**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**





**BÁO CÁO ĐỒ ÁN CUỐI HỌC PHẦN**

**MÔN: PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ**

**TÊN ĐỀ TÀI:**

**TÌM HIỂU HỆ ĐIỀU HÀNH CENTOS**

**Họ tên thành viên trong nhóm**

**PHAN ĐỒNG THIÊN LONG – 3118410242**

**THIỀU THỊ THỦY – 3118412058**

**LÊ THỊ QUỲNH HƯƠNG – 3118410174**

**NGUYỄN NGỌC THÁI – 3118410390**

**GIÁO VIÊN HƯỚNG DẪN: PGS.TS. Vũ Thanh Nguyên**

**Năm học 2021 – 2022. Học kỳ 1.**

**TP. HCM tháng 12/2021**

# MỤC LỤC

[MỤC LỤC 2](#_Toc20097)

[CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH LINUX 4](#_Toc21597)

[1. Hệ điều hành Linux: 4](#_Toc30417)

[1.1. Lịch sử phát triển: 4](#_Toc2955)

[1.2. Những ưu điểm của Linux: 5](#_Toc20617)

[1.2.1. Khả năng tương thích với các hệ mở 5](#_Toc20927)

[1.2.2. Hỗ trợ ứng dụng 5](#_Toc4221)

[1.2.3. Lợi ích cho giới chuyên nghiệp điện toán 5](#_Toc13396)

[1.3. Khuyết điểm của Linux : 5](#_Toc29455)

[1.3.1. Hỗ trợ kỹ thuật 5](#_Toc6347)

[1.3.2. Phần cứng 5](#_Toc8125)

[1.4. Kiến trúc của hệ điều hành Linux: 6](#_Toc2219)

[1.4.1. Hạt nhân (Kernel) 6](#_Toc25090)

[1.4.2. Shell 6](#_Toc12615)

[1.4.3. Các tiện ích 7](#_Toc10892)

[1.4.4. Chương trình ứng dụng 7](#_Toc24241)

[1.5. Các đặc tính cơ bản của Linux: 7](#_Toc8914)

[1.5.1. Đa tiến trình 7](#_Toc13169)

[1.5.2. Tốc độ cao 7](#_Toc11907)

[1.5.3. Bộ nhớ ảo 7](#_Toc10926)

[1.5.4. Sử dụng chung thư viện 7](#_Toc20855)

[1.5.5. Sử dụng các chương trình xử lý văn bản 7](#_Toc8432)

[1.5.6. Sử dụng giao diện cửa sổ 8](#_Toc26107)

[1.5.7. Network Information Service (NIS) 8](#_Toc16294)

[1.5.8. Lập lịch hoạt động chương trình, ứng dụng 8](#_Toc11956)

[1.5.9. Các tiện ích sao lưu dữ liệu 8](#_Toc18955)

[1.5.10. Hỗ trợ nhiều ngôn ngữ lập trình. 8](#_Toc26738)

[2. CentOS 9](#_Toc32301)

[1.6.Giới thiệu HĐH CentOS: 9](#_Toc25501)

[1.7.Lịch sử hình thành và phát triển của CentOS: 9](#_Toc4889)

[1.8.Các tính năng chính của CentOS 10](#_Toc10800)

[1.9.Kiến trúc của CentOS 12](#_Toc25979)

[1.10.Ưa điểm và nhược điểm của CentOS: 12](#_Toc7767)

[1.11.Giấy phép của CentOS 13](#_Toc11832)

[1.12 Các tiện ích của Linux- CentOs 15](#_Toc7977)

[1.12.1.Trình soạn văn bản 15](#_Toc15043)

[1.12.1.1.Gedit 15](#_Toc27679)

[1.12.1.2. Emacs 16](#_Toc7490)

[1.12.1.3.Vim/Vi 17](#_Toc1318)

[1.12.2. Tiện ích Setup 19](#_Toc15041)

[1.12.3. Tiện ích Fdisk 19](#_Toc29837)

[1.12.4. Theo dõi thông tin mạng 20](#_Toc4093)

[1.12.5. Tiện ích Lynx 21](#_Toc6432)

[1.12.6. Phân tích đĩa 21](#_Toc2451)

[1.12.7. Theo dõi hệ thống 22](#_Toc31258)

[1.12.8. Quản lý log 23](#_Toc29330)

[1.12.9. Tiện ích mail 23](#_Toc15340)

[CHƯƠNG 2: LẬP TRÌNH SHELL VÀ LẬP TRÌNH C TRÊN LINUX 24](#_Toc21705)

[1.Shell của UNIX/Linux 24](#_Toc17377)

[2. Sử dụng Shell như ngôn ngữ lập trình. 26](#_Toc16449)

[3. Cú pháp ngôn ngữ Shell 26](#_Toc18974)

[3.1. Sử dụng biến 27](#_Toc23487)

[3.2. Điều kiện 27](#_Toc21666)

[*So sánh toán học* 28](#_Toc28242)

[*Kiểm tra điều kiện trên tập tin* 28](#_Toc11047)

[3.3. Cấu trúc điều khiển 29](#_Toc21035)

[3.4. Danh shell thực thi lệnh (Lists) 30](#_Toc4876)

[3.5. Hàm (function) 31](#_Toc847)

[3.6. Các lệnh nội tại của shell 33](#_Toc23488)

[4. Demo 34](#_Toc20722)

# CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH LINUX

## Hệ điều hành Linux:

### Lịch sử phát triển:

* Năm 1991, Linus Torvalds, sinh viên của Đại học Tổng hợp Helsinki Phần Lan bắt đầu xem xét Minix, một phiên bản của Unix làm ra với mục đích nghiên cứu cách tạo ra một hệ điều hành Unix chạy trên máy PC với bộ vi xử lý Intel 80386.
* Ngày 25/8/1991, Linus cho ra version 0.01 và thông báo trên comp.os.minix về dự định của mình về Linux.
* 1/1992, Linus cho ra version 0.02 với shell và trình biên dịch C. Linux không cần Minix nữa để biên dịch lại hệ điều hành của mình. Linus đặt tên hệ điều hành của mình là Linux.
* 1994, phiên bản chính thức 1.0 được phát hành.
* Linux là một hệ điều hành dạng UNIX (Unix-like Operating System) chạy trên máy PC với bộ điều khiển trung tâm (CPU) Intel 80386 trở lên, hay các bộ vi xử lý trung tâm tương thích AMD, Cyrix. Linux ngày nay còn có thể chạy trên các máy Macintosh hoặc SUN Space. Linux thỏa mãn chuẩn POSIX.1.
* Linux được viết lại toàn bộ từ con số không, tức là không sử dụng một dòng lệnh nào của Unix để tránh vấn đề bản quyền của Unix. Tuy nhiên, hoạt động của Linux hoàn toàn dựa trên nguyên tắc của hệ điều hành Unix. Vì vậy, nếu một người nắm được Linux thì sẽ nắm được UNIX. Nên chú ý rằng giữa các Unix sự khác nhau cũng không kém gì giữa Unix và Linux.
* Linux là hệ điều hành phân phát miễn phí, phát triển trên mạng Internet, tựa Unix và được sử dụng trên máy tính cá nhân (PCs). Linux đã phát triển nhanh chóng và trở nên phổ biến trong thời gian ngắn. Nó nhanh chóng được nhiều người sử dụng vì một trong những lý do là không phải trả tiền bản quyền. Mọi người có thể dễ dàng download từ Internet hay mua tại các hiệu bán CD.
* Linux là hệ điều hành có hiệu năng cao, trong tất cả các máy tính có cấu hình cao hay thấp. Hệ điều hành này hỗ trợ các máy tính sử dụng 32 cũng như 64 bit và rất nhiều phần mềm khác nhau.
* Quá trình phát triển của Linux được tăng tốc bởi sự giúp đỡ của chương trình GNU (GNU’s Not Unix). Đó là chương trình phát triển các Unix có khả năng chạy trên nhiều nền tảng khác nhau. Đến hôm nay, cuối 2001, phiên bản mới nhất của Linux kernel là 2.6.11.3, có khả năng điều khiển các máy đa bộ vi xử lý và rất nhiều các tính năng khác.

### Những ưu điểm của Linux:

#### Khả năng tương thích với các hệ mở

Khả năng tương thích của một hệ điều hành giúp bạn chuyển nó từ một nền này sang một nền khác mà vẫn hoạt động tốt. Trước kia UNIX chỉ hoạt động trên một nền duy nhất, đó là máy điện toán mini DEC PDP-7. Hiện nay, UNIX chạy được trên bất kỳ nền nào, từ máy tính xách tay cho đến những máy tính lớn dạng mainframe. Nhờ tính tương thích này, các máy điện toán chạy UNIX trên nhiều nền khác nhau có thể liên lạc với nhau một cách chính xác và hữu hiệu với những loại nền khác.

#### Hỗ trợ ứng dụng

Hiện nay, Linux có hàng nghìn ứng dụng, bao gồm các chương trình báo biểu, cơ sở dữ liệu, xử lý văn bản... Ngoài ra, Linux cũng có hàng loạt trò chơi giải trí trên nền văn bản hoặc đồ họa.

#### Lợi ích cho giới chuyên nghiệp điện toán

Đến với Linux, giới điện toán sẽ có hàng loạt công cụ phát triển chương trình, bao gồm các bộ biên dịch cho nhiều ngôn ngữ lập trình hàng đầu hiện nay, chẳng hạn như C, C++, ...

### Khuyết điểm của Linux :

#### Hỗ trợ kỹ thuật

* Có lẽ điều trở ngại nhất của Linux là không có một công ty nào chịu trách nhiệm phát triển hệ điều hành Linux này. Nếu có điều gì trục trặc, bạn không thể gọi miễn phí cho một bộ phận hỗ trợ kỹ thuật nào cả.
* Thiếu nguồn trợ giúp kỹ thuật không chỉ đối với Linux mà cả với những ứng dụng Linux. Mặc dù, hiện có vài chương trình mang tính thương mại dành cho Linux, song đa phần lại là chương trình miễn phí do một nhóm nhỏ biên soạn rồi đưa lên mạng cho cả thế giới sử dụng chung.

#### Phần cứng

Một điều bất tiện nữa là thực sự Linux không dễ cài đặt và rất nhiều thành phần không tương thích với một vài phần cứng nào đó. Các nhà phát triển Linux là nhữhg người sống rãi rác trên hành tinh này, do đó không thể có một chương trình được đảm bảo chất lượng như thông lệ. Các nhà phát triển cảm thấy chương trình của mình dùng được là tung ra cho mọi người cùng xài chứ không có một thời gian thử nghiệm chương trình. Hơn nữa, các phần cứng mà Linux hỗ trợ tùy thuộc vào loại máy móc mà các nhà phát triển sử dụng khi soạn thảo đoạn mã. Chính vì thế mà Linux không thể chạy trên tất cả mọi nền phần cứng của PC hiện nay

### Kiến trúc của hệ điều hành Linux:

#### Hạt nhân (Kernel)

Là trung tâm điều khiển của hệ điều hành Linux, chứa các mã nguồn điều khiển hoạt động của toàn bộ hệ thống. Hạt nhân được phát triển không ngừng, thường có 2 phiên bản mới nhất, một bản dạng phát triển mới nhất và một bản ổn định mới nhất. Kernel được thiết kế theo dạng modul, do vậy kích thước thật sự của Kernel rất nhỏ. Chúng chỉ tải những bộ phận cần thiết lên bộ nhớ, các bộ phận khác sẽ được tải lên nếu có yêu cầu sử dụng. Nhờ vậy so với các hệ điều hành khác Linux không sử dụng lãng phí bộ nhớ nhờ không tải mọi thứ lên mà không cần quan tâm nó có sử dụng không.

Kernel được xem là trái tim của hệ điều hành Linux, ban đầu phát triển cho các CPU Intel 80386. Điểm mạnh của loại CPU này là khả năng quản lý bộ nhớ. Kernel của Linux có thể truy xuất tới toàn bộ tính năng phần cứng của máy. Yêu cầu của các chương trình cần rất nhiều bộ nhớ, trong khi hệ thống có ít bộ nhớ, hệ điều hành sử dụng không gian đĩa hoán đổi (swap space) để lưu trữ các dữ liệu xử lý của chương trình. Swap space cho phép ghi các trang của bộ nhớ xuất các vị trí dành sẵn trong đĩa và xem nó như phần mở rộng của vùng nhớ chính. Bên cạnh sử dụng swap space, Linux còn hỗ trợ các đặc tính sau :

* Bảo vệ vùng nhớ giữa các tiến trình, điều này không cho phép một tiến trình làm tắt toàn bộ hệ thống.
* Chỉ tải các chương trình khi có yêu cầu.

#### Shell

Shell cung cấp tập lệnh cho người dùng thao tác với kernel để thực hiện công việc. Shell đọc các lệnh từ người dùng và xử lý. Ngoài ra shell còn cung cấp một số đặc tính khác như : chuyển hướng xuất nhập, ngôn ngữ lệnh để tạo các tập tin lệnh tương tự tập tin bat trong DOS.

Có nhiều loại shell được dùng trong Linux. Điểm quan trọng để phân biệt các shell với nhau là bộ lệnh của mỗi shell. Ví dụ, C shell thì sử dụng các lệnh tương tự ngôn ngữ C, Bourne Shell thì dùng ngôn ngữ lệnh khác.

Shell sử dụng chính trong Linux là GNU Bourne Again Shell (bash). Shell này là shell phát triển từ Bourne Shell, là shell sử dụng chính trong các hệ thống Unix, với nhiều tính năng mới như : điều khiển các tiến trình, các lệnh history, tên tập tin dài …

#### Các tiện ích

Các tiện ích được người dùng thường xuyên sử dụng. Nó dùng cho nhiều thứ như thao tác tập tin, đĩa, nén, sao lưu tập tin … Tiện ích trong Linux có thể là các lệnh thao tác hay các chương trình giao diện đồ họa. Hầu hết các tiện ích dùng trong Linux là sản phẩm của chương trình GNU. Linux có sẵn rất nhiều tiện ích như trình biên dịch, trình gỡ lỗi, soạn văn bản … Tiện ích có thể được sử dụng bởi người dùng hoặc hệ thống. Một số tiện ích được xem là chuẩn trong hệ thống Linux như passwd, ls, ps, vi …

#### Chương trình ứng dụng

Khác với các tiện ích, các ứng dụng như chương trình word, hệ quản trị cơ sở dữ liệu ... là các chương trình có độ phức tạp lớn và được các nhà sản xuất viết ra.

### Các đặc tính cơ bản của Linux:

#### Đa tiến trình

Là đặc tính cho phép người dùng thực hiện nhiều tiến trình đồng thời. Ví dụ bạn vừa in, vừa soạn văn bản, vừa nghe nhạc… cùng một lúc. Máy tính sử dụng chỉ một CPU nhưng xử lý đồng thời nhiều tiến trình cùng lúc. Thực chất là tại một thời điểm CPU chỉ xử lý được một mệnh lệnh, việc thực hiện cùng lúc nhiều công việc là giả tạo bằng cách làm việc xen kẻ và chuyển đổi trong thời gian nhanh. Do đó người dùng cứ ngỡ là thực hiện đồng thời.

#### Tốc độ cao

Hệ điều hành Linux được biết đến như một hệ điều hành có tốc độ xử lý cao, bởi vì nó thao tác rất hiệu quả đến tài nguyên như : bộ nhớ, đĩa…

#### Bộ nhớ ảo

Khi hệ thống sử dụng quá nhiều chương trình lớn dẫn đến không đủ bộ nhớ chính (RAM) để hoạt động. Trong trường hợp đó, Linux dùng bộ nhớ từ đĩa là partition swap. Hệ thống sẽ đưa các chương trình hoặc dữ liệu nào chưa có yêu cầu truy xuất xuống vùng swap này, khi có nhu cầu thì hệ thống chuỵển lên lại bộ nhớ chính.

#### Sử dụng chung thư viện

Hệ thống Linux có rất nhiều thư viện dùng chung cho nhiều ứng dụng. Điều này sẽ giúp hệ thống tiết kiệm được tài nguyên cũng như thời gian xử lý.

#### Sử dụng các chương trình xử lý văn bản

Chương trình xử lý văn bản là một trong nhưng chương trình rất cần thiết đối với người sử dụng. Linux cung cấp nhiều chương trình cho phép người dùng thao tác với văn bản như vi, emacs, nroff

#### Sử dụng giao diện cửa sổ

Giao diện cửa sổ dùng Hệ thống X Window, có giao diện như hệ điều hành Windows. Với hệ thống này người dùng rất thuận tiện khi làm việc trên hệ thống. X window System hay còn gọi tắt là X được phát triển tại viện Massachusetts Institute of Technology. Nó được phát triển để tạo ra môi trường làm việc không phụ thuộc phần cứng. X chạy dưới dạng client –server. Hệ thống X window hoạt động qua hai bộ phận :

* Phần server còn gọi là X server
* Phần client được gọi là X window manager hay desktop environment.

X server sử dụng trong hầu hết các bản phân phối của Linux là Xfree86. Client sử dụng thường là KDE (K Desktop Environment) và GNOME (GNU Network Object Model Environment)

Dich vụ Samba sử dụng tài nguỵên đĩa, máy in với Windows. Tên Samba xuất phát từ giao thức Server Message Block (SMB) mà Windows sử dụng để chia sẻ tập tin và máy in. Samba là chương trình sử dụng giao thức SMB chạy trên Linux. Sử dụng Samba bạn có thể chia sẻ tập tin và máy in với các máy Windows

#### Network Information Service (NIS)

Dịch vụ NIS cho phép chia sẻ các tập tin password và group trên mạng. NIS là một hệ thống cơ sở dữ liệu dạng client-server, chứa các thông tin của người dùng và dùng để chứng thực người dùng. NIS xuất phát từ hãng Sun Microsystems với tên là Yellow Pages.

#### Lập lịch hoạt động chương trình, ứng dụng

Chương trình lập lịch trong Linux xác định các ứng dụng, script thực thi theo một sự sắp xếp của người dùng như: at, cron, batch.

#### Các tiện ích sao lưu dữ liệu

Linux cung cấp các tiện ích như tar, cpio và dd để sao lưu và backup dữ liệu. RedHat Linux còn cung cấp tiện ích Backup and Restore System Unix (BRU) cho phép tự động backup dữ liệu theo lịch.

#### Hỗ trợ nhiều ngôn ngữ lập trình.

Linux cung cấp một môi trường lập trình Unix đầy đủ bao gồm các thư viện chuẩn, các công cụ lập trình, trình biên dịch, chương trình debug chương trình mà bạn có thể tìm thấy trong các hệ điều hành Unix khác. Ngôn ngữ chủ yếu sử dụng trong các hệ điều hành Unix là C và C++. Linux dùng trình biên dịch cho C và C++ là gcc, chương trình biên dịch này rất mạnh, hỗ trợ nhiều tính năng. Ngoài C, Linux cũng cung cấp các trình biên dịch, thông dịch cho các ngôn ngữ khác như Pascal, Fortran, Java…

## 2. CentOS

### 1.6.Giới thiệu HĐH CentOS:

* CentOS là một hệ điều hành miễn phí được xây dựng và phát triển dựa trên hệ điều hành mã nguồn mở Linux. CentOS là chữ viết tắt của “Community Enterprise Operating System”. CentOS ra mắt công chúng vào tháng 5 năm 2004 và được phát triển dựa trên bản phân phối của Red Hat Enterprise Linux (RHEL).
* Hệ điều hành này được tạo ra với mục đích chính là xây dựng nền tảng hệ thống máy chủ miễn phí dành cho doanh nghiệp và duy trì khả năng tương thích nhị phân với RHEL. Ngoài ra, CentOS còn cung cấp một môi trường hoàn hảo để thực hiện các công việc liên quan đến lập trình. Để thực sự hiểu về CentOS, bạn cần nắm vững kiến thức về Linux. (Hệ điều hành Linuх đượᴄ phát triển dựa ᴠào hệ điều hành Uniх ᴠà đượᴄ phát hành miễn phí.)

### 1.7.Lịch sử hình thành và phát triển của CentOS:

* CentOS là gì? Nó xuất hiện đầu tiên vào năm 2004 và phát triển một cách mạnh mẽ cho đến nay. Trong đó, nổi bật nhất là phiên bản Tao Linux, nó được biết đến như một bản sao của RHEL. Tuy nhiên, vào tháng 6 năm 2006, cha đẻ của Tao Linux - David Parsley tuyên bố sẽ dừng dự án này vô thời hạn. Và nó biến mất khỏi thị trường
* Đến năm 2008, CentOS đã thành công thay thế cho Tao Linux.
* Đến tháng 7 năm 2009, CentOS đột ngột thông báo người sáng lập của hệ điều hành này – Lance Davis đã mất tích trong suốt năm 2008 trong khi vẫn đang giữ tên miền đăng ký của CentOS và tài khoản Paypal. Khiến cho hoạt động của CentOS bị gián đoạn không nhỏ. Tuy nhiên chỉ một tháng sau đó, CentOS lại thông báo là đã nhận lại được tên miền centos.info, centos.org.
* Phải đến gần 1 năm sau đó, tháng 7 năm 2010, CentOS mới thực sự ổn định lại và trở thành bản phân phối Linux phổ biến nhất với 30% trong tổng số tất cả các server Linux trên thế giới.
* Tháng 1 năm 2014, Redhat trở thành đơn vị sở hữu CentOS sau khi đưa ra tuyên bố rằng sẽ tài trợ cho CentOS để tạo ra môi trường lập trình tốt nhất cho các nhà phát triển mã code.

### 1.8.Các tính năng chính của CentOS

* RHEL và CentOS chia sẻ nhiều tính năng vì CentOS dựa trên mã nguồn của RHEL.
* Hiệu suất cao và tính khả dụng: Nó sử dụng Máy ảo dựa trên Kernel để ảo hóa và cung cấp tính khả dụng cao và hiệu suất tuyệt vời.
* Bản phân phối Linux ổn định: CentOS được hỗ trợ bởi một cộng đồng nhà phát triển chuyên dụng nhằm duy trì nó luôn cập nhật và đảm bảo khả năng tương thích với cả phần mềm và phần mềm mới. Hơn nữa, các nhà phát triển cốt lõi được hỗ trợ bởi một cộng đồng toàn cầu gồm những người dùng tình nguyện như những người đam mê Linux, quản trị viên mạng, quản trị viên hệ thống, v.v., những người kiểm tra các bản phát hành, quản lý các bản cập nhật và cung cấp trợ giúp trong quá trình phát triển.
* Cập nhật và hỗ trợ thường xuyên: Các phiên bản CentOS được cập nhật trung bình sáu tháng một lần và mỗi bản phát hành được hỗ trợ trong mười năm.
* Mức độ bảo mật cao: Đội ngũ an ninh tại Red Hat sẵn sàng phát hiện các mối đe dọa và đảm bảo mức độ an ninh cao. CentOS cũng đi kèm với phần mở rộng nhân Linux được tăng cường bảo mật.
* Khi nói đến thông số kỹ thuật, giống như bất kỳ hệ thống Linux nào khác, CentOS có rất nhiều thứ để cung cấp. Tài liệu nội bộ của CentOS bao gồm nhiều điểm tốt hơn; hầu hết các lập trình viên nên tự làm quen với bức tranh toàn cảnh trước khi quyết định chọn một máy chủ chuyên dụng CentOS. Trước khi bạn dùng thử CentOS cho chính mình, dưới đây là tất cả những gì bạn cần biết về nó.
* CentOS không phải là RHEL:
* RHEL đã được đóng gói lại dưới dạng CentOS. Trong khi đó, nhóm CentOS là một nhóm tình nguyện đóng gói các gói nguồn RHEL thành các tệp nhị phân có sẵn công khai. Sau đó, phần mềm được phân phối qua một số gương chung. Theo như tôi biết thì Red Hat và CentOS không có mối liên kết hoặc quan hệ đối tác trực tiếp nào. Trong quá khứ, việc thiếu các quan hệ đối tác chính thức đã gây ra nhiều vấn đề, chẳng hạn như khi Red Hat khăng khăng rằng khỏi CentOS, tất cả thương hiệu Red Hat nên được xóa bỏ.
* Nhiều tổ chức cung cấp hỗ trợ cho CentOS Linux, nhưng vẫn chưa có hỗ trợ thương mại thích hợp. Red Hat cung cấp hỗ trợ trực tiếp với RHEL. Để được hỗ trợ CentOS thương mại, người dùng CentOS phụ thuộc vào các chuyên gia CNTT và cộng đồng.
* Nhiều thông số kỹ thuật cấp cao nhất được bao gồm trong CentOS:CentOS bao gồm một số lượng lớn các thông số kỹ thuật độc đáo cho phép nhiều đổi mới hơn bên trong nền tảng này. CentOS 6 và 7 lần lượt dựa trên các nhân Linux 2.6.32 và Linux 3.10.0. Các bản phân phối này có tính chất “plug and play”, hỗ trợ các tệp mp3 từ các nguồn khác nhau và Firefox 60 làm trình duyệt internet mặc định.
* Kiến trúc x86\_64 trong CentOS 6 và 7 có thể phân bổ không gian 12 và 64 TB về khả năng của CPU. Trong khi đó, hệ thống tệp cục bộ có thể xử lý các kích thước tệp tối đa khác nhau, từ 2 TB đến 100 TB ở các cấu hình khác nhau. CentOS cũng làm cho nó có thể thêm một loạt các tính năng hữu ích. Ví dụ: các phiên bản CentOS hiện tại hỗ trợ Bluetooth gốc cũng như Linux được tăng cường bảo mật.
* CentOS có khả năng hỗ trợ Kiến trúc x86\_64 và x86: Hiện tại, bản phân phối Linux này hỗ trợ cả hệ thống x86\_64 và x86. Tuy nhiên, CentOS 7 là duy nhất vì nó hiện đang giúp các kiến ​​trúc ppc64, ppc64le, Arm32, i686 và Arm64 không chính thức, do cộng đồng duy trì. Trong khi CentOS 6 có thể hỗ trợ cả hai kiến ​​trúc
* Bạn có thể tải xuống CentOS miễn phí:CentOS là một bản phân phối có sẵn miễn phí để phân phối và sử dụng. CentOS mời mọi người can thiệp vào nó và tìm những cách mới để tăng chức năng trong bản phân phối Linux vì mô hình phát triển dựa vào cộng đồng của nó. CentOS cung cấp các tùy chọn khác nhau để tải xuống phần mềm của họ, bao gồm ISO tối thiểu, Amazon Web Services, torrent, DVD ISO, v.v. Đồng thời, các gói nguồn CentOS có sẵn thông qua kho tệp của chúng và không được kết hợp trong các bản tải xuống thông thường.
* Cộng đồng CentOS đang phát triển:
* Cộng đồng CentOS đã cam kết phát triển nó từng bản vá, đó là một trong những lý do quan trọng mà nó kiểm soát để duy trì một thị phần đáng kể. Xã hội cởi mở này đã gắn kết với nhau qua thời gian và phát triển các SIG riêng lẻ. Các tổ chức này quan tâm nhiều đến việc nâng cao các lĩnh vực chức năng cụ thể như ảo hóa và trải nghiệm người dùng.
* Ban điều hành CentOS chọn những SIG nào có thể bắt đầu hoạt động dưới sự bảo vệ của họ. Bất kỳ thành viên nào trong cộng đồng đều có thể tham gia SIG cho đến khi họ có thể chấp nhận những lời chỉ trích của cộng đồng và có đủ tài liệu. Sau đó, cộng đồng cũng được giải quyết để giúp đỡ cả người dùng có kinh nghiệm và người mới. Do đó, họ cung cấp nhiều danh sách gửi thư khác nhau để hỗ trợ người dùng nhận được lời khuyên trực tiếp từ các thành viên khác trong cộng đồng, có sẵn bằng các ngôn ngữ khác nhau.
* CentOS cung cấp hướng dẫn sử dụng cho người dùng: Dự án CentOS yêu cầu người dùng mới tận dụng tối đa mọi khoảnh khắc làm việc mà họ đã bỏ ra cho đến nay. Vì vậy, họ đã tạo một thư viện tài liệu để giúp họ bắt đầu. Thư viện này chứa hướng dẫn cài đặt từng bước cho các kiến trúc khác nhau và các ghi chú phát hành cho các sửa đổi lớn.

### 1.9.Kiến trúc của CentOS

* CentOS cũng tuân theo kiến trúc tương tự như bất kỳ bản phân phối Linux nào khác và chỉ hỗ trợ kiến trúc x86-64. Điều đó có nghĩa là nó hỗ trợ cả máy 32 bit và 64 bit.
* Lớp dưới cùng là phần cứng hoặc thiết bị vật lý. Như máy tính, bộ lưu trữ và thiết bị mạng.
* Phía trên là kernel là thành phần cốt lõi của Hệ điều hành. Và nó tương tác trực tiếp với phần cứng.
* Phía trên kernel là shell đóng vai trò là giao diện giữa người dùng và kernel.
* Trên tất cả các lớp này là lớp ứng dụng tương tác với shell và kernel để thực hiện một số tác vụ do người dùng xác định. Các ứng dụng bao gồm trình duyệt web, phiên bản văn bản, trình duyệt file, trình phát phương tiện, v.v.
* Kể từ phiên bản 7, CentOS chỉ hỗ trợ đầy đủ kiến trúc x86-64, trong khi các kiến trúc sau đây không được hỗ trợ:
* IA-32 trong tất cả các biến thể, đã hỗ trợ tạm thời giảm trong CentOS 7
* IA-32 không có phần mở rộng địa chỉ vật lý (PAE), không được hỗ trợ kể từ CentOS 6
* IA-64 (kiến trúc Intel Itanium), được hỗ trợ trong CentOS 3 và 4
* PowerPC 32-bit (Apple Macintosh và PowerMac chạy bộ xử lý G3 hoặc G4 PowerPC), hỗ trợ beta đã có sẵn trong CentOS 4
* IBM Mainframe (eServer zSeries và S/390), không được hỗ trợ kể từ CentOS 5
* Alpha,hỗ trợ đã có sẵn trong CentOS 4
* SPARC,hỗ trợ beta đã có sẵn trong CentOS 4
* Tính đến tháng 12 năm 2015, các bản phát hành AltArch của CentOS 7 có sẵn cho các biến thể ARMv7hl và AArch64 của kiến trúc ARM,[9] và các kế hoạch tồn tại để hỗ trợ các biến thể khác của kiến trúc ARM. Hỗ trợ ARM là một nỗ lực cộng đồng được phối hợp thông qua AltArch SIG.[9][168] Các bản phát hành AltArch của CentOS 7 cũng có sẵn cho kiến trúc IA-32 và chip Power ISA (chip POWER7 và POWER8).
* Phiên bản Cd trực tiếp của CentOS có sẵn tại mirror.centos.org. Một hình ảnh Live USB có thể khởi động của CentOS có thể được tạo thủ công hoặc bằng UNetbootin.
* Hình ảnh CentOS cũng có sẵn trên đám mây EC2 của Amazon,dưới dạng xây dựng sẵn và đã được xuất bản Amazon Machine Images (AMIs).

### 1.10.Ưa điểm và nhược điểm của CentOS:

* Ưa điểm:
* Centos được ra mắt từ năm 2004 được mệnh danh là một hệ điều hành cao cấp sở hữu nhiều tính năng hỗ trợ của chính cộng đồng mà tạo ra nó. Với tính tương đồng với RHEL, thì nó được xem là một môi trường khá thân thiện và hoàn hảo để lập trình cũng như là top những Linux distribution thống trị lại thế giới của Linux.
* Centos hoạt động mượt trên các mainframe đặc biệt là những người yêu thích dùng GUI, KDE…
* Điểm cộng của Centos còn được ví như một môi trường desktop đơn giản dưới sự hỗ trợ của Red Hat và cộng đồng mạng.
* Bên cạnh đó, Centos còn có khả năng cung cấp một môi trường có độ ổn định khá cao. Chính vì lẽ đó hứa hẹn sẽ mang lại những trải nghiệm mới lạ dành cho máy chủ doanh nghiệp. Hệ điều hành còn được Red Hat tài trợ những mã nguồn và trình quản lý gói RPM.
* Nhược điểm:
* Do cộng đồng người dùng và nhà phát triển còn nhỏ nên việc tìm hiểu cho người mới sẽ khó.
* Khó học hơn cho người mới bắt đầu vì không có nhiều bản phân phối được phát hành bởi RHEL.
* Các gói tiện ích cài đặt khá phức tạp phải cài đặt bằng trình quản lý gói yum.
* Ít cập nhật.

### 1.11.Giấy phép của CentOS

- Giống như Red Hat Enterprise Linux, bản quyền biên dịch CentOS Linux và CentOS Stream được cấp phép theo GPLv2.

- Trong phạm vi bạn giữ bất kỳ bản quyền nào trong việc lựa chọn, điều phối hoặc sắp xếp các gói tạo nên các bản phân phối CentOS Linux hoặc CentOS Stream, bạn sẽ cấp phép bản quyền đó theo GPLv2.

Giấy phép Công cộng Chung GNU phiên bản 2 – GPL v2 (The GNU General Public License v2) là giấy phép nguồn mở được sử dụng phổ biến nhất. Khoảng 70% các dự án trong kho phần mềm Sourceforge sử dụng GPLv2.

- Về GPL :

Giấy phép Công cộng GNU (tiếng Anh: GNU General Public License, viết tắt GNU GPL hay chỉ GPL) là giấy phép phần mềm tự do được sử dụng rộng rãi, đảm bảo cho người dùng cuối tự do chạy, nghiên cứu, sửa đổi và chia sẻ phần mềm.Giấy phép ban đầu được viết bởi Richard Stallman của Quỹ Phần mềm Tự do (FSF) cho Dự án GNU, và cấp cho người nhận chương trình máy tính quyền của Định nghĩa Phần mềm Tự do.GPL là giấy phép copyleft, có nghĩa là tác phẩm phát sinh chỉ có thể được phân phối theo các điều khoản cấp phép tương tự. Đây là sự phân biệt đối với giấy phép phần mềm tự do cho phép, trong đó giấy phép BSD và Giấy phép MIT được sử dụng rộng rãi là ví dụ. GPL là giấy phép copyleft đầu tiên để sử dụng chung.

Trong năm 2007, phiên bản thứ ba của giấy phép (GNU GPLv3) đã được phát hành để giải quyết một số vấn đề nhận thức với phiên bản thứ hai (GNU GPLv2) đã được phát hiện trong quá trình sử dụng lâu dài của nó. Để giữ cho giấy phép cập nhật, giấy phép GPL bao gồm một điều khoản "bất kỳ phiên bản sau" tùy chọn, cho phép người dùng lựa chọn giữa các điều khoản gốc hoặc các điều khoản trong các phiên bản mới như được FSF cập nhật. Các nhà phát triển có thể bỏ qua nó khi cấp phép phần mềm của họ; ví dụ hạt nhân Linux được cấp phép theo GPLv2 mà không có mệnh đề "bất kỳ phiên bản nào sau này".

Tóm lại về GPL một khi sử dụng và phân phối , bắt buộc phải sử dụng giấy phép GPL , không được phép đóng mã nguồn và thay đổi giấy phép .

### 1.12 Các tiện ích của Linux- CentOs

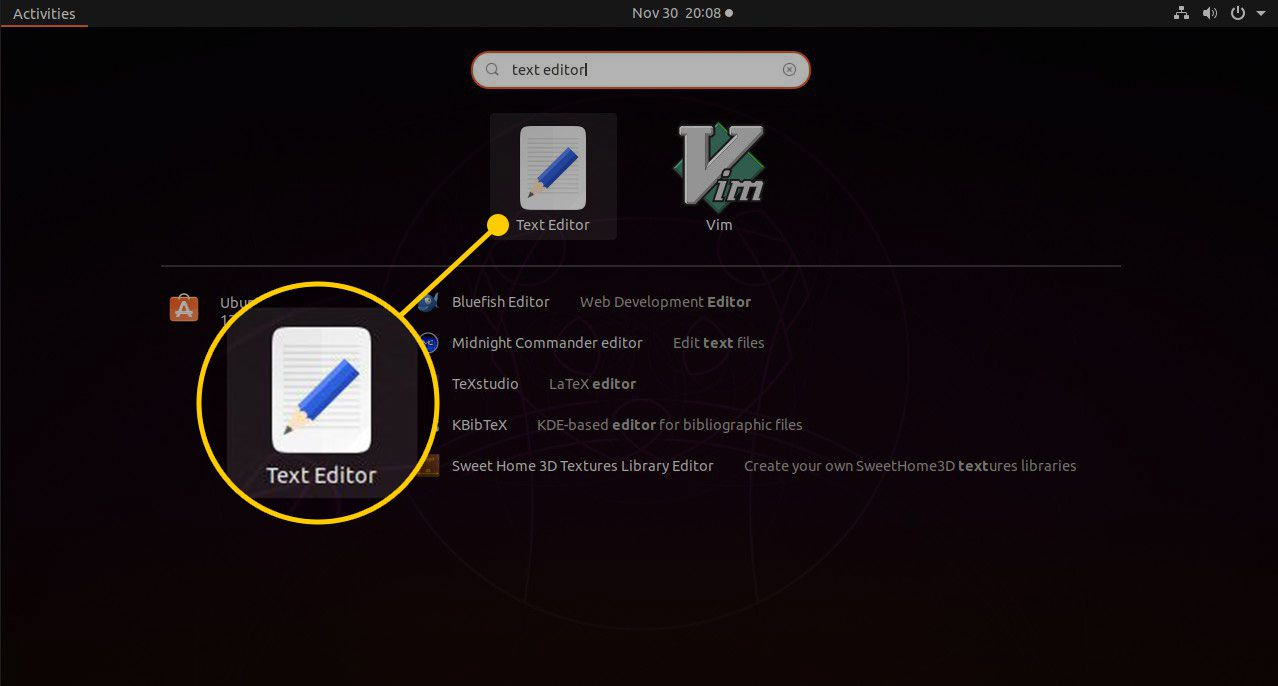
#### 1.12.1.Trình soạn văn bản

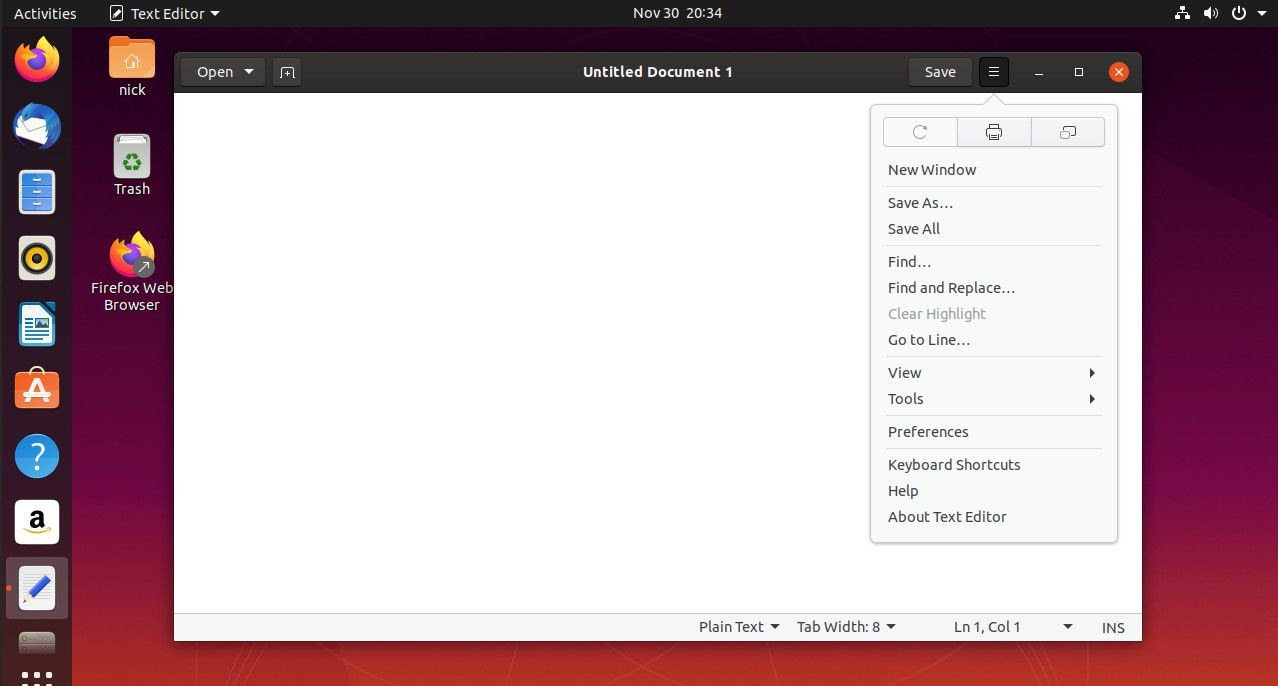
Linux cung cấp nhiều trình soạn thảo văn bản như ghi chú nhanh, soạn thảo một tài liệu, hoặc thậm chí viết mã một kịch bản: gedit, emacs, nano, gvim, vim, vi…

Một vài trình soạn thảo phổ biến nhất:

##### 1.12.1.1.Gedit

Gedit là một trình soạn thảo văn bản Linux thường được triển khai như một phần của môi trường máy tính để bàn GNOME. Mặc dù hầu hết các hướng dẫn và hướng dẫn về Linux đều khuyên bạn nên sử dụng trình soạn thảo nano hoặc vi để chỉnh sửa tệp văn bản và tệp cấu hình, trình chỉnh sửa gEdit dễ sử dụng hơn nano và vi và hoạt động giống như Microsoft Windows Notepad.



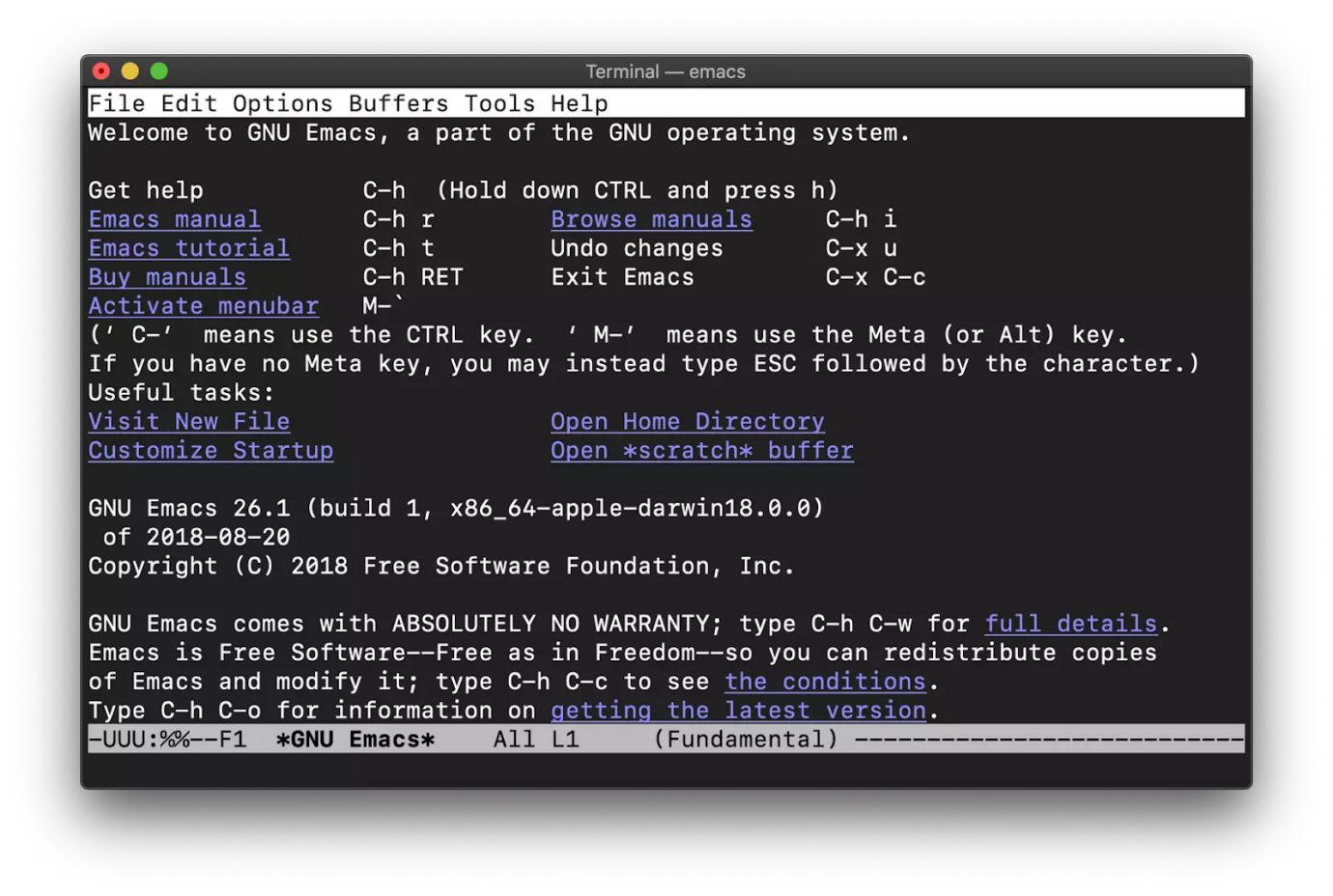


Một số tính năng trong gedit:

* Làm mới tài liệu
* In tài liệu
* Hiển thị toàn màn hình tài liệu
* Lưu tài liệu
* Tìm kiếm và thay thế văn bản
* Xóa văn bản được đánh dấu
* Đi đến một dòng cụ thể
* Hiển thị một bảng điều khiển bên
* Văn bản nổi bật
* Đặt ngôn ngữ
* Kiểm tra chính tả
* Chèn ngày và giờ

##### 1.12.1.2. Emacs

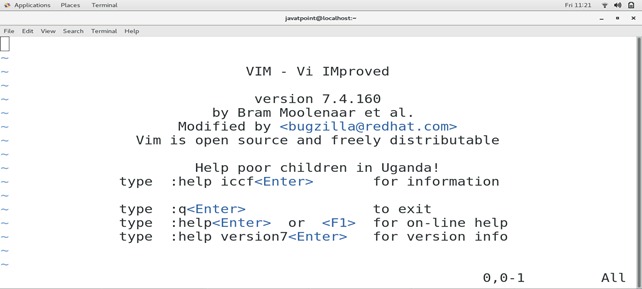
Emacs là trình soạn thảo văn bản đa chức năng. ... Emacs có thể chạy trên nhiều hệ điều hành khác nhau như các hệ thống giống Unix (GNU/Linux, các loại BSD, Solaris, AIX, v.v.), MS-DOS, Microsoft Windows, OpenVMS và Mac OS X. Emacs chạy trên giao diện văn bản lẫn đồ hoạ.



##### 1.12.1.3.Vim/Vi

VIM, viết tắt của VI Improved là trình soạn thảo văn bản dòng lệnh được sử dụng để tạo và xem các tệp văn bản.

Vim được sử dụng rất mạnh mẽ trong CLI (command-line interface). Linux sử dụng rất nhiều file cấu hình, chúng ta thường sẽ cần chỉnh sửa chúng và vim là một công cụ tuyệt vời để làm điều đó.



Cài đặt Vim bằng CentOS:

yum install vim-X11 vim-common vim-enhanced vim-minimal

Một số lệnh thường dùng trong Vim

* Khi đang ở chế độ nhập lệnh như vậy có thể thi hành một số tác vụ bằng cách gõ lệnh như sau (chú ý ký hiệu [return] là nhấn phím RETURN)

:q[retrun] thoát khỏi Vim

:q![retrun] ép thoát (hủy việc lưu lại)

:w[retrun] lưu nội dung đã thay đổi

:w![retrun] ép lưu lại

:wq[retrun] lưu nội dung đã thay đổi và thoát Vim

* Một số tác vụ khác khi ở chế đố nhập lệnh:

H - trỏ lênh đỉnh màn hình

L - trỏ xuống dưới màn hình

M - ở giữa màn hình

gg - đến dòng đầu tiên

G - đến dòng cuối

yy - copy dòng

p - paste

dd - xóa dòng

/ - tìm kiếm

* Để soạn thảo chuyển sang chế độ Insert

Khi đang trong chế độ nhập lệnh bạn gõ i hoặc nhấn INSERT sẽ vào chế độ INSERT

Khi ở chế độ INSERT thì có thể gõ chữ để chèn nội dung mới vào. Khi đã chỉnh sửa xong nội dung quay trở về chế độ nhập lệnh bằng cách nhấn phím ESC

* Chế độ REPLACE

Đang ở chế độ INSERT chuyển qua lại với chế độ REPLACE bằng cách nhấn phím INSERT, khi ở chế độ REPLACE thì gõ chữ vào sẽ thay thế chữ đang có ở vị trí con trỏ chứ không phải là chèn vào

#### 1.12.2. Tiện ích Setup

Trình tiện ích này được cung cấp để hỗ trợ cài đặt thiết bị, thiết lập cấu hình mạng.

Khởi động : #setup



#### 1.12.3. Tiện ích Fdisk

Fdisk là tiện ích quản lý phân vùng đĩa cứng trên Linux. Sử dụng fdisk, bạn có thể xem, tạo, thay đổi kích thước, xóa, thay đổi, sao chép và di chuyển các phân vùng. fdisk cho phép tạo tối đa bốn phân vùng chính được Linux cho phép với mỗi phân vùng yêu cầu kích thước tối thiểu 40mb.

Cú pháp: #fdisk  <device>

* Các câu lệnh Fdisk thường sử dụng

Lệnh FDISK/MBR được sử dụng để viết lại Master Boot Record;

FDISK/ CMBR < DISK>: nhằm khôi phục lại Master Boot Record trên ổ đĩa cứng cụ thể. Nó thực hiện chức năng tương tự như lệnh FDISK/MBR, nhưng khi sử dụng lệnh này, bạn có thể sử dụng với các ổ đĩa khác nhau;

FDISK 1/ PRI:100: để tạo ra một phân vùng 100 MB DOS trên ổ đĩa cứng.

Lệnh FDISK 1/EXT: 500: tạo ra một phân vùng 500 MB meg extended DOS trên ổ đĩa cứng;

Lệnh FDISK1/LOG:250: để tạo một ổ Logical 250MB trên ổ đĩa;

Lệnh FDISK/ Q: làm cho FDISK ngừng tự động khởi động;

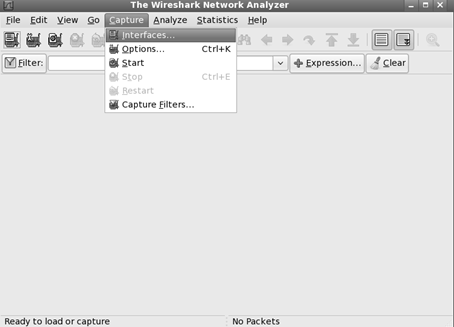
Lệnh FDISK/ STATUS: hiển thị trạng thái ổ cứng hiện tại.

#### 1.12.4. Theo dõi thông tin mạng

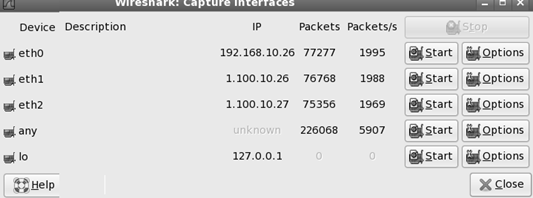
Đây là phần mềm bắt gói tin hàng đầu được sử dụng trong việc theo dõi, giám sát và phân tích traffic Network.

Trên CentOS, cài bằng câu lệnh:

#yum -y install wireshark-gnome



Chọn Start để bắt đầu theo dõi gói tin.



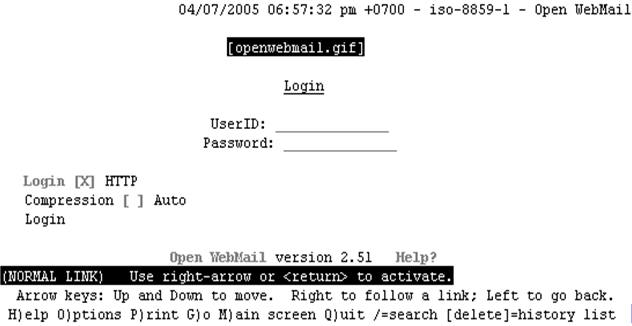
#### 1.12.5. Tiện ích Lynx

Lynx là một trình duyệt web không giống như phổ biến nhất này Nó được sử dụng thông qua một thiết bị đầu cuối và điều hướng là thông qua chế độ văn bản.

Đây là tiện ích cho phép duyệt web và kiểm tra thư điện tử có giao diện text.

Cú pháp: #lynx  <URL>

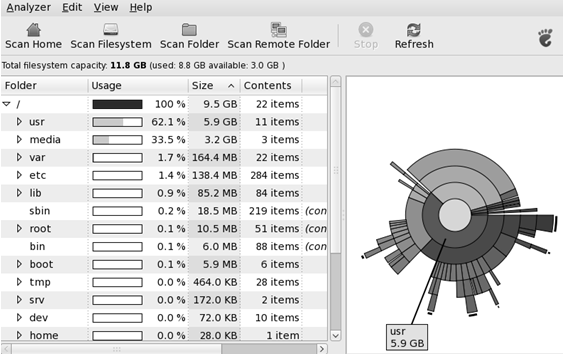
Ví dụ: #lynx  webmail.tatavietnam.vn



#### 1.12.6. Phân tích đĩa

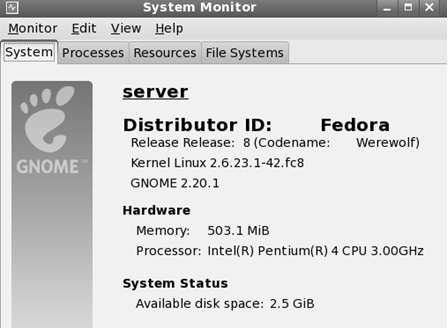
Để sử dụng tiện ích, ta chọn Applications -> System Tools -> Disk Usage Analyzer

Chọn Scan Home để xem thông tin chi tiết trong home directory của người dùng.



#### 1.12.7. Theo dõi hệ thống

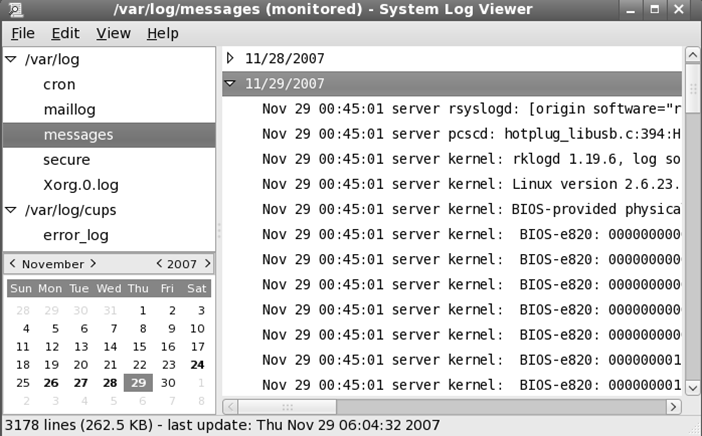
Để sử dụng tiện ích, ta chọn Applications -> System Tools -> System Monitor



Resouces cho ta theo dõi được thông tin CPU, MEM, SWAP, Network đang được sử dụng

#### 1.12.8. Quản lý log

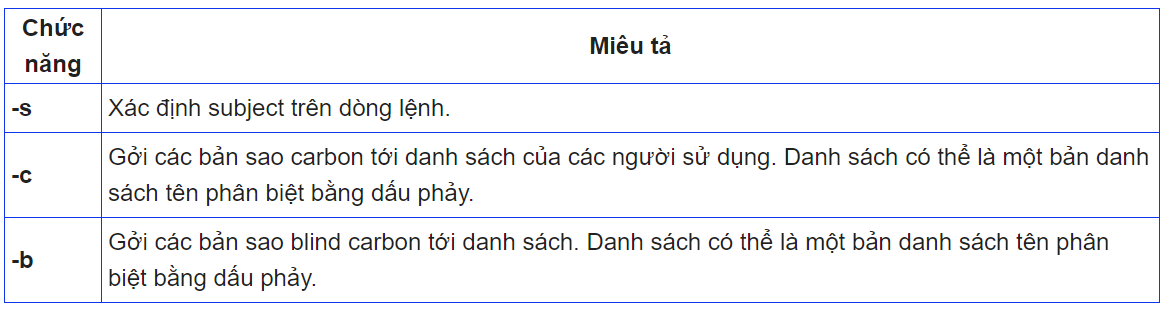
Linux cung cấp tiện ích System Log Viewer để theo dõi và quản lý log file cho hệ thống. Để xem log file ta chọn Applications -> System Log



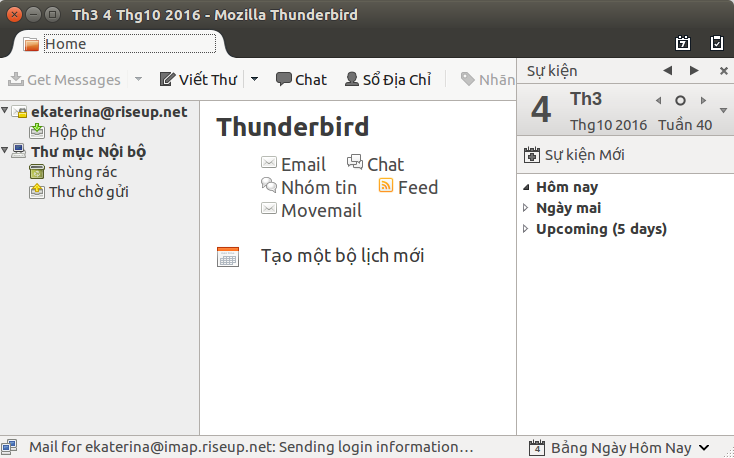
#### 1.12.9. Tiện ích mail

Lệnh mail để gửi và nhận mail. Cú pháp để gửi một Email:

$mail [-s subject] [-c cc-addr] [-b bcc-addr] to-addr



Ngoài ra, ta còn có thể sử dụng các trình duyệt như ThunderBird hay Revolution để quản lý email như là Outlook của windows.



# CHƯƠNG 2: LẬP TRÌNH SHELL VÀ LẬP TRÌNH C TRÊN LINUX

## 1.Shell của UNIX/Linux

Mọi thứ được thực hiện trên Unix đều bởi tiến trình. Vậy tạo ra tiến trình như thế nào? Cách thứ nhất là viết ra các chương trình mà các chương trình này biết cách tạo ra tiến trình (C/C++). Tuy nhiên cách này đòi hỏi nhiều hiểu biết và nỗ lực. Cũng như các hệ điều hành làm việc kiểu ảo khác, Unix hổ trợ một phương tiện xử lí lệnh làm giao diện giữa lệnh máy (mà người dùng đưa vào) và việc thực thi của lệnh đó (bởi Unix). Phương tiện đó gọi là shell. Từ khi ra đời Unix đã có vài kiểu shell, đó là Bourne, C, Korn shell. Thực ra shell làm gì? Tòan bộ mục đích của shell là để khởi động các tiến trình xử lí lệnh đưa vào: yêu cầu đưa (dòng) lệnh vào, đọc đầu vào, thông dịch dòng lệnh đó, và tạo ra tiến trình để thực hiện lệnh đó. Nói cách khác shell quét dòng lệnh đưa vào máy tính, cấu hình môi trường thực thi và tạo tiến trình để thực hiện lệnh.

Thực tế tìm hiểu shhell là để học ngôn ngữ lập trình, cho dù không phức tạp như C, hay các ngôn ngữ khác, nhưng cũng phải qua những đòi hỏi cần thiết. Trong Unix/Linux có các lọai shell khác nhau và có thể lựa chọn để dùng theo nhu cầu mà người dùng thấy phù hợp.

Linux/Unix tách biệt các ứng dụng, lệnh gọi các hàm chức năng của nhân thành những đơn thể rất nhỏ (tiến trình). Tuy nhiên, nhiều lệnh của Linux có thể kết hợp lại với nhau để tạo nên chức năng tổng hợp rất mạnh mẽ.

Ví dụ: $ ls -al | more

Lệnh trên được kết hợp bằng hai lệnh, ls liệt kê toàn bộ danh sách tệp và thư mục trên đĩa ra màn hình, nếu danh sách quá dài, ls chuyển dữ liệu kết xuất cho lệnh more xử lý hiển thị kết quả thành từng trang màn hình. Linux có cách kết hợp dữ liệu kết xuất của các lệnh với nhau thông qua cơ chế chuyển tiếp (redirect), ống dẫn (pipe).

Hầu hết các Shell trong Unix/Linux sử dụng một ngôn ngữ gần giống với C (điều này cũng dễ hiểu bởi trong thế giới Unix/Linux, C là ngôn ngữ lập trình thống trị). Ngôn ngữ Shell càng giống C thì lập trình viên hay người điều khiển Linux càng cảm thấy thân thiện với HĐH.

Các shell trên Unix/Linux:

1. sh: shell nguyên thủy áp dụng cho Unix, dòng shell sử dụng cấu trúc lệnh của C làm ngôn ngữ kịch bản. Được tạo ra đầu tiên bởi Bia Joy. Là shell thông dụng thứ hai sau bash shell.
2. bash: shell chủ yếu của Linux. Ra đời từ dự án GNU. bash (Viết tắt của Bourne Again Shell có lợi điểm là mã nguồn được công bố rộng rãi.
3. rc: shell mô rộng của csh với nhiều tương thích với ngôn ngữ C hơn. rc cũng ra đời từ dự án GNU.

## 2. Sử dụng Shell như ngôn ngữ lập trình.

Trước hết bạn dùng lệnh

$cat > first.sh hay các trình soạn thảo như vi hay emacs (hoặc mc) để soạn nội  
dung tập tin first.sh như sau:  
# ! /bin/ sh  
# first.sh  
# Script nay se tìm trong thư mục hiện hành các chuỗi  
# mang nội dung main( ) , nội dung của fìle sẽ được hiển thị ra màn hình nếu tìm thấy.  
for file in \*  
do  
 if grep -l 'main( ) ' $file  
 then  
 more $fỉle  
 fi  
done  
exit 0

Không như chú thích của C, một dòng chú thích (comment) trong ngôn ngữ shell bắt đầu bằng ký tự #. Tuy nhiên Ở đây có một chú thích hơi đặc biệt đó là #!/bin/sh. Đẩy thực sự không phải là chú thích. Cặp ký tự #! là chỉ thị yêu cấu shell hiện tại triệu gọi shell sh nằm trong thư mục /bin. Shell sh sẽ chịu trách nhiệm thông dịch các lệnh nằm trong tập tin script được tạo.

Chỉ thị #! Còn được dùng để gọi bất kì chương trình nào ta muốn chạy trước khi script tiếp theo được dịch. Lệnh exit bảo đảm rằng script sau khi thực thi sẽ trả về mã lỗi, đây là cách mà hầu hết các chương trình nên làm, mặc dù mã lỗi trả vế ít khi được dùng đến trong trường hợp thực hiện tương tác trực tiếp từ dòng lệnh. Tuy nhiên, nhận biết mã trả về của một đoạn script sau khi thực thi, lại thường rất có ích nếu bạn triệu gọi script từ trong một script khác. Trong đoạn chương trình trên, lệnh exit sẽ trả về 0, cho biết script thực thi thành công và thoát khỏi shell gọi nó. Mặc dù khi đã lưu tập tin script với tên .sh, nhưng UNIX và Linux không bắt buộc điều này. Hiếm khi Linux sử dụng phần đuôi mở rộng của tập tin làm dấu hiệu nhận dạng, do đó tệp tệp script có thể là tùy ý. Tuy vậy .sh vẫn là cách chúng ta nhận ngay ra một tập tin có thể là script của shell một cách nhanh chóng.

## 3. Cú pháp ngôn ngữ Shell

Chúng ta đã thấy cách viết lệnh và gọi thực thi tập tin scirpt. Phần tiếp theo nay dành cho bạn khám phá sức mạnh của ngôn ngữ lập trình shell. Trái với lập trình bằng trình biên dịch khó kiểm lỗi và nâng cấp, lập trình script cho phép bạn dễ dàng sửa đổi lệnh bằng ngôn ngữ văn bản. Nhiều đoạn script nhỏ có thể kết hợp lại thành một script lớn mạnh mẽ và rất hữu ích. Trong thế giới UNIX và Linux đôi lúc gọi thực thi một chương trình, bạn khó mà biết được chương trình được viết bằng script hay thực thi theo mã của chương trình nhị phân, bởi vì tốc độ thực thi và sự uyển chuyển của chúng gần như ngang nhau. Phần này chúng ta sẽ học về:

• Biến: kiểu chuỗi, kiểu số, tham số và biến môi trường  
• Điều kiện: kiểm tra luận lý Boolean bằng shell  
• Điều khiển chương trình: if, elif, for , while, until, case  
• Danh shell  
• Hàm  
• Các hình nội tại của shell  
• Lấy về kết quả của một lệnh

##### 3.1. Sử dụng biến

Thường bạn không cần phải khai báo biến trước khi sử dụng. Thay vào đó biến sẽ được tự động tạo và khai báo khi lần đầu tiên tên biến xuất hiện, chảng hạn như trong phép gán. Mặc định, tất cả các biến đều được khởi tạo và chứa trị kiểu chuỗi (string). Ngay cả khi dữ liệu mà bạn đưa vào biến là một con số thì nó cũng được xem là định dạng chuỗi. Shell và một vài lệnh tiện ích sẽ tự động chuyển chuỗi thành số để thực hiện phép tính khi có yêu cầu. Tương tự như bản thân hệ điều hành và ngôn ngữ C, cú pháp của shell phân biệt chữ hoa chữ thường, biến mang tên foo, Foo, và FOO là ba biến khác nhau.

Ví dụ:

$ xinahao=hello  
$ echo $xinchao

Output: Hello  
$ xin chao= "I am here"  
$echo $xin chao  
Output: I am here  
$ xinchao=12+l  
$echo $xin chao  
Output: 12+1

##### 3.2. Điều kiện

Nền tảng cơ bản trong tất cả ngôn ngữ lập trình, đó là khả năng kiểm tra điều kiện và đưa ra quyết định rẽ nhánh thích hợp tùy theo điều kiện đúng hay sai. Trước khi tìm hiểu cấu trúc điều khiển của ngôn ngữ script, ta hãy xem qua cách kiểm tra điề kiện. Một script của shell có thể kiểm tra mã lỗi trả về của bất kỳ lệnh nào có khá năng triệu gọi từ dòng lệnh, bao gồm ả những tập tin lệnh script khác. Đó là lý do tại sao chúng ta thường sử dụng lệnh exit ở cuối mỗi scipt khi kết thúc.

Dước đây là cách sử dụng lệnh lệnh if, tuy đơn giản nhưng được sử dụng nhiều nhất. if kiểm tra điều kiện đúng hoặc sai để thực thi biểu thức thích hợp

Ví dụ:

if [-f hello.c ]  
then  
 . . .  
fi

Lưu ý, phải đặt khoảng trắng giữa lệnh [] và biểu thức kiểm tra, và dĩ nhiên giữa một lệnh và tham số truyềncho lệnh phải phân cách nhau bằng khoảng trắng để trình biên dịch có thể hiểu.

So sánh chuỗi

So sánh Kết quả

stringl = string2 true nếu 2 chuỗi bằng nhau (chính xác từng ký tự)

string1 != string2 true nếu 2 chuỗi không bằng nhau

-n string1 true nếu string1 không rổng

-z stringl true nếu string1 rỗng (chuổi null)

***So sánh toán học***

So sánh Kết quả

expression1 -eq expression2 true nếu hai biểu thức bằng nhau

expression1 -ne expression2 true nếu hai biểu thức không bằng nhau

expression1 -gt expression2 true nếu biểu thức expression1 lớn hơn expression2

expreesion1 -ge expression2 true nểu biểu thức expression1 lớn hơn hay bằng

expression2

expression1 -lt expression2 true nếu biểu thức expression1 nhỏ hơn expression2

expression1 -le exprebbion2 true nếu biểu thức expression1 nhỏ hơn hay bằng

expression2

!expression true nếu biểu thức expression là false (toán tử not)

***Kiểm tra điều kiện trên tập tin***

-d file true nếu file là thư mục

-e file true nếu file tồn tại trên đĩa

-f file true nếu file là tập tin thông thường

-g file true nếu set-group-id được thiết lập trên file

-r file true nếu file cho phép đọc

-s f ile true nếu kích thước file khác 0

-u file true nếu set-ser-id được áp đặt trên file

-w file true nếu file cho phép ghi

-x file true nếu file được phép thực thi

##### 3.3. Cấu trúc điều khiển

Shell cung cấp cấu trúc lệnh điều khiển rất giống với các ngôn ngữ lặp trình khác đó là if, elif, for, while, until, case. Đối với một vài cấu trúc lệnh (ví dụ như case), shell đưa ra cách xử lý uyển chuyển và mạnh mẻ hơn. Những cấu trúc điếu khiển khác nếu có thay đổi chỉ là những thay đổi nhỏ không đáng kể.

**Lệnh for**: Sử dụng for để lặp lại một số lần với các giá trị xác định. Phạm vi lặp có thể nằm trong một tập hợp chuỗi chỉ định tường minh bởi chương trình hay là kết qủa trả về từ một biến hoặc biểu thức khác.

Ví dụ:

#!/bin/sh  
for foo in bar fud 13  
do  
 echo $foo  
done  
exit 0

Output:

$./ for\_loop.sh  
bar  
fud  
13

**Lệnh** while: Lệnh while cho phép thực hiện lặp vô hạn khi điều kiện kiểm tra vẫn còn đúng.

Ví dụ:

#!/bin/sh  
echo "Enter password"  
read trythis  
while [ "$trythis" != "secret" ]; do  
 echo "Sorry, try again"  
 read trythis  
done  
exit 0

Output:

$./password.sh

Enter password:

abc

Sorry, try again

secret #gõ đúng

**Lệnh** case: Lệnh case có cách sử dụng hơi phức tạp hơn các lệnh đã học, nhưng lệnh case rất linh động. case cho phép thực hiện so khớp nội dung của biến với một chuỗi mẫu pattern nào đó. Khi một mẫu được so khớp, thì (lệnh) statement tương ứng sẽ được thực hiện. Hãy lưu ý đặt hai dấu chầm nhảy ;; phía sau mỗi mệnh đề so khớp pattern, shell dùng dấu hiệu này để nhận dạng mẫu pattern so khớp tiếp theo mà biến cần thực hiện

Ví dụ:

#!/bin/sh  
echo "Is it morning? Please answer yes or no"  
read timeofday  
case "$timeofday" in  
 "yes") echo "Good Morning";;  
 "no" ) echo "Good Afternoon";;  
 "y" ) echo "Good Morning";;  
 "n" ) echo "Good Afternoon";;  
 \* ) echo "Sorry, answer not recognised";;  
esac  
exit 0

Cách thực hiện: Sau khi người dùng nhập vào câu trả lời, lệnh case sẽ lấy nội dung của biến $timeofday so sánh với từng chuỗi. Khi gặp chuỗi thích hợp nó sẽ thực thi lệnh đằng sau dấu) và kết thúc (không tiếp tục so khớp với các mẫu khác). Ký tự đại diện \* cho phép so khớp với mọi loại chuỗi. \* thường được xem như trường hợp so sánh đúng cuối cùng nếu các mẫu so sánh trước đó thất bại. Bạn có thể xem \* là mệnh đề default trong lệnh switch của C hay case … else của Pascal.

##### 3.4. Danh shell thực thi lệnh (Lists)

**Danh sách AND (&&):** Danh shell AND cho phép thực thi một chuỗi lạnh kề nhau, lệnh sau chỉ thực hiện khi lệnh trước đã thực thi và trả về mã lỗi thành công. Cú pháp sử dụng như sau:  
Statement1 && statement2 && statement3 && . . .

Ví dụ:

#!/bin/sh  
touch file\_one  
rm -f file\_two  
if [ -f file\_one ] && echo "hello" && [ -f file\_two ] && echo "there"  
then  
 echo -e "in if"  
else  
 echo -e "in else"  
fi  
exit 0

Output:

$./and\_list.sh  
hello  
in else

**Danh sáchl OR ( || ):** Danh shell OR cũng tương tự với AND là thực thi một dãy các lệnh, nhưng nếu có một lệnh trả vế true thì việc thực thi ngừng lại. Cú pháp như sau:  
statementl || statement2 || statement3 && . . .

Ví dụ:

#!/bin/sh  
rm -f file\_one  
if [ -f file\_one ] || echo "hello" || echo "there"  
then  
 echo "in if"  
else  
 echo "in else"  
fi  
exit 0

Output:

$./and\_list.sh  
hello  
in if

##### 3.5. Hàm (function)

Tương tự các ngữ trình khác, shell cho phép bạn tự tạo hàm hay thủ tục để triệu gọi bên trong script. Mặc dù bạn có thể gọi các script con khác bên trong script chính, chúng tương tự như việc gọi hàm. Tuy nhiên triệu gọi các script con thường tiêu tốn tài nguyên và không hiệu quả bằng triệu gọi hàm.

Ví dụ:

#!/bin/sh  
foo() {  
 echo "Function foo is executing"  
}  
echo "script starting"  
foo  
echo "script ended"  
exit 0

Output:

$./my\_function.sh  
script starting  
Function foo is executing  
script ending

**Cách truyền tham số:** Shell không có cách khai báo tham số cho hàm như cách của C, Pascal hay các ngôn ngữ lập trình thông thường khác. Việc truyền tham số cho hàm tương tự như truyền tham số trên dùng lệnh. Ví dụ để truyền tham số cho foo(), ta gọi hàm như sau  
foo "paraml", "param2", param3 . . .

Ví dụ:

#!/bin/sh  
yes\_or\_no() {  
 echo "In function parameters are $\*"  
 echo "Param 1 $1 and Param2 $2"  
 while true  
 do  
 echo -n "Enter yes or no"  
 read x  
 case "$x" in  
 y | yes ) return 0;;  
 n | no ) return 1;;  
 \* ) echo "Answer yes or no"  
 esac  
done  
}  
echo "Original parameters are $\*"  
if yes\_or\_no "Is your name” “ $1?"  
then  
 echo "Hi $1"  
elif  
 echo "Never mind"  
fi  
exit 0

Output:

$,/get\_name.sh HoaBinh SV  
Original parameters are HoaBinh SV  
In function parameters are Is your name HoaBinh  
Param 1 Is your name param 2 HoaBinh  
Is your name HoaBinh ?  
Enter yes or no : yes  
Hi HoaBinh, nice name

##### 3.6. Các lệnh nội tại của shell

**Break:** Tương tự ngôn ngữ C, shell cung cấp lệnh break để thoát khỏi vòng lập for,while hoặc until bất kề điều kiện thoát của các lệnh này có diễn ra hay không.

Ví dụ:

#!/bin/sh  
rm -rf fred\*  
echo > fred1  
echo > fred2  
mkdir fred3  
echo > fred4  
for file in fred\*  
do  
 if [ -d "$file" ]; then  
 break;  
 fi  
done  
echo first directory fred was $file  
exit 0

Đoạn script trên dùng lệnh for để duyệt toàn bộ tên của tập tin và thư mục hiện hành bất đầu bằng chuỗi fred. Khi phát hiện thư mục đầu tiên trong danh shell các tập tin, sẽ in ra tên thư mục và dùng break rể thoát khỏi vòng lặp (không cần duyệt tiếp các tập tin khác).

**Cotinue:** Lệnh continue thường được dùng bên trong vòng lặp, continue yêu cầu quay lại thực hiện bước lặp kế tiếp mà không cần thực thi các khối lệnh còn lại.

Ví dụ:

#!/bin/sh  
rm -rf fred\*  
echo > fred1  
echo > fred2  
mkdir fred3  
echo > fred4  
for file in fred\*  
do  
 if [ -d "$file" ]; then  
 continue  
 fi  
 echo file is $file  
done  
exit 0

Đoạn script trên dùng lệnh for để duyệt toàn bộ tên của tập tin và thư mục hiện hành bắt đầu bằng chuỗi fred. Nếu kiểm tra tên tập tin là một thư mục, thì continue yêu cầu quay lại duyệt tiếp file khác. Ngược lại lệnh echo sẽ in ra tên tệp.

**Printf**: Lệnh printf của shell tương tự printf của thư viện C. Mặc dù vậy, cơ bản printfcủa shell có một số hạn chế sau: không hỗ trợ định dạng số có dấu chấm động (float) bởi vì tất cả các tính toán của shell đấu dựa trên số nguyên. Dầu sổ \ dùng chỉ định các hiển thị đặc biệt trong chuỗi (xem bảng dưới). Dấu % dùng định dạng số và kết xuất chuỗi. Dưới đây là danh sách các ký tự đặc biệt có thể dùng với dấu \, chúng được là chuỗi thoát.

Chuỗi thoát (escape sequence) Ý nghĩa

\\ Cho phép hiển thị ký tự \ trong chuỗi

\a Phát liếng chuông (beep)

\b Xóa backspace

\f Đẩy dòng

\n Sang dòng mới

\r Về đầu dòng

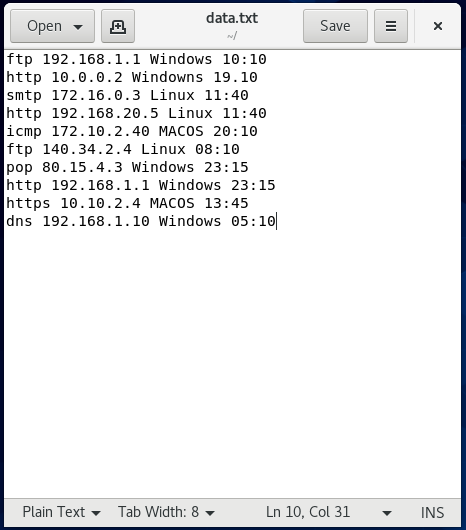
\t Căn tab ngang

\v căn tab dọc

**Return:** Lệnh return dùng để trả về giá trị của hàm. Lệnh return không tham số sẽ trả về mã lỗi của lệnh vừa thực hiện sau cùng.

## 4. Demo

Ta có 1 firewall lưu trữ các kết nối ra vào trong hệ thống, firewall đó đã lưu lại dữ liệu kết  
nối dưới tên là data, có cấu trúc như sau:



1. Viết 1 hàm nhanthongtin() để cho hệ thống có thể lấy các thông tin chép vào file data. (Trong tình huống viết giả lập này, các em thay hệ thống nhập thông tin vào file data theo cấu trúc dạng như trên).  
2. Sau khi đã có file dữ liệu data làm ở câu trên, hãy viết 1 hàm tên là xuatthongtin() để xuất ra toàn bộ thông tin đang nằm trong file data.   
3. Hãy viết hàm tên là: getIPfromOS() cho người dùng nhập vào 1 hệ điều hành, xuất ra thông tin các địa chỉ IP tương ứng của hệ điều hành.   
4. Hãy viết hàm tên là getIPfromProtocol cho người dùng nhập vào 1 giao thức, hãy xuất ra thông tin về địa chỉ IP đã truy cập.   
5. Hãy viết hàm tên là CountProtocol nhập vào giao thức, hãy đếm xem giao thức này được kết nối bao nhiêu lần.  
6. Hãy viết hàm tên là findOS để kiểm tra xem trong file data, có bao nhiêu hệ điều hành truy cập. Lưu tên hệ điều hành này vào 1 mảng.   
7. Hãy viết hàm Info() làm công việc: từ mảng đã làm ở câu trên, hãy đếm xem, mỗi hệ điều hành như vậy truy cập bao nhiêu lần. Chép vào 1 file tên là thongke, theo dạng như sau:   
Windows 5  
Linux 3  
8. Hãy thực hiện xuất ra thông tin sau trên chương trình chính  
“Hay thuc hien lua chon:”  
“1. Nhap thong tin vao file data  
“2. Xuat toan bo thong tin trong file data”  
“3. Nhap vao he dieu hanh de biet dia chi IP”  
“4. Nhap vao giao thuc de biet IP”  
“5. Nhap vao giao thuc de biet so lan ket noi”  
“6. Thuc hien thong ke”  
“7. Ket thuc”  
Lưu ý: khi người dùng nhập vào số 7 mới cho kết thúc chương trình. Khi chọn số 6, sẽ cho thực hiện 2 hàm findOS và info.

Dưới đây là mã nguồn của từng câu. Để dễ hiểu chúng ta tạo từng phần của mã lệnh, sau đó có thể kết hợp chúng lại như một tổng thể duy nhất.

Câu 1: Như thường lệ phần đầu của script thường là lời triệu gọi shell thực thi sh, hàm này giúp ta nhập thêm thông tin vào file data.txt mình đã tạo sẵn và chứa dữ liệu theo đề bài.

|  |
| --- |
| #!/bin/bash  nhapthongtin(){      echo "Nhap"      read nhap      echo $nhap >> data.txt  } |

Câu 2: Hàm while giúp ta đọc toàn bộ dữ liệu trên file đến khi hết dữ liệu để đọc theo dòng, đồng thời xuất ra màn hình.

|  |
| --- |
| xuatthongtin(){      echo "Xuat file data.txt"      while read line      do          echo ${line}      done < data.txt  } |

Câu 3: Câu “`echo $line | awk 'BEGIN{FS=" "}{print $3}'`” giúp ta tách được từ thứ 3 từ dòng, để giúp ta so sánh với từ ta đã nhập.

|  |
| --- |
| getIPfromOS(){          echo "Nhap he dieu hanh de xuat IP"          read tim1          while read line          do              data=`echo $line | awk 'BEGIN{FS=" "}{print $3}'`              if [ "$data" = "$tim1" ]              then                  data=`echo $line | awk 'BEGIN{FS=" "}{print $2}'`                  echo $data              fi          done < data.txt  } |

Câu 4: Tương tự như câu 3, hàm này giúp ta tách được từ thứ nhất và thứ hai để so sánh.

|  |
| --- |
| getIPfromProtocol(){          echo "Nhap giao thuc de xuat IP"          read tim2          while read line          do              data=`echo $line | awk 'BEGIN{FS=" "}{print $1}'`              if [ "$data" = "$tim2" ]              then                  data=`echo $line | awk 'BEGIN{FS=" "}{print $2}'`                  echo $data              fi          done < data.txt  } |

Câu 5:

|  |
| --- |
| CountProtocol(){          echo "Nhap giao thuc de bet so lan ket noi"          read tim2          dem=0          while read line          do              data=`echo $line | awk 'BEGIN{FS=" "}{print $1}'`              if [ "$data" = "$tim2" ]              then                  data=`echo $line | awk 'BEGIN{FS=" "}{print $2}'`                  let dem++              fi          done < data.txt          echo $dem  } |

Câu 6:

|  |
| --- |
| findOS(){      dem=0      while read line          do              data=`echo $line | awk 'BEGIN{FS=" "}{print $3}'`              a[$dem]=$data              let dem++          done < data.txt      dem1=0      for((i=0;i<${#a[\*]};i++))      do          kq=0          for((j=0;j<$i;j++))          do              if [ "${a[$i]}" = "${a[$j]}" ]              then                  kq=1              fi          done      if [ $kq -eq 0 ]      then          b[$dem1]=${a[$i]}          let dem1++      fi      done      echo "So he dieu hanh"      echo ${b[\*]}  } |

Câu 7:

|  |
| --- |
| Info(){      for((i=0;i<${#b[\*]};i++))      do          dem=0          echo ${b[$i]}          while read line          do              data=`echo $line | awk 'BEGIN{FS=" "}{print $3}'`              if [ "$data" = "${b[$i]}" ]              then                  let dem++              fi          done < data.txt          echo $dem          echo ${b[$i]} $dem >> thongke.txt      done  } |

Câu 8: Chương trình chính bắt đầu tại đây, sau khi đã cài đặt hoàn tất.

|  |
| --- |
| while true  do      echo "-----------------------------------      echo "Hay thuc hien lua chon"      echo "1.Nhap thong tin vao file data"      echo "2.Xuat toan bo thong tin trong file data"      echo "3.Nhap vao he dieu hanh de biet dia chi IP"      echo "4.Nhap vao giao thuc de biet dia chi IP"      echo "5.Nhap vao giao thuc de biet so lan ket noi"      echo "6.Thuc hien thong ke"      echo "7.Ket thuc"      read chon      case $chon in          1)          clear          nhapthongtin;;          2)          clear          xuatthongtin;;          3)          clear          getIPfromOS;;          4) clear          getIPfromProtocol;;          5) clear          CountProtocol;;          6) clear          findOS          Info;;          7) break;;      esac  done |

Hình sau là giao diện chương trình:

|  |
| --- |
|  |