ thuật quản lý bộ nhớ hiệu suất cao được sử dụng trong hệ thống Linux. Nó cho phép hệ điều hành chia bộ nhớ vật lý thành các trang cố định (thường có kích thước là 4 KB) và quản lý trang này trên đĩa cứng. Dưới đây là mô tả cơ bản về cách phân trang hoạt động trong Linux:

- Trang Vật Lý: Bộ nhớ vật lý chia thành các trang cố định với kích thước thường là 4 KB (có thể thay đổi tùy thuộc vào kiến trúc hệ thống).
- Trang Ẩn và Trang Hiện:
 - Mỗi trang có thể ở trạng thái "ẩn" (chưa được đặt vào bộ nhớ vật lý) hoặc "hiện" (đã được đặt vào bộ nhớ vật lý).
 - Khi một tiến trình cần truy cập dữ liệu trên một trang chưa được đặt vào bộ nhớ, hệ điều hành sẽ thực hiện một trang ẩn.
- Trang An (Page Fault):
 - Khi một trang chưa được đặt vào bộ nhớ vật lý (trang ẩn), xảy ra một trang ẩn (page fault).
 - Hệ điều hành sẽ phải đọc trang từ đĩa cứng và đặt nó vào bộ nhớ vật lý.
- Trang Thay Thế (Page Replacement):

- Khi bộ nhớ vật lý đã đầy, hệ điều hành c↓ hay thế một trang hiện tại để có chỗ cho một trang mới.
- Có nhiều thuật toán thay thế trang như LRU (Least Recently Used), FIFO (First-In-First-Out), và NRU (Not Recently Used).
- Swap Space:
 - Swap space là một khu vực trên đĩa cứng được sử dụng để lưu trữ các trang khi chúng không còn chỗ trong bộ nhớ vật lý.
 - Khi một trang cần được thay thế, nó có thể được đẩy ra swap space và trang mới được đọc vào bộ nhớ.
- Mô Hình Vùng Trang: Các trang thường được quản lý theo mô hình vùng trang (page region) để tối ưu hiệu suất và quản lý.
- Hệ Thống Tập Trung (Demand Paging): Hệ thống Linux sử dụng mô hình demand paging, nghĩa là chỉ đọc và đặt trang vào bộ nhớ khi chúng thực sự cần thiết.
- Thống Kê Hiệu Suất:
 - Hệ thống Linux cung cấp các công cụ như 'vmstat', 'free', và 'sar' để theo dõi và đánh giá hiệu suất phân trang và sử dụng bộ nhớ.
 - Phân trang là một cơ sở quan trọng của quản lý bộ nhớ trong Linux và nhiều hệ điều hành khác, giúp tối ưu hiệu suất của hệ thống và quản lý tài nguyên một cách hiệu quả.

4.2 Phân đoạn

Trong hệ điều hành Linux, quản lý bộ nhớ được thực hiện thông qua việc chia bộ nhớ thành các phân đoạn khác nhau. Các phân đoạn này được sử dụng để lưu trữ các loại dữ liệu và có các đặc tính và mục đích khác nhau. Dưới đây là một số phân đoạn quan trọng:

- Phân đoạn hệ thống (System Segment)
 - Chứa mã lệnh hệ điều hành và dữ liệu quan trọng.
 - Thường tải vào bộ nhớ từ 512K đến 640K, tùy thuộc vào cấu hình cụ thể của hệ thống.
- Bảng vectơ ngắt (Interrupt Vector Table - IVT)
 - Nằm ở địa chỉ bắt đầu của bộ nhớ (0-4K).
 - Chứa địa chỉ của các dịch vụ ngắt (interrupt services) của hệ điều hành.
- DOS Data Area (DDA)
 - Chứa các biến và dữ liệu hệ thống của DOS.
 - Nằm ở địa chỉ 4K đến 5K trong bộ nhớ.
- Bộ nhớ dự trữ cho hạt chương trình (Transient Program Area - TPA)
 - Chứa chương trình và dữ liệu ứng dụng.
 - Bắt đầu từ địa chỉ 1MB và có thể mở rộng xuống dưới tận 640K, tùy thuộc vào cấu hình.

- Upper Memory Area (UMA)
 - Vùng bộ nhớ từ 640K đến 1MB.
 - Đôi khi được sử dụng cho các phần mềm mở rộng hoặc các trình quản lý bộ nhớ của bên thứ ba.
- Conventional Memory
 - Vùng từ 0 đến 640K được chia thành các khối nhỏ để chứa các chương trình và dữ liệu.
 - Các chương trình ứng dụng thường chạy trong phạm vi này.

→ Quản lý bộ nhớ trong MS-DOS thường gặp hạn chế do giới hạn 640K cho ứng dụng và sự phức tạp khi sử dụng bộ nhớ cao. Các công cụ như Himem.sys, EMM386.exe và các tiện ích của bên thứ ba thường được sử dụng để mở rộng khả năng quản lý bộ nhớ của MS-DOS.

4.3 Bộ nhớ ảo

Hệ điều hành MS-DOS không có hỗ trợ bộ nhớ ảo như các hệ điều hành hiện đại. Khái niệm về bộ nhớ ảo thường đi kèm với khả năng tạo ra một không gian bộ nhớ ảo lớn hơn so với bộ nhớ vật lý có sẵn trên hệ thống. Hệ điều hành MS-DOS thường xuyên phải đối mặt với giới hạn bộ nhớ 640K và không cung cấp chức năng bộ nhớ ảo như các hệ điều hành như Windows, Linux hoặc macOS. Tuy nhiên, có một số tiện ích và trình quản lý bộ nhớ của bên thứ ba được phát triển để mở rộng khả năng quản lý bộ nhớ của MS-DOS. Một số trong số đó có thể giúp tạo ra một ảo trang (virtual page) giả mạo, nhưng nó không phải là bộ nhớ ảo như ta hiểu trong ngữ cảnh của các hệ điều hành hiện đại.

Các công cụ và tiện ích này bao gồm:

- Himem.sys và EMM386.exe:
 - Himem.sys là một trình điều khiển chính trong MS-DOS để hỗ trợ quản lý bộ nhớ cao (Upper Memory).
 - EMM386.exe là một trình quản lý bộ nhớ mở rộng, cung cấp khả năng sử dụng bộ nhớ chia sẻ và mở rộng.
- Virtual Memory Manager (VMM):
 - VMM của Quarterdeck là một sản phẩm từ bên thứ ba cung cấp khả năng giả mạo bộ nhớ ảo trong môi trường MS-DOS.
 - Tuy nhiên, đối với một trải nghiệm bộ nhớ ảo đầy đủ, người dùng thường phải chuyển sang các hệ điều hành được thiết kế với sự hỗ trợ bộ nhớ ảo, chẳng hạn như Windows, Linux hoặc macOS.

5 Quản lý File

5.1 Linux

5.1.1 Các khái niệm

Trong hệ điều hành Linux, mọi thứ, kể cả thiết bị và tệp tin, được biểu diễn như một tệp tin. Dưới đây là một số khái niệm quan trọng về tệp tin trong Linux:

1. Tập Tin:

File Types:

. Trong Linux, có nhiều loại tập tin, bao gồm:

Regular Files: Chứa dữ liệu văn bản hoặc nhị phân.

Directories (Thư mục): Chứa danh sách các tập tin và thư mục khác.

Symbolic Links (Liên kết tượng trưng): Một loại liên kết trỏ đến một tập tin hoặc thư mục khác.

. Devices (Thiết bị): Đại diện cho các thiết bị phần cứng.

. Special Files: Đại diện cho các thiết bị như /dev/null, /dev/random, v.v.

File Permissions:

Mỗi tập tin và thư mục có ba nhóm quyền: quyền đọc (r), quyền ghi (w), và quyền thực thi (x), được xác định cho chủ sở hữu, nhóm và người dùng khác.

Path: Đường dẫn (path) của một tập tin là địa hoặc vị trí của nó trong hệ thống tập tin.

Đường dẫn tuyệt đối bắt đầu từ thư mục gốc (/) và đường dẫn tương đối bắt đầu từ thư mục hiện tại.

2. Cấu Trúc Thư Mục và Tập Tin: Hệ Thống Tập Tin Linux (File System Hierarchy):

Thư mục gốc: /

. Thư mục chứa chương trình thực thi: '/bin' Thư mục cấu hình: '/etc'

Thư mục chứa thư mục home cá nhân của người dùng: '/home'

Thư mục chứa các thư viện: '/lib' và '/lib64'. Thư mục chứa mã nguồn hệ thống: '/usr/src'

Thư mục chứa các tập tin lưu trữ tạm thời: '/tmp'

Thư mục chứa các ứng dụng cài đặt của hệ thống: '/sbin'

Thư mục chứa tập tin hệ thống: '/boot'

Thư mục chứa các ứng dụng cài đặt bổ sung: '/opt'

Thư mục home của người dùng: '/home/username'

Dấu chấm (.) và Dấu hai chấm (..):

... đại diện cho thư mục hiện tại.

*... đại diện cho thư mục cha.

3. Mở Rộng Tên Tập Tin và Dấu Sao (*):

Một số ký tự đặc biệt như dấu sao (*) có thể được sử dụng để khớp nhiều tập tin hoặc thư mục.

4. Lệnh Tập Tin và Thư Mục:

'ls': Liệt kê nội dung thư mục.

'cp': Sao chép tập tin hoặc thư mục.

mv: Di chuyển hoặc đổi tên tập tin và thư mục.

rm*: Xóa tập tin hoặc thư mục.

mkdir': Tạo thư mục mới.

chmod': Thay đổi quyền truy cập của tập tin hoặc thư mục.

Cấu trúc và quản lý tập tin trong Linux có tính linh hoạt cao và đa dạng, cho phép người dùng tận dụng nhiều tính năng và công cụ mạnh mẽ của hệ điều hành này.

5.1.2 Các phương pháp truy cập file

Dưới đây là một số phương pháp để truy cập tập tin trong hệ điều hành Linux:

1. Dòng Lệnh: Sử dụng các lệnh dòng lệnh như 'ls', 'cd', 'cp', 'mv', 'rm', 'mkdir' để liệt kê, di chuyển, sao chép, đổi tên, xóa và tạo thư mục.

2. Trình Quản Lý Tệp Tin Đồ Họa: Sử dụng các trình quản lý tệp tin đồ họa như Nautilus (đôi với GNOME), Dolphin (đôi với KDE), hoặc Thunar (đối với XFCE).
3. Ứng Dụng Soạn Thảo Văn Bản: Mở và chỉnh sửa tệp tin văn bản bằng các trình soạn thảo như Vim, Nano, Emacs hoặc các trình soạn thảo đồ họa như Gedit.
4. Ứng Dụng Xem Hình Ảnh hoặc Video: Sử dụng các ứng dụng xem hình ảnh hoặc video để mở tệp tin đa phương tiện.
5. Ứng Dụng Đọc Tài Liệu: Mở tệp tin văn bản hoặc tài liệu bằng các ứng dụng đọc tài liệu như Evince, Okular, hoặc Adobe Reader.
6. Kết Hợp với Lệnh Find: Sử dụng lệnh 'find' để tìm kiếm tệp tin trong hệ thống.
7. Thao Tác Trực Tuyến:
8. Sử Dụng Đường Dẫn URL: Sử dụng các dịch vụ lưu trữ trực tuyến như Dropbox, Google Drive, hoặc OneDrive để truy cập và quản lý tệp tin từ xa.
9. Quản Lý Tệp Tin qua SSH: Sử dụng giao thức như ftp://, 'ssh://', hoặc smb:// để truy cập tệp tin từ xa.
10. Ứng Dụng Chỉnh Sửa Hình Ảnh hoặc Âm Thanh: Sử dụng các ứng dụng chỉnh sửa hình ảnh hoặc âm thanh để làm việc với các loại tệp tin đặc biệt.
11. Sử Dụng File Manager Trong Terminal: Sử dụng lệnh 'xdg-open' hoặc 'gnome-open' để mở tệp tin hoặc thư mục bằng cửa sổ mặc định trong môi trường đồ họa.
12. Sử Dụng Đường Dẫn URL: Sử dụng các giao thức như ftp: //, 'ssh://', hoặc smb: // để truy cập tệp tin từ xa.

Các phương pháp này cung cấp sự linh hoạt cho người dùng Linux để truy cập và quản lý tệp tin theo nhiều cách khác nhau, dựa vào nhu cầu và ưu tiên của họ.

5.2 MS-DOS

5.2.1 Các khái niệm

MS-DOS (Microsoft Disk Operating System) là một hệ điều hành đơn nhiệm được phát triển bởi Microsoft trong thập kỷ 1980 và sử dụng rộng rãi trên các máy tính cá nhân trong giai đoạn đầu của lịch sử máy tính cá nhân. Dưới đây là một số khái niệm và đặc điểm cơ bản về file và cấu trúc file trong MS-DOS:

1. Khái Niệm File: . Trong MS-DOS, file là một tập tin lưu trữ thông tin hoặc dữ liệu. Mỗi file được xác định bằng một tên và có thể có một phần mở rộng (extension) để chỉ định loại dữ liệu. Ví dụ: document.txt".
2. Cấu Trúc File: MS-DOS sử dụng một hệ thống tệp FAT (File Allocation Table) để quản lý cấu trúc lưu trữ file trên đĩa. Hệ thống tệp FAT chia đĩa thành các cluster, và thông tin về file được lưu trữ trong các entry của bảng FAT.
3. . Tên File: Tên file trong MS-DOS được giới hạn đến 8 ký tự (8.3 format), với phần mở rộng có thể chứa tối đa 3 ký tự. Ví dụ: 'filename.ext'.

4. . Thư Mục (Directory): • Thư mục trong MS-DOS là một cách để tổ chức và quản lý các file. Thư mục có thể chứa nhiều file và thư mục con. Tên của thư mục cũng phải tuân theo quy tắc 8.3.
5. Đường Dẫn (Path): Đường dẫn trong MS-DOS là cách mô tả vị trí của một file hoặc thư mục trong hệ thống tệp. Ví dụ: 'C:\Folder\file.txt'.
6. Lệnh Duyệt File (DIR): * Lệnh 'DIR' được sử dụng để liệt kê các file và thư mục trong thư mục hiện tại hoặc thư mục được chỉ định.
7. Lệnh Tạo Thư Mục (MD) và Xóa Thư Mục (RD): * Lệnh 'MD' dùng để tạo một thư mục mới, và lệnh 'RD' dùng để xóa một thư mục.
8. . Lệnh Copy và Move: Lệnh COPY được sử dụng để sao chép file, trong khi lệnh 'MOVE' được sử dụng để di chuyển file.
9. Lệnh Ren (Rename): * Lệnh 'REN' dùng để đổi tên file hoặc thư mục.
10. Lệnh Del (Delete): Lệnh 'DEL' dùng để xóa file.

Cấu trúc file và các lệnh trong MS-DOS thường giới hạn và đơn giản so với các hệ điều hành hiện đại, nhưng chúng là những yếu tố cơ bản mà người sử dụng MS-DOS phải làm quen khi làm việc với hệ thống tệp của nó.

KẾT LUẬN

1 Mục đích sử dụng

1.1 Linux

Hệ điều hành Linux được phát triển để phục vụ nhiều mục đích khác nhau và có thể được tùy chỉnh linh hoạt để đáp ứng nhu cầu cụ thể của người sử dụng. Dưới đây là một số phiên bản Linux phổ biến và mục đích chính mà chúng thường phù hợp:

1. Ubuntu:

Mục đích chung sử dụng máy tính cá nhân và văn phòng.

Dễ sử dụng và cài đặt, thích hợp cho người mới sử dụng Linux.

Có phiên bản dành cho máy chủ (Ubuntu Server) và máy tính xách tay (Ubuntu Desktop).

2. Debian:

Stabil và độ tin cậy cao.

Thích hợp cho các máy chủ, đặc biệt là các máy chủ web.

3. Fedora:

Sử dụng cho môi trường làm việc máy tính cá nhân.

Cập nhật thường xuyên, chú trọng vào tính năng mới.

4. CentOS:

Phiên bản tái tạo từ mã nguồn mở của Red Hat Enterprise Linux (RHEL).

Thích hợp cho môi trường doanh nghiệp và máy chủ.

5. Arch Linux:

Hướng đến người dùng có kinh nghiệm và muốn tùy chỉnh hệ thống của mình một cách chi tiết.

Sử dụng hệ thống quản lý gói Pacman.

6. Kali Linux:

Được thiết kế cho penetration testing và bảo mật mạng.

Chứa nhiều công cụ dành cho chuyên gia bảo mật.

7. Raspberry Pi OS (Raspbian):

Dành cho Raspberry Pi, một máy tính nhỏ và giá rẻ.

Tối ưu hóa cho việc chạy trên các thiết bị có tài nguyên hạn chế.

8. Slackware:

Mang lại trải nghiệm gần với "nguyên trạng" của Linux.

Thích hợp cho người dùng muốn hiểu rõ cách hệ thống Linux hoạt động.

Mỗi phiên bản Linux có những đặc điểm riêng biệt và được tối ưu hóa cho một mục đích cụ thể.

Việc chọn lựa phiên bản phụ thuộc vào nhu cầu và kinh nghiệm của người sử dụng.

1.2 MS-DOS

MS-DOS (Microsoft Disk Operating System) là một hệ điều hành dòng lệnh được Microsoft phát triển. MS-DOS thường được sử dụng trên các máy tính cá nhân trong thập kỷ 1980 và đầu thập kỷ 1990, và nó đã đặt nền tảng cho các hệ điều hành Windows sau này. Dưới đây là một số mục đích chính mà MS-DOS đã phù hợp trong quá khứ:

1. Chạy ứng dụng DOS:
MS-DOS chủ yếu được thiết kế để chạy các ứng dụng và trò chơi dòng lệnh được phát triển cho môi trường DOS. Nó hỗ trợ việc thực thi các chương trình thông qua các tập lệnh và tập tin thực thi có đuôi ".exe".
2. Quản lý tệp tin và thư mục:
MS-DOS cung cấp các lệnh cơ bản để quản lý tệp tin và thư mục trên hệ thống tệp tin FAT (File Allocation Table). Người dùng có thể di chuyển, xóa, tạo và đổi tên tệp tin và thư mục.
3. Chạy trò chơi và ứng dụng giáo dục:
Nhiều trò chơi và ứng dụng giáo dục được phát triển để chạy trên MS-DOS. Trong thời kỳ này, nó thường là môi trường chính để trò chơi và các ứng dụng giáo dục.
4. Khởi động máy tính nhanh chóng:
MS-DOS là một hệ điều hành nhẹ và đơn giản, giúp máy tính khởi động nhanh chóng và sử dụng ít tài nguyên so với các hệ điều hành hiện đại.

Tuy nhiên, do sự giới hạn của giao diện dòng lệnh và khả năng quản lý hệ thống so với các hệ điều hành đồ họa hiện đại, MS-DOS đã bị thay thế bởi các hệ điều hành như Windows, Linux và macOS. Ngày nay, MS-DOS không còn phù hợp cho nhiều mục đích do hạn chế về tính năng và khả năng tương thích với các ứng dụng và phần cứng mới.

2 SO SÁNH

2.1 Linux

1. Đa Mục Đích và Linh Hoạt
Cả hai đều linh hoạt và có thể sử dụng cho nhiều mục đích khác nhau, từ máy chủ đến máy tính cá nhân.
2. Sự Phổ Biến và Cộng Đồng Hỗ Trợ
Cả Linux và mục đích sử dụng của nó đều được hỗ trợ mạnh mẽ bởi một cộng đồng lớn và tích hợp nhiều giải pháp và ứng dụng từ cộng đồng.
3. Tính Mở Nguồn:
Linux là hệ điều hành mã nguồn mở, có nghĩa là mã nguồn của nó là công khai và có thể được sửa đổi và phân phối lại.

Tóm lại, Linux là một hệ điều hành đa mục đích, linh hoạt và có tính mở nguồn, phù hợp cho nhiều ngữ cảnh từ phát triển ứng dụng đến triển khai máy chủ và máy tính cá nhân. Sự phù hợp của Linux phản ánh khả năng của nó để đáp ứng đa dạng nhu cầu của người sử dụng.

1. Phong Cách Giao Diện:

Linux có giao diện dòng lệnh và có thể có giao diện đồ họa (GUI). Ngược lại, MS-DOS chỉ có giao diện dòng lệnh, không có giao diện đồ họa mặc định.

2. Phạm Vi Ứng Dụng và Linh Hoạt:

Linux đã phát triển thành một hệ điều hành đa nhiệm, linh hoạt và có thể phục vụ nhiều mục đích khác nhau. MS-DOS chủ yếu được thiết kế để chạy các ứng dụng DOS và không có tính linh hoạt lớn như Linux.

3. Tính Mở Nguồn:

. Linux là một hệ điều hành mã nguồn mở, trong khi MS-DOS không có tính mở nguồn.

4. Tiến Bộ Công Nghệ:

Linux đã tiếp tục phát triển và thích ứng với các tiến bộ công nghệ mới, trong khi MS-DOS đã dần dần bị thay thế bởi các hệ điều hành Windows và không còn được phát triển tích cực.

Tóm lại, MS-DOS đã đóng vai trò quan trọng trong lịch sử máy tính cá nhân, đặc biệt trong giai đoạn đầu, nhưng hiện tại đã lạc lõng với sự phát triển của các hệ điều hành hiện đại như Linux. Linux, với tính linh hoạt và khả năng đáp ứng nhiều mục đích, tiếp tục được ưa chuộng và phát triển.