

# **Trường Đại học Văn Lang**

# **KHOA CÔNG NGHỆ THÔNG TIN**

**Tài liệu hướng dẫn học tập**  
**“Hệ điều hành và Lập trình Linux”**  
**(DIT0100)**

**Ngành Công nghệ thông tin (7480201)**  
**Học kỳ 2 - Năm học 2019 – 2020 (Khóa 24)**

**Bộ môn Khoa học Dữ liệu**  
**TP. HCM - 2020**

# NỘI DUNG

Chương 1 Giới thiệu và cài đặt Hệ điều hành Linux .....	7
1.1 Tổng quan .....	7
1.2 Lịch sử của Dự án GNU/Linux .....	8
1.3 Debian Linux.....	8
1.4 SuSE Linux .....	9
1.5 RedHat Linux .....	9
1.6 Ubuntu Linux .....	10
1.7 Cài đặt Hệ điều hành Linux .....	12
1.8 Cài đặt Ubuntu Desktop trên Desktop PC hoặc trong máy ảo trên PC.....	14
1.9 Cài đặt Ubuntu Bash Shell trên Windows 10 từ Microsoft Store .....	23
1.10 Cài đặt máy ảo (VM) Ubuntu Server trong điện toán đám mây Microsoft Azure.....	29
1.11 Câu hỏi ôn tập Chương 1 .....	42
Chương 2 Sử dụng Hệ điều hành Linux .....	53
2.1 Khởi động Linux .....	53
2.2 Cấu hình BIOS .....	53
2.3 Master Boot Record (MBR) và Boot Loader.....	53
2.4 Hoạt động của Boot Loader .....	54
2.5 Ô đĩa Ram khởi động (Initial RAM Disk) .....	55
2.6 Đăng nhập chế độ dòng lệnh.....	56
2.7 Nhân Linux .....	56
2.8 /sbin/init và các dịch vụ Hệ điều hành .....	57
2.9 Các quy trình khởi động Linux tương tự .....	57
2.10 Các tính năng của systemd.....	58
2.11 Tìm hiểu hệ thống tập tin và thư mục trên Linux.....	59
2.12 Câu hỏi ôn tập Chương 2 .....	60
Chương 3. Làm việc trên môi trường đồ họa Desktop Linux .....	64
3.1 Giới thiệu .....	64
3.2 X Window System .....	65
3.3 Khởi động giao diện đồ họa (GUI) .....	67
3.4 Tắt máy và khởi động lại Linux với giao diện đồ họa .....	73
3.5 Một số thao tác cơ bản với giao diện đồ họa trên Linux.....	76
3.6 Quản lý tập tin.....	78
3.7 Thự mục nhà (Home Directories) .....	79

3.8 Xem nội dung tập tin.....	81
3.9 Tìm kiếm tập tin.....	83
3.10 Chính sửa tập tin .....	83
3.11 Xóa tập tin .....	84
3.12 Thiết lập các tham số hệ thống chạy Linux bằng giao diện đồ họa .....	85
3.13 Cấu hình mạng .....	91
3.14 Cài đặt và nâng cấp các gói phần mềm .....	94
3.15 Kết nối với thiết bị và dịch vụ bên ngoài .....	97
3.16 Câu hỏi ôn tập Chương 3 .....	98
<b>Chương 4. Quản lý tiến trình trong Linux .....</b>	<b>99</b>
4.1 Tiến trình là gì ? .....	99
4.2 Phân loại tiến trình .....	99
4.3 Định thời gian và trạng thái của tiến trình .....	100
4.4 Định danh (ID) của tiến trình và chủ thể trạng thái (threat) .....	100
4.5 Kết thúc tiến trình .....	101
4.6 Định danh của người dùng và nhóm .....	101
4.7 Mức độ ưu tiên của tiến trình .....	102
4.8 Mức trung bình tải.....	102
4.9 Các tiến trình chạy nền và trên mặt.....	104
4.10 Quản lý các công việc (jobs).....	105
4.11 Cây tiến trình.....	107
4.12 Lệnh top .....	107
4.13 Lập kế hoạch cho các tiến trình trong tương lai.....	114
4.14 Câu hỏi ôn tập Chương 4 .....	116
<b>Chương 5. Quản lý hệ thống tập tin và soạn thảo tập tin trên Linux .....</b>	<b>118</b>
5.1 Giới thiệu hệ thống tập tin Linux .....	118
5.2 Phân vùng Linux .....	119
5.3 Các điểm kết nối của hệ thống tập tin .....	120
5.4 NFS - Hệ thống tập tin mạng .....	122
5.5 Thư mục nhà của người dùng .....	125
5.6 Thư mục /bin và /sbin .....	125
5.7 Hệ thống tập tin /proc.....	126
5.8 Thư mục /dev .....	126
5.9 Thư mục /var .....	127

5.10 Thư mục /etc .....	127
5.11 Thư mục /boot .....	128
5.12 Thư mục /lib và /lib64 .....	128
5.13 Các thư mục ổ đĩa tháo lắp rời: /media, /run, /mnt .....	128
5.14 Các thư mục khác dưới thư mục gốc / .....	129
5.15 So sánh các tập tin với lệnh diff.....	130
5.16 Sử dụng các tiện ích tập tin.....	131
5.17 Sao lưu dữ liệu trong Linux .....	132
5.18 Nén dữ liệu tập tin.....	133
5.19 Lệnh sao chép ổ đĩa (dd).....	135
5.20 Các trình soạn thảo văn bản text.....	137
5.21 Lệnh nano và gedit.....	139
5.22 Lệnh vi and emacs.....	142
5.23 Câu hỏi ôn tập Chương 5 .....	151
<b>Chương 6 Môi trường làm việc cho người dùng trên Linux.....</b>	<b>152</b>
6.1 Xác nhận người dùng hiện hữu trên Linux .....	152
6.2 Căn bản về người dùng và nhóm người dùng .....	154
6.3 Tài khoản root .....	158
6.4 Các biến môi trường.....	160
6.5 Gọi lại các lệnh đã thực thi .....	164
6.6 Quyền sở hữu tập tin và thư mục .....	166
6.7 Các lệnh tiện ích làm việc với văn bản và dòng lệnh.....	169
6.8 Sử dụng các lệnh sed và awk .....	173
6.9 Các lệnh thao tác với tập tin văn bản .....	178
6.10 Các biểu thức chính quy và mẫu tìm kiếm.....	183
6.11 Câu hỏi ôn tập Chương 6 .....	189
<b>Chương 7 Cấu hình và làm việc mạng với Linux .....</b>	<b>190</b>
7.1 Môi trường làm việc mạng trên Linux .....	190
7.2 Địa chỉ IP .....	190
7.3 Phân hoạch và cấp phát địa chỉ IP.....	194
7.4 Phân giải tên miền qua địa chỉ IP .....	195
7.5 Các tập tin cấu hình mạng.....	195
7.6 Các giao tiếp với thiết bị mạng .....	196
7.7 Các lệnh tiện ích làm việc mạng .....	197

7.8 Các trình duyệt dùng dòng lệnh (phi đồ họa).....	200
7.9 Các lệnh di chuyển tập tin.....	202
7.10 SSH - Thực thi lệnh trên Linux từ xa.....	203
7.11 Sao chép tập tin an toàn với scp.....	204
7.12 Câu hỏi ôn tập Chương 7 .....	204
<b>Chương 8 Lập trình Bash Shell.....</b>	<b>206</b>
8.1 Giới thiệu lập trình Shell Scripting .....	206
8.2 Các lựa chọn Command Shell.....	206
8.3 Shell Scripts .....	207
8.4 Giá trị trả lại từ Shell Script .....	210
8.5 Cú pháp cơ bản và ký tự đặc biệt.....	211
8.6 Các lệnh tích hợp trong Shell.....	214
8.7 Các lệnh được xây dựng để chạy Bash .....	216
8.8 Tham số tập lệnh .....	216
8.9 Biến môi trường .....	219
8.10 Định nghĩa hàm trong Shell Scripting.....	220
8.11 Kiểm tra điều kiện if .....	222
8.12 Lệnh kiểm tra tập tin test.....	224
8.13 Các biểu thức luận lý Boolean .....	225
8.15 Thao tác với các chuỗi .....	229
8.16 Biểu thức case .....	231
8.17 Cấu trúc vòng lặp .....	232
8.18 Gỡ lỗi bash Script.....	235
8.19 Tạo tập tin tạm thời và thư mục .....	237
8.20 Loại bỏ đầu ra với / dev / null .....	239
8.21 Số ngẫu nhiên và dữ liệu.....	239
8.22 Câu hỏi ôn tập Chương 8 .....	241
<b>Chương 9 Bảo mật Hệ điều hành Linux .....</b>	<b>243</b>
9.1 Tài khoản người dùng .....	243
9.2 Các kiểu tài khoản .....	244
9.3 Hiểu biết về tài khoản root.....	245
9.4 Các đặc trưng của lệnh sudo .....	246
9.5 Ghi nhật ký lệnh .....	248
9.6 Cách ly các tiến trình .....	249

9.7 Truy cập thiết bị phần cứng.....	250
9.8 Giữ cho Linux luôn cập nhật.....	251
9.9 Mật khẩu được lưu trữ như thế nào.....	252
9.10 Thuật toán mật khẩu.....	252
9.11 Yêu cầu đặt mật khẩu cho bộ tải khởi động .....	254
9.12 Các lỗ hổng trong Linux .....	254
9.13 Câu hỏi ôn tập Chương 9 .....	256

# Chương 1 Giới thiệu và cài đặt Hệ điều hành Linux

## 1.1 Tổng quan

Linux là phần mềm hệ điều hành cho máy tính (*Operating System*) dùng các máy chủ cũng như máy trạm, với nhiều gói phần mềm ứng dụng (*Application Software Packages*) như dịch vụ web, hệ quản trị cơ sở dữ liệu, dịch vụ thư tín điện tử, bảo mật tường lửa, dịch vụ thư mục LDAP và nhiều dịch vụ mạng khác nhau.

Có nhiều bản phân phối của Linux (*Linux Distribution*) được xây dựng bởi cộng đồng hoặc các công ty thương mại, với quy trình phát triển và đóng gói khác nhau, nhằm đơn giản hóa việc cài đặt nhân của Linux (*Linux Kernel*) cùng các ứng dụng liên quan bằng những công cụ quản trị khác nhau. Ứng dụng trên Linux có thể là các Phần mềm nguồn mở (*PMNM - OpenSource Software*) hoặc các phần mềm thương mại (*Commerce Software*) có bản quyền. Khách hàng sử dụng bản phân phối Linux do các công ty phát hành thường sẽ phải trả tiền cho các ứng dụng, tiện ích, chương trình được cài đặt sẵn với nhân Linux, cùng với tài liệu hướng dẫn sử dụng kèm theo.

Mục đích tạo ra các bản phân phối Linux là để giảm chi phí quản trị hệ điều hành Linux cho khách hàng, do nhân của Linux không bao gồm các ứng dụng dành riêng. Điều này có nghĩa là nhân Linux luôn tuân thủ giấy phép bản quyền phần mềm nguồn mở, nhưng các bản phân phối Linux và các ứng dụng kèm theo thì có thể là phần mềm thương mại.

Người dùng Linux khi lựa chọn cài đặt một bản phân phối sẽ nhận các đĩa CD cài đặt và tài liệu hướng dẫn kèm theo. Mặc dù số lượng và chất lượng của tài liệu là khác nhau đối với từng nhà phân phối, nhưng đều chứa hướng dẫn cài đặt và nhiều thông tin chi tiết liên quan tới sử dụng Linux. Linux tuân thủ Giấy phép Công cộng Chung (*GNU - General Public License - GPL*) thường áp dụng cho rất nhiều ứng dụng và cả nhân hệ điều hành Linux. Điều này cho phép người sử dụng có thể biên dịch lại phần mềm GNU GPL khi có vấn đề phát sinh, hay sửa đổi phần mềm theo yêu cầu riêng biệt của họ.

Các bản phân phối Linux khác nhau tồn tại dưới dạng các gói hoàn chỉnh (CD, tài liệu) có thể mua từ các đại lý (phải trả tiền), hoặc có thể tải về từ Internet (miễn phí). Các gói phần mềm mua từ các đại lý thường bao gồm cả các dịch vụ hỗ trợ từ nhà cung cấp và nó sẽ không có đối với các bản phân phối miễn phí tải về từ Internet.

Tính tương thích của các bản phân phối Linux và việc tiêu chuẩn hóa các bản phân phối Linux khác nhau là những vấn đề quan trọng. Để tránh các khác biệt không mong đợi giữa các bản phân phối riêng lẻ, tiêu chuẩn Phân bậc của hệ thống tập tin (*File System Hierarchy Standard*) luôn được xác định nhất quán cho mọi cấu trúc cây thư mục của Linux<sup>1</sup>. Các nhà cung cấp các bản phân phối Linux khác nhau sẽ triển khai tiêu chuẩn này trong các phiên bản của họ. Như một thành tố đảm bảo tính tương tác, tiêu chuẩn này cũng

---

<sup>1</sup> <http://www.pathname.com/fhs/>

tích hợp vào Nền tảng Chuẩn Linux (*Linux Standard Base - LSB*). Mục đích của LSB là xác định các tiêu chuẩn được thiết kế để đạt tới tính tương thích cao nhất của các bản phân phối và để tránh sự bất đồng, phân rã giữa các hệ thống sử dụng Linux.

Tiêu chí lựa chọn một bản phân phối Linux dựa trên những yêu cầu liên quan tới hỗ trợ và quản trị, cũng như khả năng vận hành trên các phần cứng khác nhau. Một lý do khác cho sự lựa chọn bản phân phối Linux là quy mô của Công ty hỗ trợ, khả năng tài chính, tính sẵn sàng trong tuân thủ thỏa thuận khung hoặc khả năng tùy biến ứng dụng theo yêu cầu của người sử dụng, doanh nghiệp và các cơ quan chính phủ.

## 1.2 Lịch sử của Dự án GNU/Linux

Nhân Linux là trái tim của hệ điều hành Linux. Nhân là bộ phận quan trọng của bất kỳ hệ điều hành nào, vì Nhân cung cấp cầu nối giữa phần cứng và phần mềm ứng dụng.

Linux được ra đời năm 1991 bởi một sinh viên Phần Lan tên là *Linus Torvalds*. Lúc đầu, Linux chỉ hoạt động với các máy tính có CPU i386 và được coi như một bản sao chép của UNIX, nhằm khai thác thế mạnh của kiến trúc i386 mới ra đời khi đó. Tuy nhiên đến ngày nay, nhờ nỗ lực của cộng đồng các nhà phát triển và người sử dụng trên toàn thế giới, Linux đã trở thành một Hệ điều hành máy tính rất phổ biến, có thể hoạt động trên mọi nền tảng phần cứng với cấu hình hiện đại nhất. Hơn nữa, Nhân Linux đã đạt tới tầm quan trọng mang tính ý thức hệ cũng như kỹ thuật. Có cả một cộng đồng đang tin vào các lý tưởng của Phần mềm tự do và Nguồn mở (FOSS), đang tích cực công hiến thời gian và trí tuệ của mình để cải tiến các phần mềm FOSS. Chính những người thuộc cộng đồng FOSS nói trên là cơ sở để hình thành các dự án lớn như *Ubuntu Linux*, Ủy ban chuẩn hoá định hình cho sự phát triển của mạng Internet (*Internet Task Force*), các tổ chức nghề nghiệp như *Mozilla Foundation* với trình duyệt web nổi tiếng *Firefox*, cũng như hàng loạt các dự án phát triển phần mềm khác mà người dùng được hưởng lợi ích.

## 1.3 Debian Linux

*Debian Linux* là dự án do một nhóm các nhà lập trình làm việc với nhau để cùng phát triển hệ điều hành tự do nguồn mở Linux. Nhóm tụ tập khoảng 1000 thành viên có tên tuổi trên toàn thế giới. Tính năng quan trọng nhất của *Debian Linux* là sự tuân thủ hoàn toàn giấy phép GPLv2. Do đó *Debian Linux* có thể được sao chép và được sử dụng với mục đích thương mại mà không có bất cứ sự hạn chế nào.

*Debian Linux* có thể tải miễn phí từ mạng Internet, hoặc có thể dùng đĩa cài đặt mua từ các đại lý (mặc dù *Debian Linux* là một sản phẩm không mang tính thương mại). Việc sửa lỗi *Debian Linux* được quản lý chặt chẽ bởi nhóm phát triển. Cơ chế đảm bảo chất lượng này đã tạo cho *Debian Linux* có tính ổn định rất cao.

*Ubuntu Linux* là một biến thể của bản phân phối *Debian Linux*, đã và đang phát triển rất mạnh và là chủ đề sẽ được trình bày trong các phần

## 1.4 SuSE Linux

*SuSE Linux AG* là một trong những nhà cung cấp quốc tế lớn nhất bản phân phối Linux, chiếm thị trường lớn tại Đức. *SuSE Linux* đã phát triển nhiều sản phẩm với các ứng dụng khác nhau. *SuSE Linux* cũng dựa trên chuẩn về hệ thống quản trị gói RPM (Red Hat Packet Management) được phát triển bởi Công ty Red Hat. Hệ thống đóng gói theo RPM cho phép người dùng dễ dàng cài đặt và tháo gỡ cài đặt các phần mềm, kể cả các gói phần mềm được cung cấp từ các công ty khác nhau. *SuSE Linux* chuẩn bao gồm hệ quản trị và phần mềm cài đặt được tích hợp chặt chẽ với nhau.

Các sản phẩm SUSE Linux	Các ứng dụng chủ chốt
Personal - Professional	Được nhà sản xuất khuyến cáo đầu tiên cho các ứng dụng máy để bàn. Các phiên bản đi kèm với các gói phần mềm có thể cài đặt được trên các máy tính một cách dễ dàng. Phiên bản chuyên nghiệp Professional đi kèm vô số thành phần máy chủ trên các đĩa CD, phù hợp với việc sử dụng trong các doanh nghiệp.
Enterprise	Các máy chủ SuSE Linux là các hệ điều hành được thiết kế để sử dụng trong các môi trường CNTT với mọi phạm vi và mọi lĩnh vực. Hỗ trợ các nền tảng phần cứng khác nhau như các vi xử lý AMD và Intel 32-bit và 64-bit cũng như các dãy máy chủ eServer của IBM, kể cả các máy tính lớn Mainframe.

## 1.5 RedHat Linux

*Red Hat Linux* là một bản phân phối thương mại của Linux, với các phương án (phiên bản) khác nhau cho khách hàng khác nhau. *Red Hat Linux* sử dụng hệ thống quản trị ứng dụng dành riêng (được phát triển trong nội bộ công ty) là RPM (RedHat Package Management) cho phép quản lý theo cách duy nhất và gần gũi với người sử dụng.

Các sản phẩm RedHat Linux	Các ứng dụng chủ chốt
Fedora Linux / CentOS	Đây là các bản phân phối PMNM tương thích với Red Hat, được thiết kế ban đầu cho các giải pháp máy trạm và / hoặc máy chủ trong các môi trường nhỏ lẻ, không được hỗ trợ bởi Công ty RedHat.

RedHat Enterprise Linux	Giải pháp doanh nghiệp được chào chủ yếu cho các doanh nghiệp. Các hệ thống RHEL được trao quyền cho các nền tảng khác nhau của các nhà sản xuất phần cứng khác nhau bao gồm cả các công nghệ máy chủ clustering.
-------------------------	---

Khác biệt chính giữa các bản phân phối khác nhau là các ứng dụng khác nhau được cài đặt ngầm định, cũng như các khuyến cáo cùng những khác biệt về phạm vi hỗ trợ được mời chào trong các thỏa thuận về giấy phép sử dụng và giá sản phẩm.

## 1.6 Ubuntu Linux

### Đặc điểm chung

*Ubuntu Linux* là hệ điều hành Linux cho máy tính theo triết lý PMNM, được xây dựng dựa trên nhân của GNU/Linux bởi **Công ty Canonical Ltd.** (có trụ sở tại Nam Phi). Cộng đồng người dùng Ubuntu được hình thành bởi những lý tưởng gắn kèm với triết lý *Ubuntu* (*Ubuntu Philosophy*) đó là: người dùng được *tự do sử dụng phần mềm miễn phí*, mỗi một phần mềm đều có thể sử dụng thông qua *giao diện ngôn ngữ bản địa* của người dùng, và quan trọng nhất là *người dùng hoàn toàn có quyền tự do chỉnh sửa và thay đổi* phần mềm để phù hợp với nhu cầu sử dụng riêng của mình. Vì những lý do đã nêu trên đây, Ubuntu sẽ được phân phối miễn phí mãi mãi, người dùng không phải trả thêm bất kỳ chi phí nào, ngay cả với phiên bản doanh nghiệp (*Ubuntu Linux Enterprise Edition* – chi phí hỗ trợ người dùng là doanh nghiệp).

*Ubuntu Linux* bao gồm các bản phiên dịch tốt nhất, tạo ra một cơ sở hạ tầng truy cập tốt nhất (theo khả năng cung cấp của cộng đồng lập trình phần mềm tự do), nhằm mục đích cho phép nhiều người có thể tiếp cận và sử dụng *Ubuntu Linux*.

Mỗi 6 tháng, một phiên bản *Ubuntu* mới sẽ được công bố. Mỗi phiên bản sẽ được hỗ trợ tối thiểu trong vòng 18 tháng. Và cứ mỗi 02 năm một lần, *Ubuntu* sẽ phát hành phiên bản ổn định LTS (Long Term Support) hỗ trợ 05 năm. Tùy vào mục đích nhu cầu sử dụng, chúng ta có thể dùng phiên bản *Ubuntu* ổn định hoặc phiên bản đang phát triển.

*Ubuntu* hoàn toàn tuân theo những nguyên tắc phát triển phần mềm mã nguồn mở và khuyến khích mọi người dùng phần mềm mã nguồn mở, cải tiến chúng và phân phối lại cho những người khác.

Công ty **Canonical** do công tước **Mark Shuttleworth**, người khách du hành đầu tiên đã trả 20 triệu USD để bay vào vũ trụ, thành lập. Công ty Canonical hỗ trợ thương mại cho dự án nguồn mở *Ubuntu* và cung cấp các dịch vụ liên quan, cũng như tài trợ cho một số dự án phần mềm nguồn mở khác.

### Cộng đồng Ubuntu

Cộng đồng Ubuntu bao gồm người phát triển, lập trình viên, người thử nghiệm, người soạn tài liệu kỹ thuật, người dùng thử các tài liệu kỹ thuật, người dịch và, quan trọng nhất, những người dùng Ubuntu hàng ngày. Để cộng đồng này càng lớn mạnh, việc đầu tiên mà bạn cần làm đó là sử dụng và giới thiệu chúng cho gia đình, bạn bè và đồng nghiệp của bạn. Bạn có thể tham khảo thêm thông tin trên trang web của Ubuntu: <http://www.ubuntu.com/>.

## Ý nghĩa tên gọi Ubuntu

*Ubuntu* là tên của một lý tưởng đạo đức Nam Phi tập trung vào các quan hệ và sự gắn kết giữa con người. Từ Ubuntu xuất phát từ ngôn ngữ thổ dân Zulu và Xhosa. Ubuntu được coi như một khái niệm truyền thống của Châu Phi, là một cơ sở cho sự hình thành của Tân Cộng hoà Nam Phi và có gắn liền với sự Phục hưng Châu Phi.

Dịch và tóm lại ý nghĩa của từ Ubuntu ta có thể ghi tạm “tính nhân bản cho mọi người” (*humanity towards others*), hoặc “niềm tin vào mối quan hệ toàn cầu để chia sẻ tất cả những gì là chung của loài người” (*the belief in a universal bond of sharing that connects all humanity*).

Vậy, dựa trên GNU/Linux, hệ điều hành Ubuntu mang theo tư tượng tự do và nhân bản đã ẩn sâu trong từ Ubuntu cho cả thế giới phần mềm.

## Giấy phép bản quyền Ubuntu

Hệ điều hành Ubuntu được phân phối theo giấy phép phần mềm tự do (GNU/GPL). Dự án Ubuntu hoàn toàn tuân theo những nguyên tắc phát triển phần mềm mã nguồn mở. Ý nghĩa ở đây là Ubuntu sẽ luôn là một phần mềm miễn phí.

Tuy nhiên, không chỉ có ý nghĩa là dùng Ubuntu sẽ không phải mất một đồng xu nào hết. Thiên hướng của Phần mềm tự do là mỗi người đều phải có tự do dùng phần mềm theo mọi cách “có ích cho xã hội”. “Phần mềm tự do” không chỉ có ý nghĩa là không phải mất tiền để dùng chúng, chúng còn có ý nghĩa là mỗi người có thể dùng chúng theo mọi yêu cầu: Phần mềm tự do có mã nguồn mở, cho phép mỗi người khả năng tải mã nguồn về, sửa đổi lại mã nguồn gốc, phù hợp cho bất kỳ yêu cầu chính đáng. Vì vậy, ngoài ra sự thuận lợi do đa số các phần mềm tự do là miễn phí, còn có thêm những ưu điểm về mặt kỹ thuật: khi cần phát triển phần mềm, chúng ta có thể tái sử dụng các công trình của những lập trình viên khác. Điều đó không thể nào có đối với các phần mềm có mã nguồn đóng: mỗi lần cần phát triển một phần mềm phải bắt đầu từ số không. Vì sự khác nhau đó, việc phát triển Phần mềm tự do vừa nhanh, vừa có hiệu quả cao và mang tính kích thích.

Bạn có thể tìm hiểu thêm về triết lý của Phần mềm tự do trên trang web <http://www.gnu.org/philosophy/>.

## Sự khác biệt của Ubuntu

Có rất nhiều hệ điều hành máy tính dựa trên nhân GNU/Linux: như Debian, SuSE, RedHat... Ubuntu cũng chỉ là một nhà cung cấp trong một thế giới đầy cạnh tranh. Vậy cái gì gây nên sự khác biệt cho Ubuntu ?

Dựa trên Debian, một trong những bản phân phối Linux được ca ngợi nhiều nhất, có tính kỹ thuật tiên tiến và được hỗ trợ tốt, Ubuntu nhằm tạo ra một bản phân phối Linux cập nhật và ổn định dành cho cả máy để bàn và máy chủ. Bắt đầu từ phiên bản 7.10, Ubuntu còn hỗ trợ cả các thiết bị (embedded devices). Ubuntu bao gồm các gói phần mềm đã được chọn lọc rất kỹ trong kho các gói phần mềm của bản phân phối Debian, giữ lại hệ thống quản lý gói phần mềm mạnh mẽ của Debian là Synaptic, cho phép cài đặt và gỡ bỏ các phần mềm ứng dụng một cách dễ dàng và sạch sẽ. Không giống các bản phân phối Linux khác đi kèm theo một số lượng rất lớn các phần mềm có thể dùng có ích hay không có ích, danh mục các gói phần mềm của Ubuntu chỉ bao gồm những phần mềm ứng dụng quan trọng và có chất lượng cao.

Nhờ sự tập trung vào chất lượng, Ubuntu tạo ra một môi trường ổn định và nhiều tính năng, thích hợp cho cả môi trường làm việc ở nhà cũng như ở nơi công sở. Dự án Ubuntu tập trung các nỗ lực để hoàn chỉnh mọi chi tiết phần mềm, cho phép công bố một phiên bản mới, mỗi 6 tháng, kèm theo phiên bản mới nhất của các phần mềm ứng dụng nổi tiếng. Ubuntu được phát triển để dùng cùng các cấu hình máy tính dựa trên các dòng CPU của Intel, AMD, PowerPC,... cũng như CPU trên các thiết bị nhúng khác (ARM)

## **Phương thức đánh số các bản phân phối Ubuntu**

Cơ chế đánh số cho các phiên bản Ubuntu dựa vào thời điểm công bố phiên bản của bản phân phối (cách này giống như các công ty bán xe hơi đưa ra các mẫu mã mới hàng năm). Số phiên bản xuất phát từ năm và tháng của bản phân phối chứ không phải là phiên bản thật của phần mềm. Phiên bản Ubuntu đầu tiên, mang tên *Warty Warthog*, được công bố vào tháng 10 năm 2004, do đó được đặt phiên bản số “4.10”. Phiên bản Ubuntu phổ biến nhất hiện nay ra đời tháng 04 năm 2018, mang tên *Bionic Beaver* và có số hiệu “**18.04 LTS**” (LTS - Long Time Support).

### **1.7 Cài đặt Hệ điều hành Linux**

#### **Nguyên tắc đầu tiên: lập kế hoạch phân vùng ổ đĩa**

Linux là hệ điều hành cho máy tính, cần được cài đặt trên ổ đĩa cứng với bộ trí phân vùng thích hợp. Bộ cục phân vùng cần được quyết định tại thời điểm cài đặt; nó có thể khó thay đổi sau này. Mặc dù các hệ thống Linux xử lý nhiều phân vùng ổ đĩa bằng cách gắn chúng tại các điểm cụ thể trong hệ thống tệp và bạn luôn có thể sửa đổi thiết kế sau đó, nhưng luôn dễ dàng hơn để thử và bắt đầu ngay.

Gần như tất cả các trình cài đặt cung cấp bộ cục mặc định hợp lý, với tất cả không gian dành riêng cho các tệp bình thường trên một phân vùng lớn và phân vùng trao đổi nhỏ hơn hoặc với các phân vùng riêng cho một số khu vực nhạy cảm với không gian như nơi lưu trữ thư mục và tài nguyên của người dùng, các tập tin ghi nháp.v.v... Nhìn chung, cấu hình cài đặt mặc định (default) của Linux trong hầu hết các bản phân phối là phù hợp với nhu cầu của đa số người dùng, bạn vẫn có thể cần ghi đè lên mặc định hoặc làm một cái gì đó khác, nếu bạn có nhu cầu đặc biệt hoặc nếu bạn muốn sử dụng nhiều hơn một ổ đĩa.

### **Nguyên tắc thứ hai: lựa chọn bản phân phối và các gói phần mềm ứng dụng Linux**

Tất cả các bản phân phối Linux đều kèm theo tiện ích cài đặt, bao gồm nhân Linux và các gói phần mềm tối thiểu để chạy Linux.

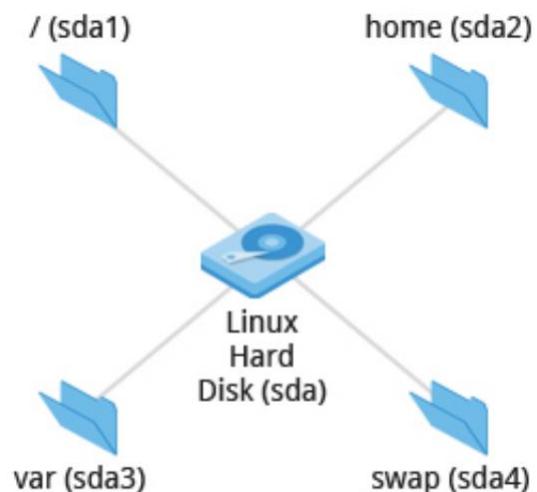
Hầu hết các trình cài đặt cũng cung cấp các tùy chọn để thêm các danh mục phần mềm. Các ứng dụng phổ biến (như trình duyệt web Firefox và bộ ứng dụng văn phòng LibreOffice), các công cụ dành cho nhà phát triển (như trình soạn thảo văn bản vi và emacs, chúng ta sẽ khám phá sau trong khóa học này) và các dịch vụ phổ biến khác, (như các công cụ máy chủ web Apache hoặc Cơ sở dữ liệu MySQL) thường được bao gồm. Ngoài ra, đối với bất kỳ hệ thống nào có máy tính để bàn đồ họa, giao diện đồ họa được chọn thường là như Gnome hoặc KDE và sẽ được cài đặt mặc định.

Tất cả các trình cài đặt sẽ thiết lập một số tính năng bảo mật ban đầu trên hệ thống mới. Một bước cơ bản đầu tiên thường là thiết lập mật khẩu cho siêu người dùng (root), hoặc người dùng ban đầu có quyền sudo qua root. Trong một số trường hợp (như Ubuntu), chỉ người dùng ban đầu được thiết lập; đăng nhập root trực tiếp không được cấu hình và truy cập root yêu cầu đăng nhập trước như một người dùng bình thường và sau đó sử dụng sudo, như chúng tôi sẽ mô tả sau. Một số bản phân phối cũng sẽ cài đặt các khung bảo mật nâng cao hơn, chẳng hạn như SELinux hoặc AppArmor. Ví dụ: tất cả các hệ thống dựa trên Red Hat bao gồm Fedora và CentOS luôn sử dụng SELinux theo mặc định và Ubuntu đi kèm với AppArmor và chạy.

### **Nguyên tắc thứ ba: tuân thủ quy trình cài đặt**

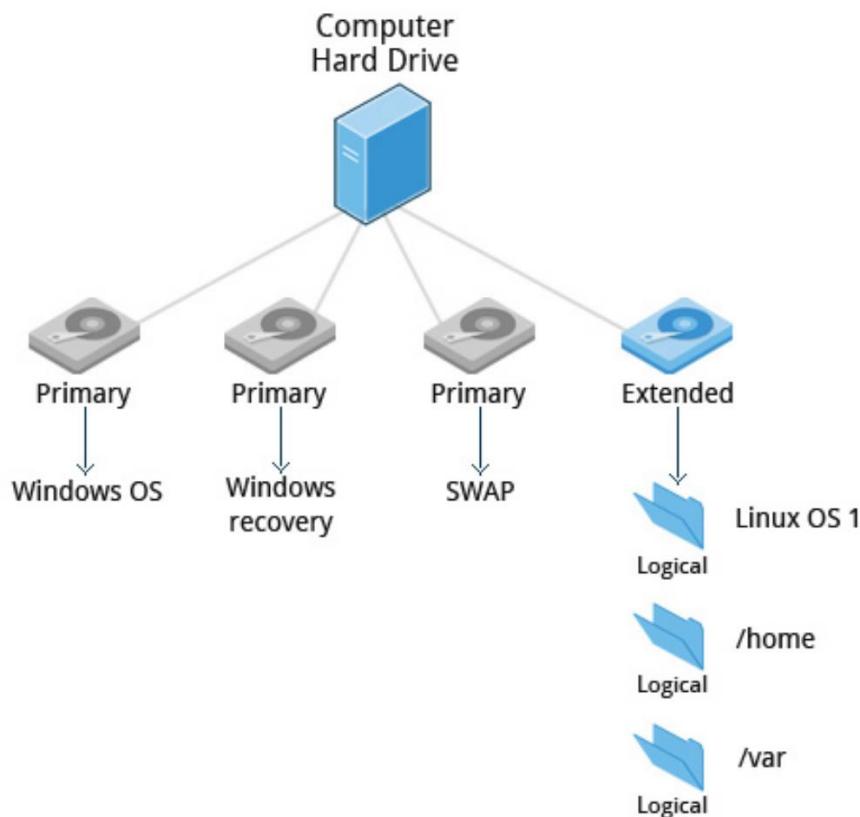
Quy trình cài đặt thực tế của hầu hết các bản phân phối và các môi trường vận hành Linux là khá giống nhau. Sau khi được khởi động từ phương tiện DVD/USB cài đặt hoặc môi trường cài đặt (Windows Subsystem hay điện toán đám mây), trình cài đặt sẽ khởi động

**Partitions in the Linux Hard Disk**



và đưa ra câu hỏi về cách thiết lập hệ thống. Những câu hỏi này có thể được bỏ qua, nếu có một tập tin cấu hình tự động được cung cấp. Sau đó, việc cài đặt được thực hiện tuần tự theo các bước. Cuối cùng, máy tính sẽ khởi động lại để người dùng có thể vào hệ điều hành Linux mới được cài đặt.

Trên một số bản phân phối, các câu hỏi bổ sung được hỏi sau khi hệ thống khởi động lại. Hầu hết các trình cài đặt có tùy chọn tải xuống và cài đặt các bản cập nhật như là một phần của quá trình cài đặt; điều này đòi hỏi truy cập Internet. Mặt khác, hệ thống sử dụng cơ chế cập nhật thông thường để lấy các bản cập nhật đó sau khi cài đặt xong.

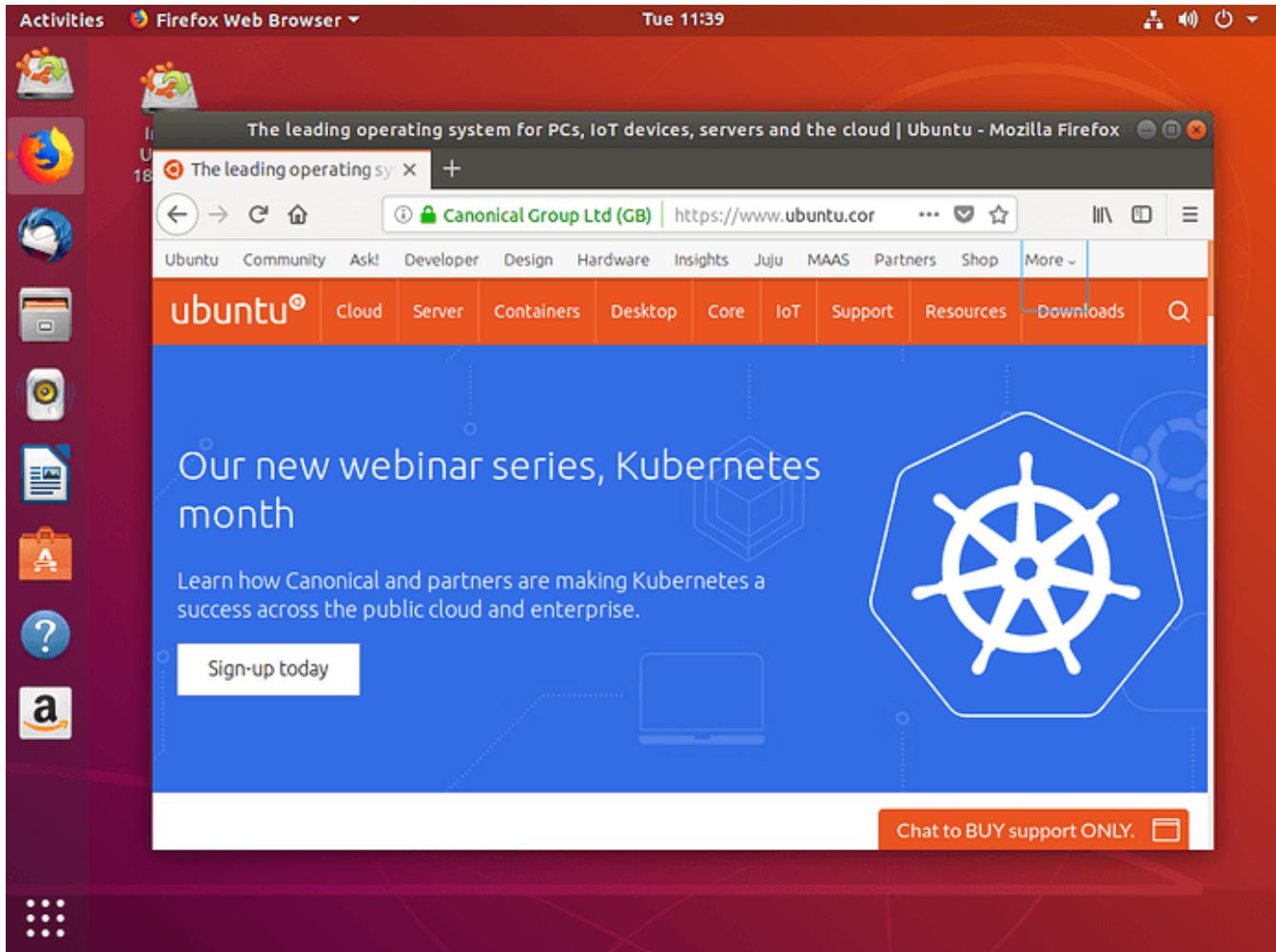


### Linux Installation Software Choices

## 1.8 Cài đặt Ubuntu Desktop trên Desktop PC hoặc trong máy ảo trên PC

### Giới thiệu

Máy tính để bàn Ubuntu dễ sử dụng, dễ cài đặt và bao gồm mọi thứ bạn cần để chạy tổ chức, trường học, nhà hoặc doanh nghiệp của bạn. Nó cũng mã nguồn mở, an toàn, có thể truy cập và miễn phí để tải về.



Trong hướng dẫn này, chúng ta sẽ cài đặt máy tính để bàn Ubuntu vào máy tính cá nhân hoặc trong máy ảo (Virtual Box), sử dụng ổ đĩa DVD trên máy tính hoặc ổ đĩa USB.

## Các yêu cầu cài đặt

Bạn cần phải xem xét những điều sau đây trước khi bắt đầu cài đặt:

- Kết nối máy tính xách tay của bạn với một nguồn năng lượng.
- Đảm bảo bạn có ít nhất 25 GB dung lượng lưu trữ miễn phí hoặc 5 GB cho cài đặt tối thiểu.
- Có quyền truy cập vào ổ đĩa DVD hoặc USB có chứa phiên bản Ubuntu bạn muốn cài đặt.

- Hãy chắc chắn rằng bạn có một bản sao lưu dữ liệu gần đây. Mặc dù nó không chắc chắn rằng bất cứ điều gì sẽ đi sai, bạn không bao giờ có thể quá chuẩn bị.

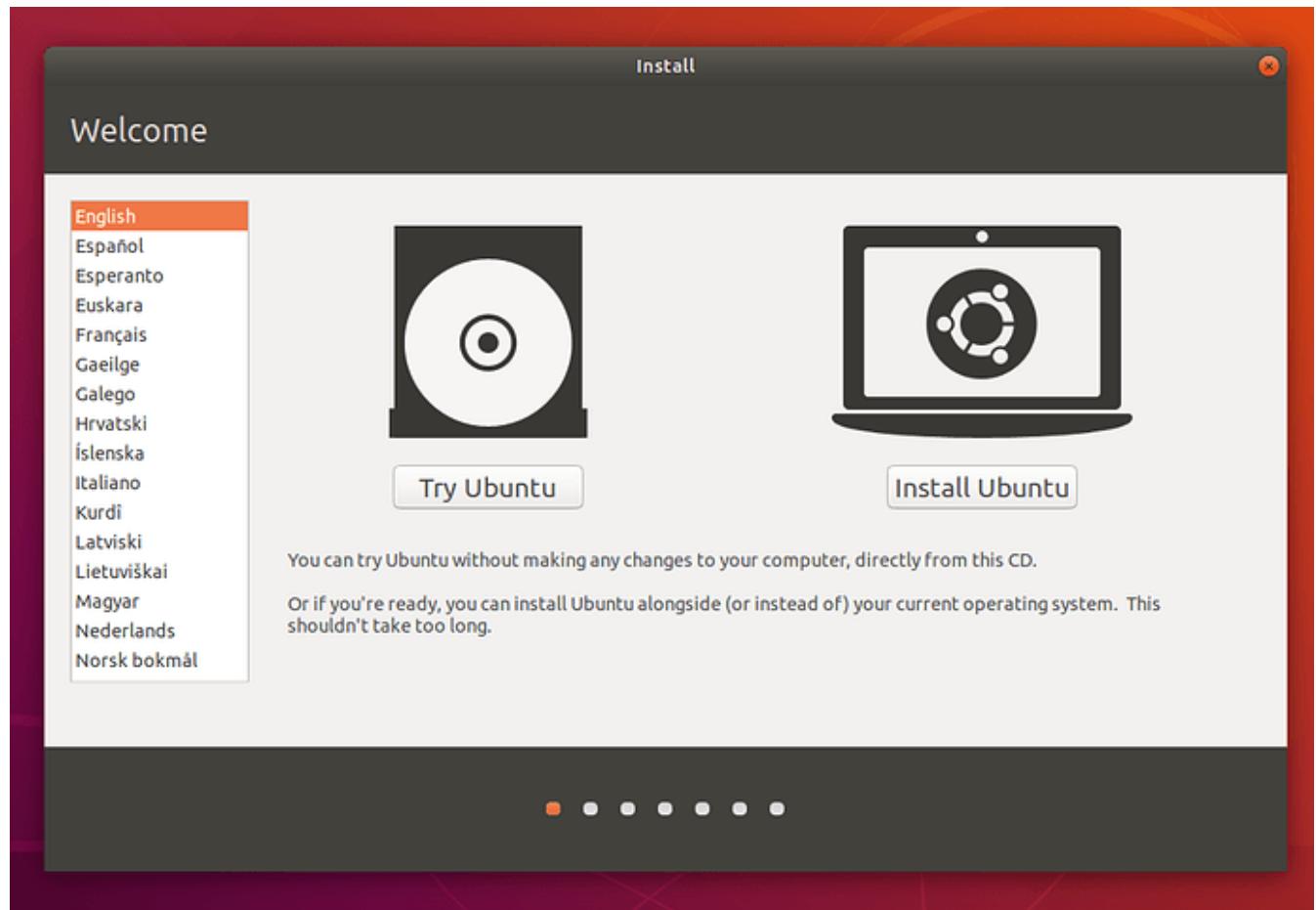
Xem Yêu cầu cài đặt / hệ thống để biết thêm chi tiết cụ thể về yêu cầu phần cứng. Chúng tôi cũng có một số hướng dẫn giải thích cách tạo ổ đĩa flash Ubuntu DVD hoặc USB.

## **Khởi động máy tính từ ổ đĩa DVD**

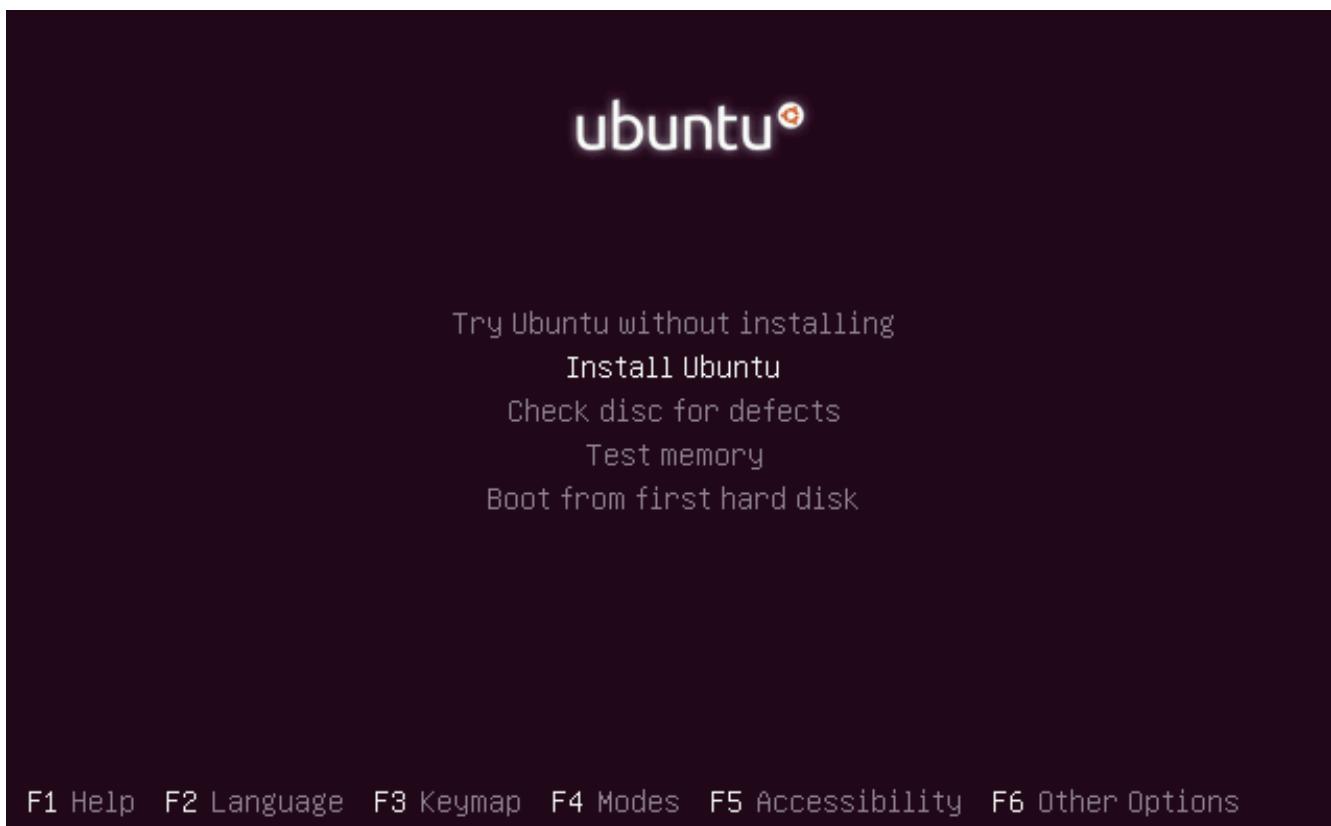
Sẽ dễ dàng cài đặt Ubuntu từ DVD. Đây là những gì bạn cần làm:

1. Đặt Ubuntu DVD vào ổ đĩa quang / DVD của bạn.
2. Khởi động lại máy tính của bạn.

Ngay khi máy tính của bạn khởi động, bạn sẽ thấy cửa sổ chào mừng.



Từ đây, bạn có thể chọn ngôn ngữ của mình từ danh sách bên trái và chọn giữa cài đặt trực tiếp Ubuntu hoặc thử máy tính để bàn trước (nếu bạn thích những gì bạn thấy, bạn cũng có thể cài đặt Ubuntu từ chế độ này). Tùy thuộc vào cấu hình máy tính của bạn, thay vào đó, bạn có thể thấy một menu khởi động thay thế hiển thị ngăn chọn ngôn ngữ lớn. Sử dụng chuột hoặc phím con trỏ để chọn ngôn ngữ và bạn sẽ được trình bày với một menu đơn giản.



Chọn tùy chọn thứ hai, ‘Cài đặt Ubuntu, và nhấn return để tự động khởi chạy trình cài đặt máy tính để bàn. Ngoài ra, chọn tùy chọn đầu tiên, ‘Thử Ubuntu mà không cần cài đặt, để kiểm tra Ubuntu (như trước đây, bạn cũng có thể cài đặt Ubuntu từ chế độ này).

Một lát sau, sau khi máy tính để bàn được tải, bạn sẽ thấy cửa sổ chào mừng. Từ đây, bạn có thể chọn ngôn ngữ của mình từ danh sách bên trái và chọn giữa cài đặt trực tiếp Ubuntu hoặc thử máy tính để bàn trước.

Nếu bạn không nhận được một trong hai menu, hãy đọc phần khởi động từ hướng dẫn DVD để biết thêm thông tin.

### **Khởi động máy tính từ ổ đĩa USB**

Hầu hết các máy tính sẽ tự động khởi động từ USB. Chỉ cần lắp ổ flash USB và bật nguồn máy tính của bạn hoặc khởi động lại nó. Bạn sẽ thấy cùng một cửa sổ chào mừng mà chúng ta đã thấy trong phần trước ‘Cài đặt từ DVD DVD, nhắc bạn chọn ngôn ngữ của mình và cài đặt hoặc thử máy tính để bàn Ubuntu.

Nếu máy tính của bạn không tự động khởi động từ USB, hãy thử giữ F12 khi máy tính của bạn khởi động lần đầu tiên. Với hầu hết các máy, điều này sẽ cho phép bạn chọn thiết bị USB từ menu khởi động dành riêng cho hệ thống.

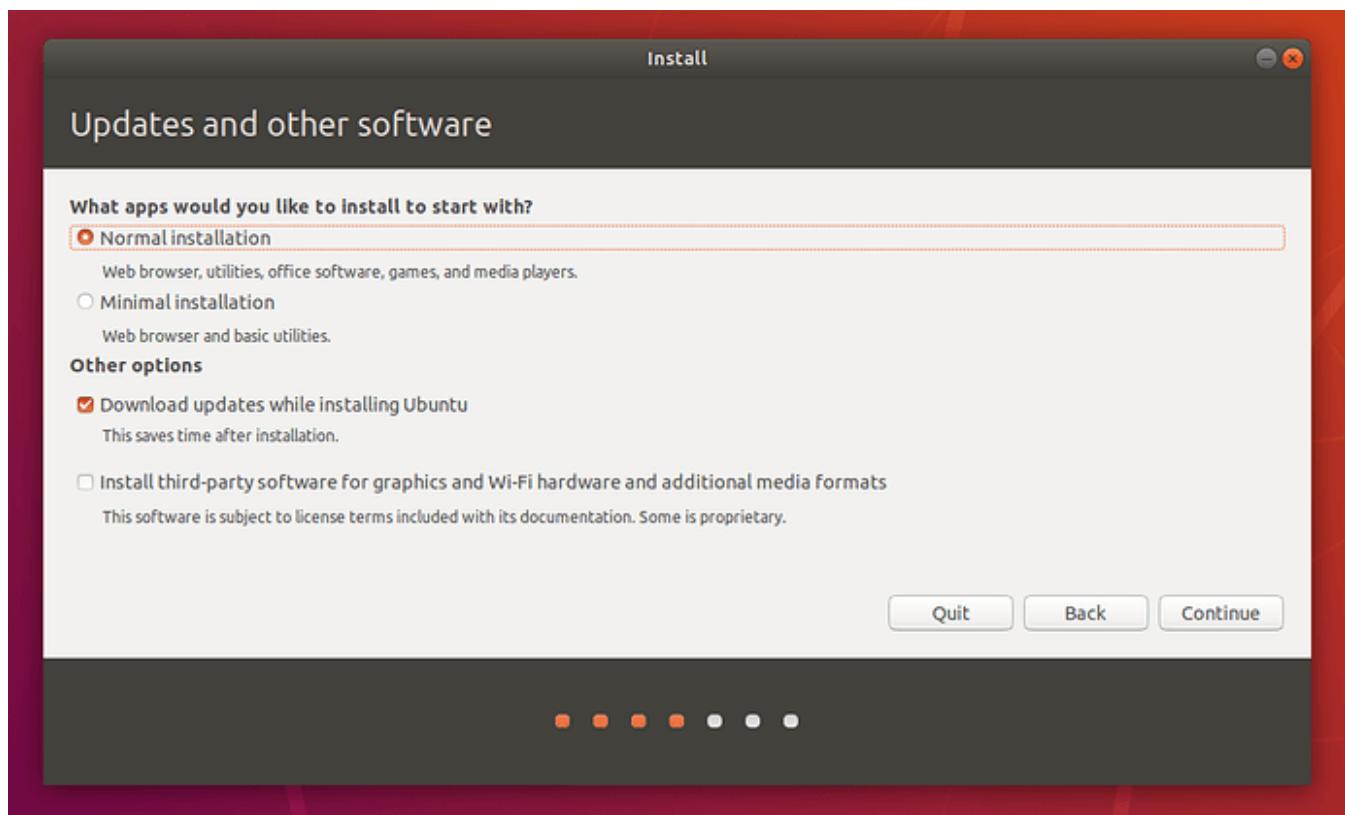
F12 là khóa phím biên nhất để hiển thị menu khởi động hệ thống của bạn, nhưng Escape, F2 và F10 là các lựa chọn thay thế phím biên. Nếu bạn không chắc chắn, hãy tìm một thông

báo ngắn gọn khi hệ thống của bạn khởi động - điều này thường sẽ thông báo cho bạn về phím nào cần nhấn để hiển thị menu khởi động.

## Chuẩn bị cài đặt Ubuntu

Trước tiên bạn sẽ được yêu cầu chọn bộ trí bàn phím của bạn. Nếu trình cài đặt không đoán đúng bộ cục mặc định, hãy sử dụng nút ‘Phát hiện bàn phím Bố cục bàn phím để chạy qua một quy trình cấu hình ngắn gọn.

Sau khi chọn Tiếp tục, bạn sẽ được hỏi Bạn muốn cài đặt ứng dụng nào để bắt đầu? Hai tùy chọn là installation Cài đặt bình thường và cài đặt tối thiểu. Đầu tiên là tương đương với gói tiện ích, ứng dụng, trò chơi và trình phát phương tiện mặc định cũ - một bệ phóng tuyệt vời cho mọi cài đặt Linux. Thứ hai chiếm ít không gian lưu trữ hơn và cho phép bạn chỉ cài đặt những gì bạn cần.



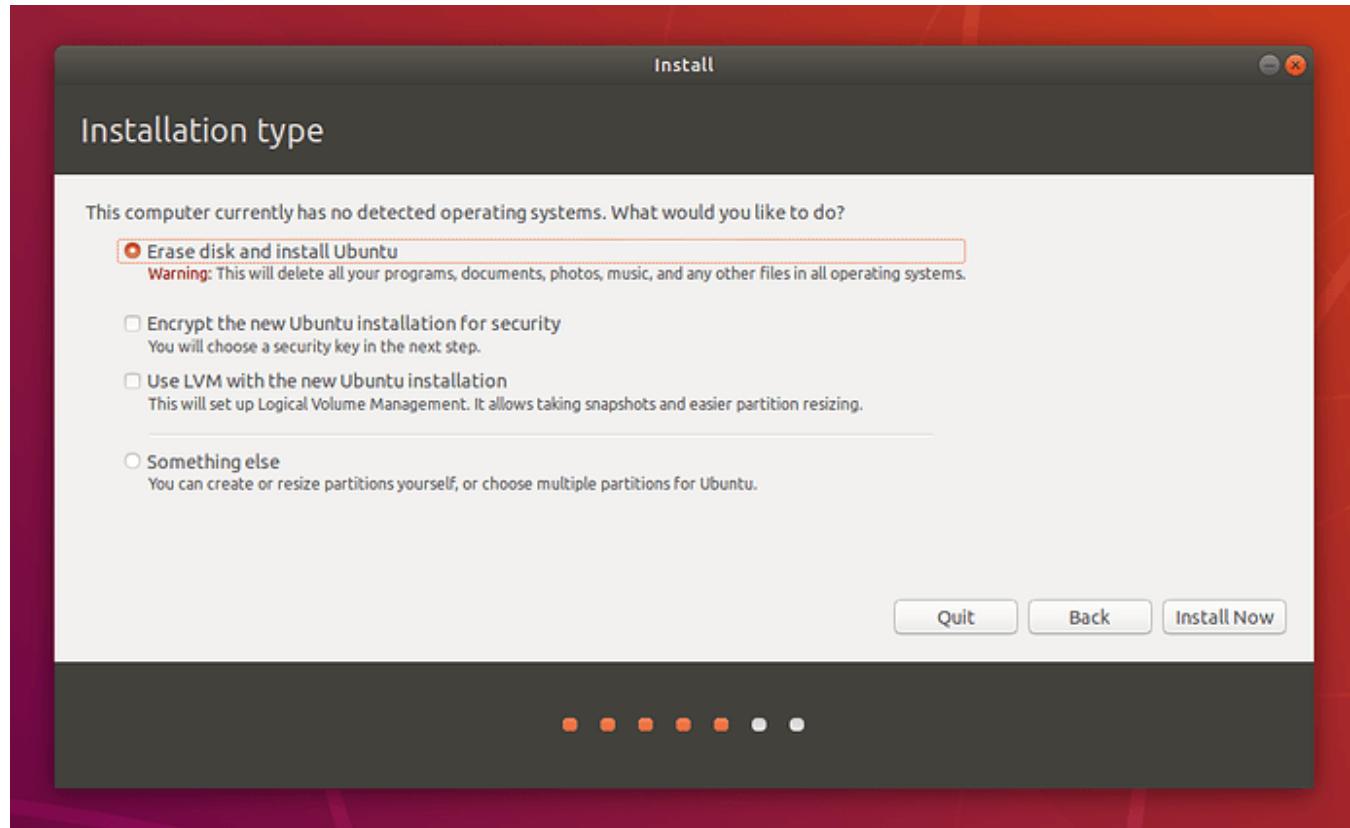
Bên dưới câu hỏi kiểu cài đặt là hai hộp kiểm; một để kích hoạt cập nhật trong khi cài đặt và một để kích hoạt phần mềm của bên thứ ba.

- Chúng tôi khuyên bạn nên bật cả Tải xuống bản cập nhật và Cài đặt phần mềm của bên thứ ba.
- Luôn kết nối với internet để bạn có thể nhận được các bản cập nhật mới nhất trong khi cài đặt Ubuntu.

- Nếu bạn không kết nối với internet, bạn sẽ được yêu cầu chọn một mạng không dây, nếu có. Chúng tôi khuyên bạn nên kết nối trong khi cài đặt để chúng tôi có thể đảm bảo máy của bạn được cập nhật

## Chuẩn bị không gian trên ổ đĩa

Sử dụng các hộp kiểm để chọn xem bạn có muốn cài đặt Ubuntu cùng với hệ điều hành khác không, xóa hệ điều hành hiện tại của bạn và thay thế nó bằng Ubuntu hoặc - nếu bạn là một người dùng nâng cao - hãy chọn tùy chọn Một cái gì đó khác.

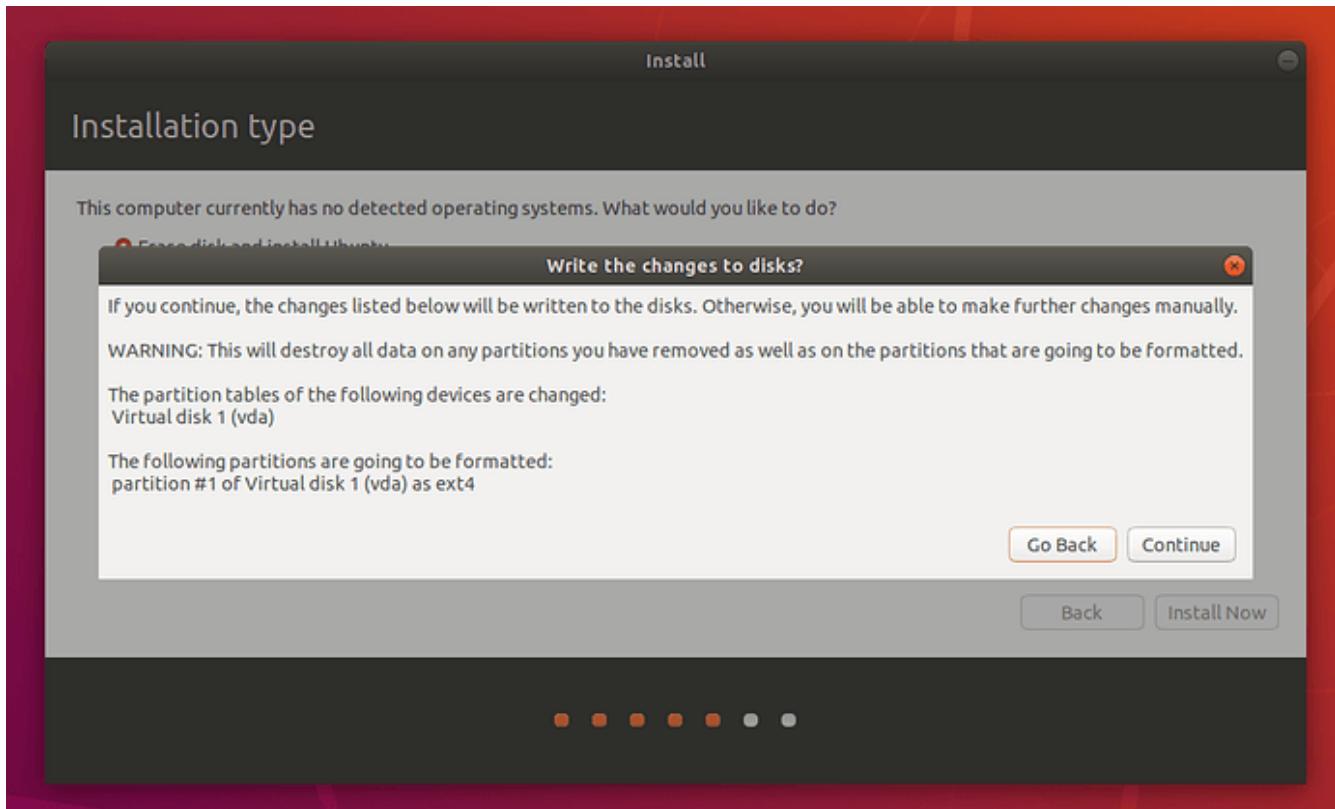


Các tùy chọn liên quan đến cài đặt song song hoặc xóa cài đặt trước đó chỉ được cung cấp khi phát hiện các cài đặt có sẵn.

## Bắt đầu cài đặt

Sau khi định cấu hình lưu trữ, nhấp vào nút ‘Cài đặt ngay bây giờ’. Một khung nhỏ sẽ xuất hiện với tổng quan về các tùy chọn lưu trữ mà bạn đã chọn, với cơ hội quay lại nếu chi tiết không chính xác.

Nhấp vào Tiếp tục để khắc phục những thay đổi tại chỗ và bắt đầu quá trình cài đặt.

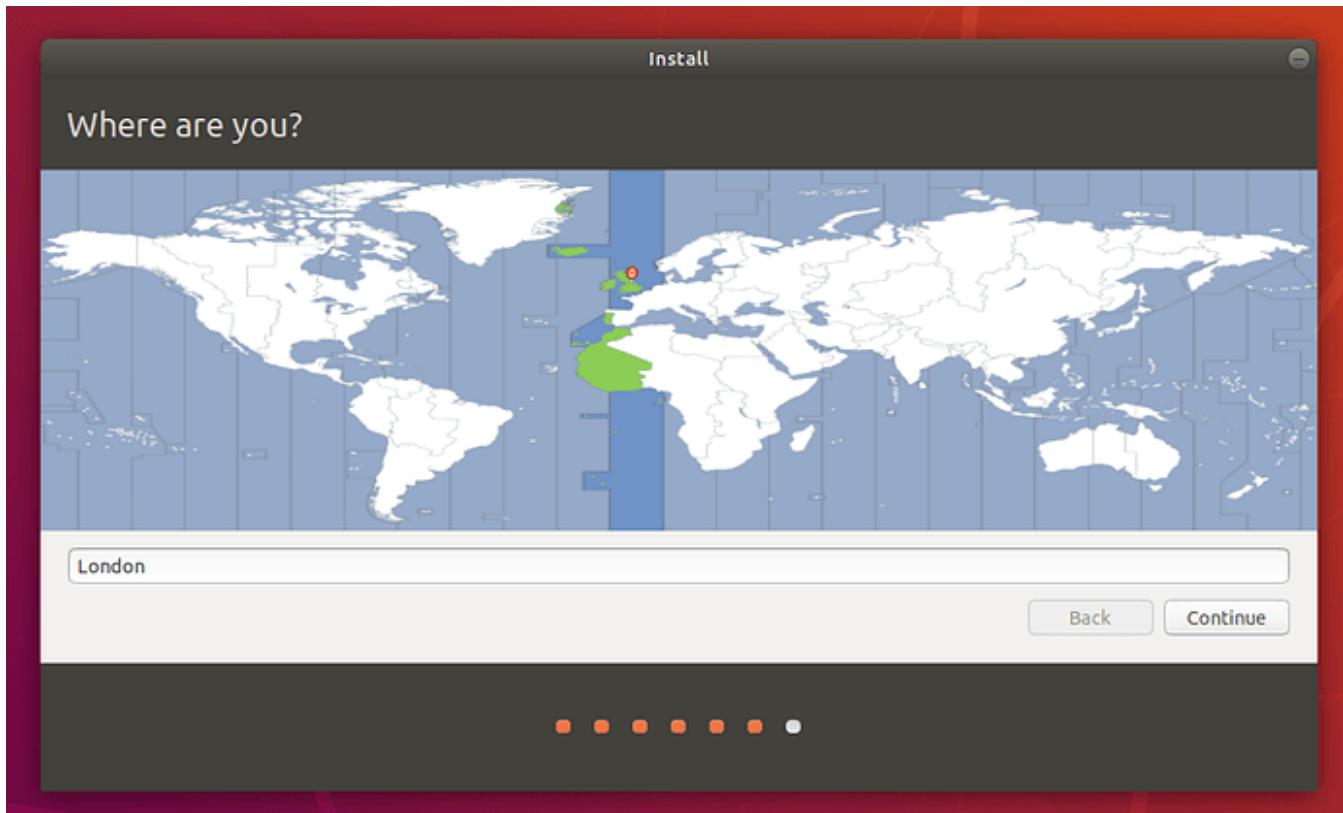


## Chọn vị trí lưu trữ

Nếu bạn được kết nối với internet, vị trí của bạn sẽ được phát hiện tự động. Kiểm tra vị trí của bạn là chính xác và nhấp vào **Chuyển tiếp liên tục** để tiếp tục.

Nếu bạn không chắc chắn về múi giờ của mình, hãy nhập tên của thị trấn hoặc thành phố địa phương hoặc sử dụng bản đồ để chọn vị trí của bạn.

Nếu bạn gặp vấn đề khi kết nối Internet, hãy sử dụng menu ở góc trên bên phải để chọn mạng.

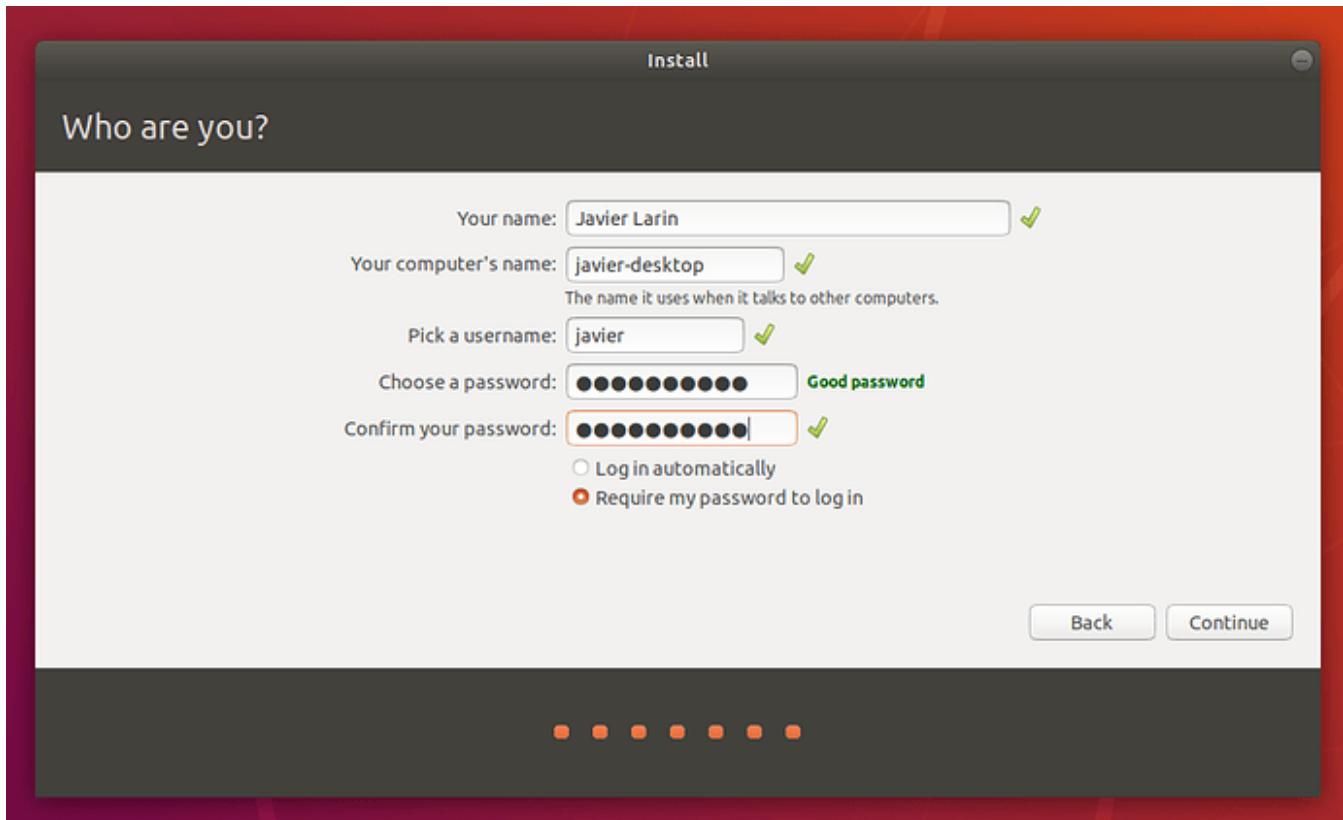


## Thông tin đăng nhập

Nhập tên của bạn và trình cài đặt sẽ tự động đề xuất tên máy tính và tên người dùng. Đây có thể dễ dàng được thay đổi nếu bạn thích. Tên máy tính là cách máy tính của bạn sẽ xuất hiện trên mạng, trong khi tên người dùng của bạn sẽ là tên đăng nhập và tên tài khoản của bạn.

Tiếp theo, nhập một mật khẩu mạnh. Trình cài đặt sẽ cho bạn biết nếu nó quá yếu.

Bạn cũng có thể chọn bật mã hóa thư mục nhà và đăng nhập tự động. Nếu máy của bạn là thiết bị di động, chúng tôi khuyên bạn nên tắt đăng nhập tự động và bật mã hóa. Điều này sẽ ngăn mọi người truy cập các tệp cá nhân của bạn nếu máy bị mất hoặc bị đánh cắp.

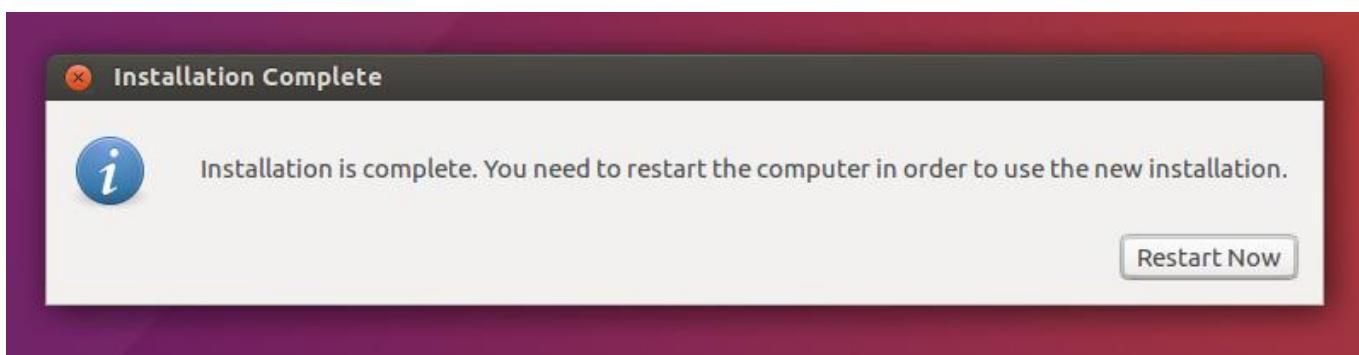


Nếu bạn kích hoạt mã hóa thư mục nhà và bạn quên mật khẩu, bạn đã có thể lấy lại bất kỳ dữ liệu cá nhân nào được lưu trong thư mục nhà của bạn.

### Cài đặt chạy ngầm trong nền

Trình cài đặt bây giờ sẽ hoàn tất trong nền trong khi cửa sổ cài đặt dạy cho bạn một chút về mức độ tuyệt vời của Ubuntu. Tùy thuộc vào tốc độ của máy và kết nối mạng của bạn, việc cài đặt chỉ mất vài phút.

### Hoàn tất cài đặt



Sau khi mọi thứ đã được cài đặt và định cấu hình, một cửa sổ nhỏ sẽ xuất hiện yêu cầu bạn khởi động lại máy. Nhập vào Khởi động lại ngay và xóa ổ đĩa flash DVD hoặc USB

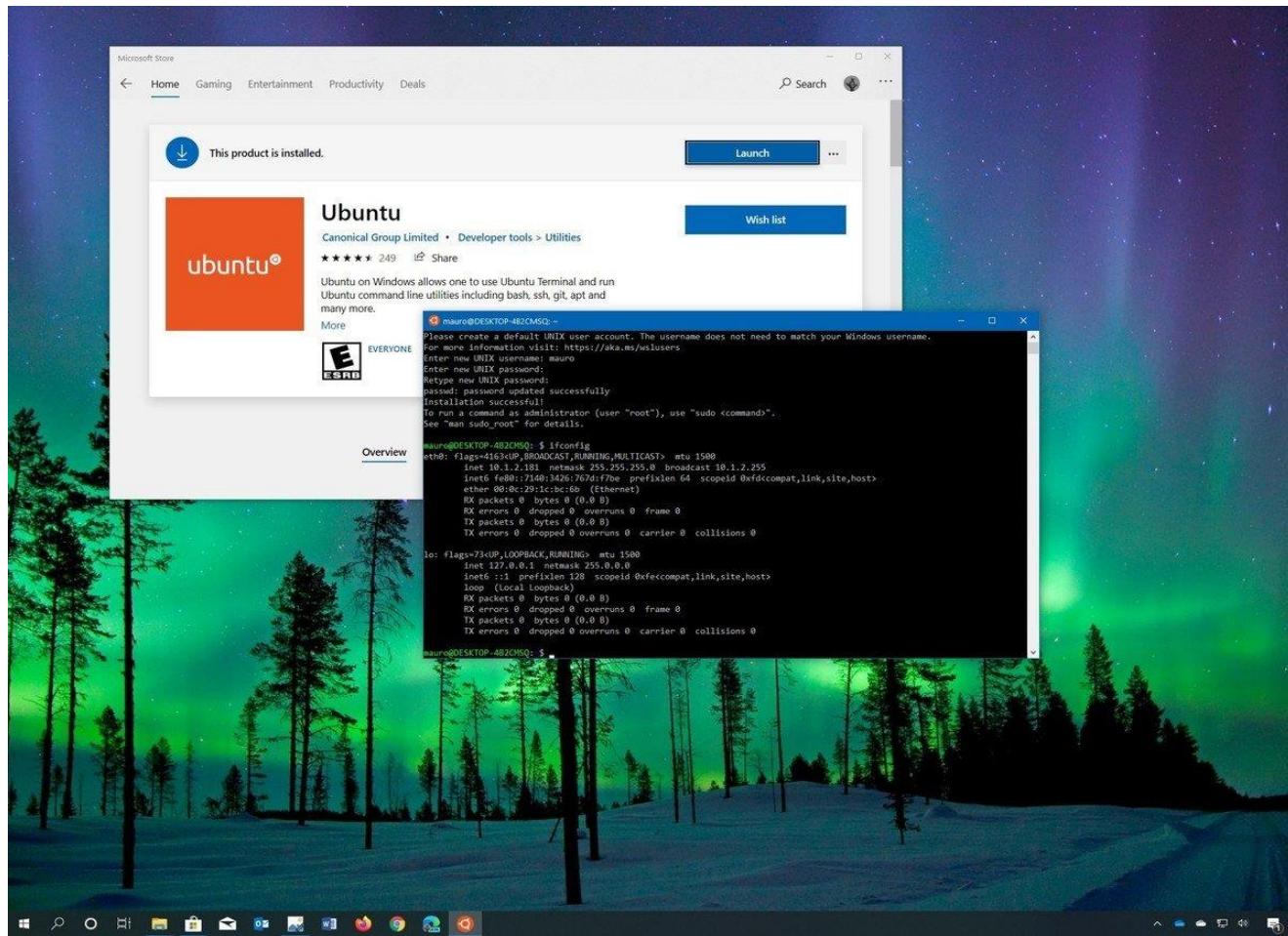
khi được nhắc. Nếu bạn đã bắt đầu cài đặt trong khi thử nghiệm máy tính để bàn, bạn cũng có tùy chọn để tiếp tục thử nghiệm.

Xin chúc mừng! Bạn đã cài đặt thành công hệ điều hành Linux phổ biến nhất thế giới!

Bây giờ đã đến lúc bắt đầu thưởng thức Ubuntu!

## 1.9 Cài đặt Ubuntu Bash Shell trên Windows 10 từ Microsoft Store

### Giới thiệu chung



Trên Windows 10, Hệ thống con Windows cho Linux (WSL) là một tính năng tạo ra một môi trường nhẹ cho phép bạn cài đặt và chạy các phiên bản Linux được hỗ trợ (như Ubuntu, OpenSuse, Debian, v.v.) mà không cần thiết lập phức tạp máy ảo hoặc máy tính khác nhau.

Mặc dù bạn sẽ không có giao diện đồ họa, cách tiếp cận này cho phép các nhà phát triển và quản trị viên mạng sử dụng một danh sách dài các công cụ và dịch vụ lệnh (như vim, tmux, andemacs, sshd, Apache và MySQL) và chạy các tập lệnh shell Bash được viết trong Python, Ruby, C #, F # và nhiều thứ khác bằng cách điều khiển lệnh.

Trong hướng dẫn Windows 10 này, chúng tôi sẽ hướng dẫn bạn các bước để cài đặt Hệ thống con Windows cho Linux bằng ứng dụng Cài đặt cũng như PowerShell. Ngoài ra, chúng tôi sẽ chỉ cho bạn các hướng dẫn để tải xuống và cài đặt các bản phân phối Linux và các bước để loại bỏ đúng tất cả các thành phần khi bạn không còn cần môi trường nguồn mở.

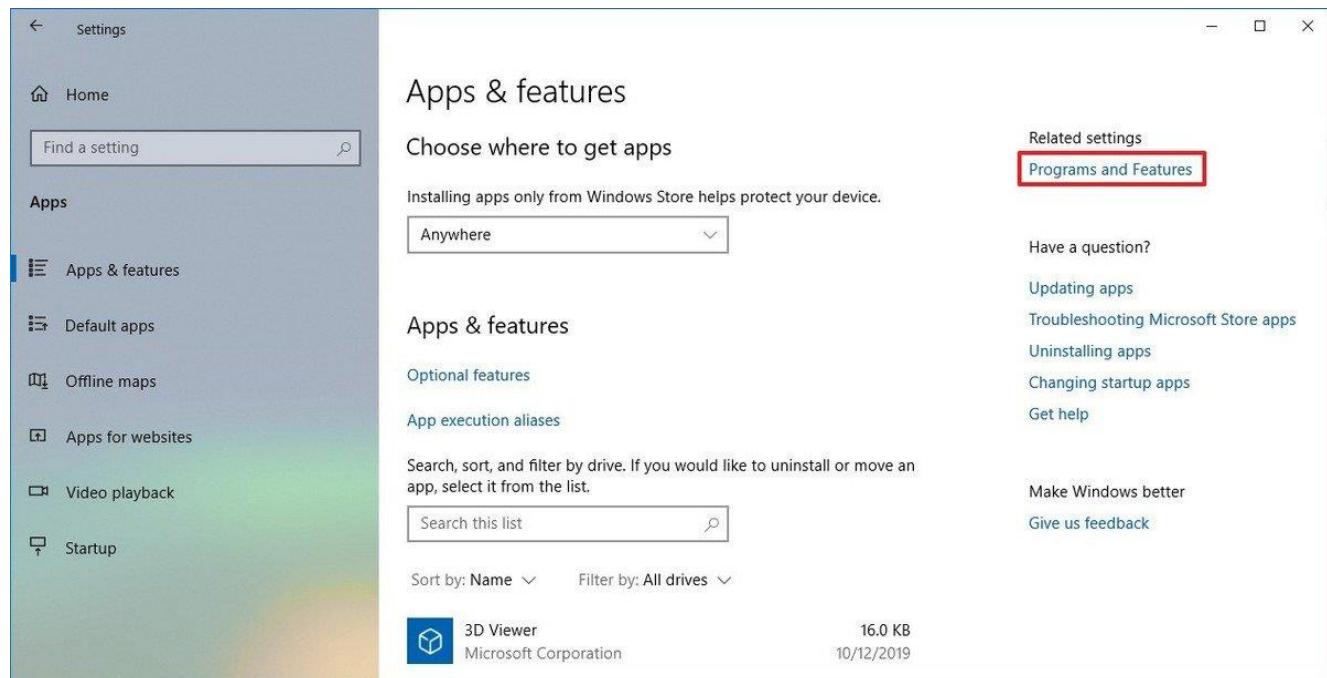
## Cài đặt WSL (Windows Subsystem for Linux)

Nếu bạn muốn chạy distro của Linux trên Windows 10, trước tiên bạn phải bật tính năng Windows subsystem cho Linux trước khi bạn có thể tải xuống và cài đặt hương vị của Linux mà bạn muốn sử dụng.

Kích hoạt hệ thống con Windows cho Linux bằng Cài đặt

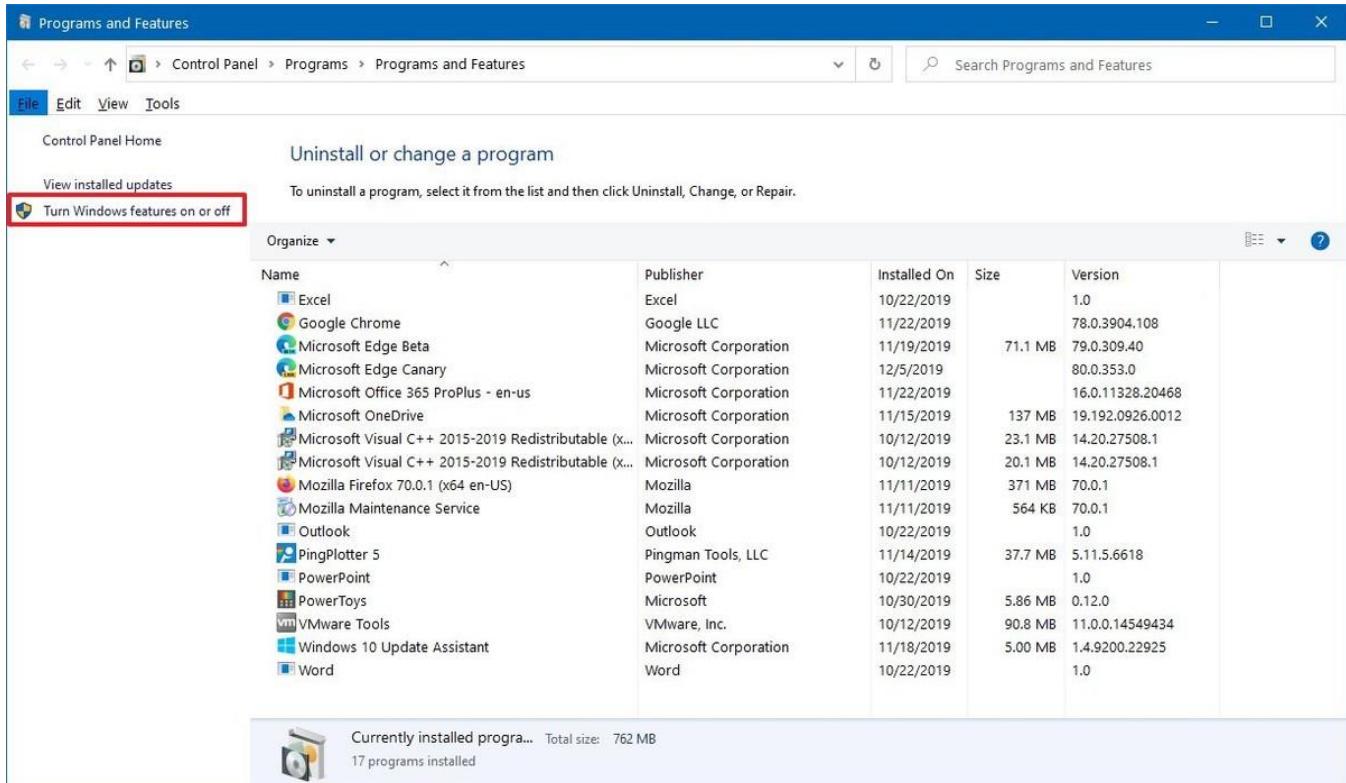
Để cài đặt WSL bằng Cài đặt trên Windows 10, hãy sử dụng các bước sau:

1. Mở **Settings**.
2. Nhập vào **Apps**.
3. Trong phần "**Related settings**", nhập vào tùy chọn **Programs and Features**.



4. Nhập vào tùy chọn **Bật** hoặc **tắt** các tính năng của Windows từ khung bên trái.
5. Chọn mục **Windows Subsystem for Linux**.
6. Nhập vào nút **OK**.
7. Nhập vào nút **Khởi động lại** ngay.

Khi bạn hoàn thành các bước, môi trường sẽ được cấu hình để tải xuống và chạy các bản phân phối của Linux trên Windows 10.



## Cài đặt các bản phân phối Linux trong Microsoft Store

Để cài đặt bản phân phối Linux trên Windows 10, hãy sử dụng các bước sau:

1. Mở Microsoft Store.

2. Tìm kiếm bản phân phối Linux mà bạn muốn cài đặt.

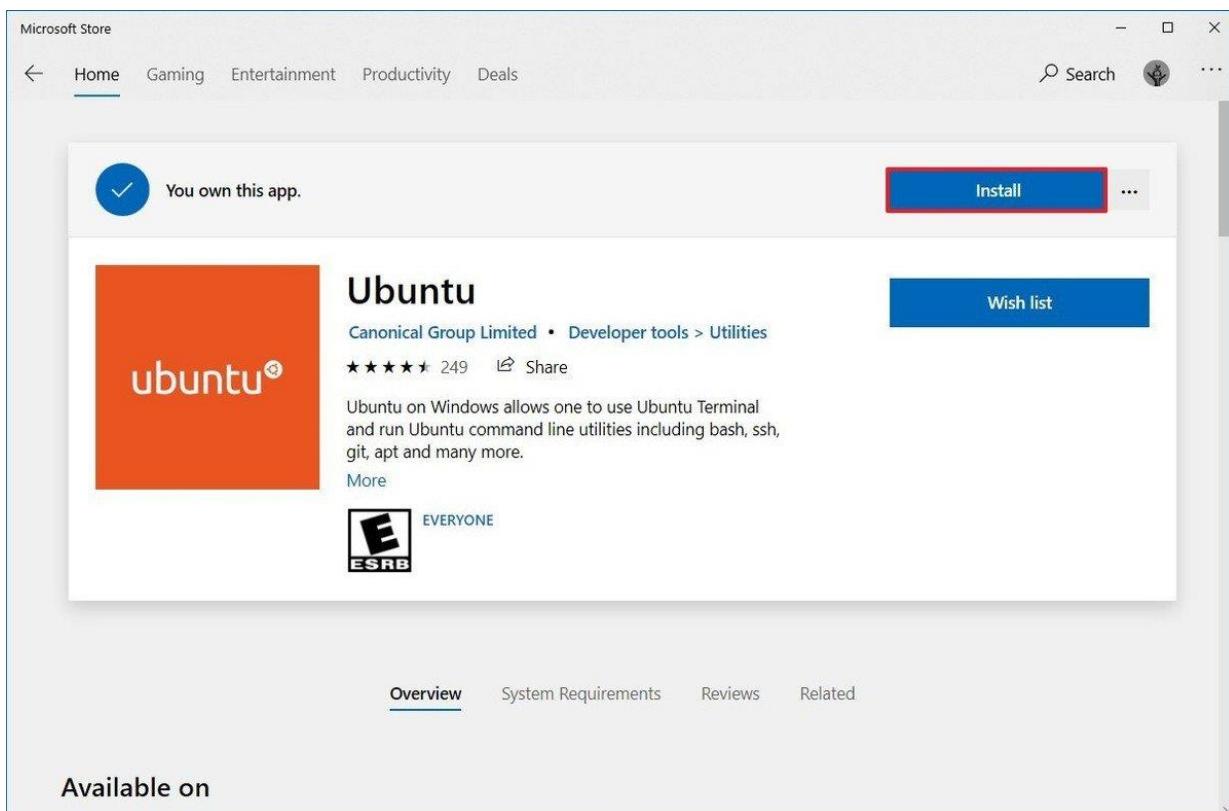
Một số bản phân phối có sẵn bao gồm:

- Ubuntu.
- OpenSuse Leap 15.
- Kali Linux.
- Debian.
- Alpine WSL.
- Suse Linux Enterprise 12.

3. Chọn bản phân phối của Linux để cài đặt trên thiết bị của bạn.



2. Chọn nút **Get** (hoặc **Install**)
3. Chọn nút **Launch**.
4. Tạo người dùng *username* cho bản cài đặt Linux và nhấn **Enter**.
5. Tạo mật khẩu *password* cho người dùng *username* và nhấn **Enter**.



```
mauro@DESKTOP-4B2CMSQ: ~
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: mauro
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

mauro@DESKTOP-4B2CMSQ: $
```

6. Lặp lại *password* và nhấn Enter để xác nhận.

7. Sau khi bạn hoàn thành các bước, bạn có thể bắt đầu sử dụng bản phân phối như bất kỳ hương vị nào khác của Linux (tất nhiên không có giao diện người dùng đồ họa).

### Gỡ cài đặt các bản phân phối Linux bằng Settings

Nếu bạn không còn cần chạy Linux trên thiết bị của mình nữa, bạn nên xóa các bản phát hành và sau đó tắt WSL.

Để gỡ cài đặt bản phân phối Linux, hãy sử dụng các bước sau:

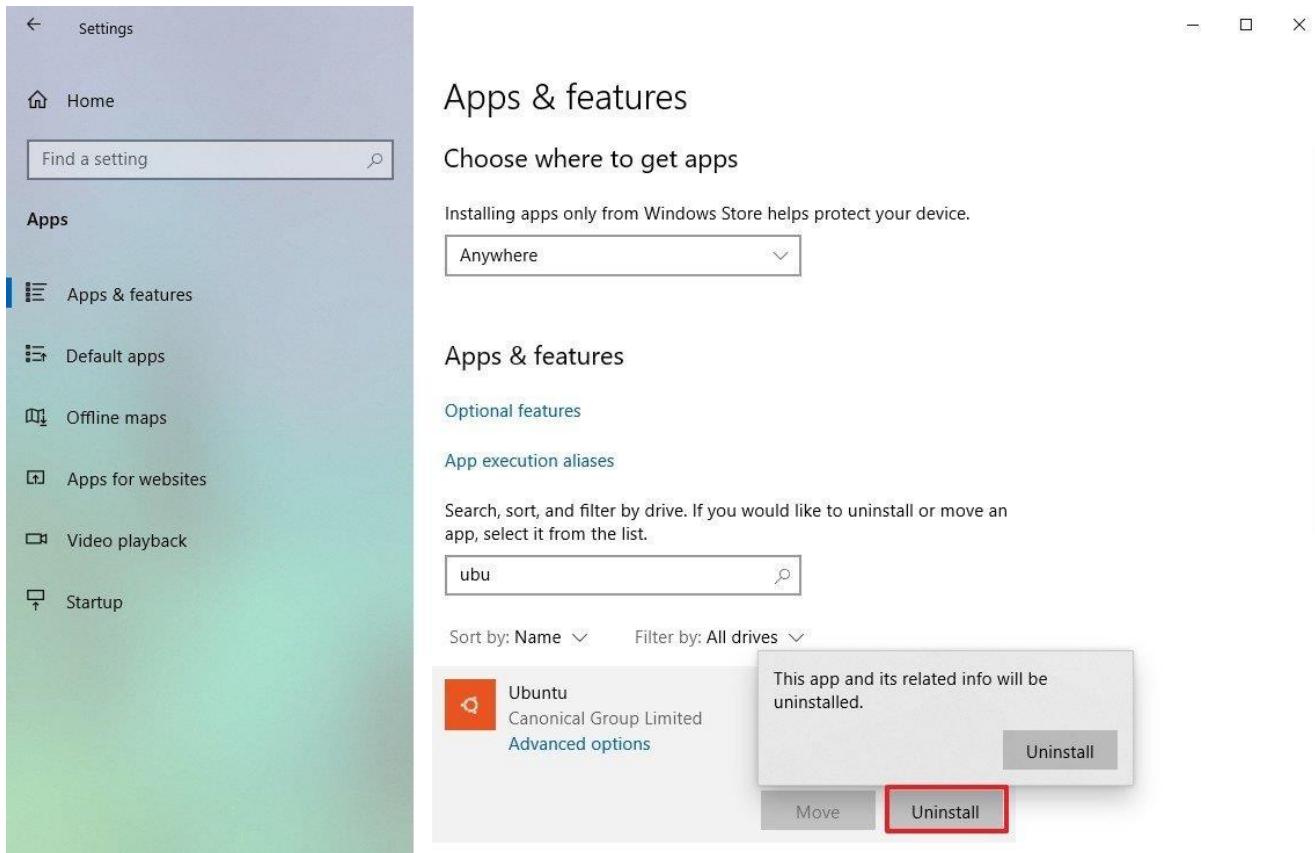
1. Mở **Settings**.

2. Chọn **Apps**.

3. Chọn **Apps & features**.

4. Trong mục "**Apps & features**", chọn bản phân phối — thí dụ **Ubuntu**.

5. Chọn **Uninstall**.



6. Chọn **Uninstall** lần nữa để khẳng định.

Khi bạn hoàn thành các bước, việc cài đặt Linux sẽ bị xóa và bạn có thể tiến hành các bước để vô hiệu hóa WSL.

## **1.10 Cài đặt máy ảo (VM) Ubuntu Server trong điện toán đám mây Microsoft Azure**

### **Giới thiệu chung**

Mục đích của bài thực hành là tạo ra một máy ảo Azure (VM) bằng cách sử dụng ảnh máy chủ (image) **Ubuntu** mới nhất (phiên bản 18.04 LTS) và sau đó di chuyển sang dịch vụ đám mây. Trong mục này, bạn sẽ cần tìm hiểu về các tùy chọn khi tạo một máy ảo trong Microsoft Azure.

### **Máy ảo trong Microsoft Azure**

Azure Virtual Machines là một tài nguyên điện toán đám mây có thể mở rộng theo yêu cầu. Chúng bao gồm bộ xử lý, bộ nhớ, lưu trữ và tài nguyên mạng. Bạn có thể bắt đầu và dừng các máy ảo theo ý muốn và quản lý chúng từ cổng thông tin Azure hoặc với Azure CLI. Bạn cũng có thể sử dụng Secure Shell (SSH) từ xa để kết nối trực tiếp với VM đang chạy và thực thi các lệnh như thể bạn đang ở trên một máy tính cục bộ.

### **Sử dụng Linux trong Microsoft Azure**

Tạo các máy ảo dựa trên Linux trong Azure rất dễ dàng. Microsoft đã hợp tác với các nhà cung cấp Linux nổi tiếng để đảm bảo các bản phân phối của họ được tối ưu hóa cho nền tảng Azure. Bạn có thể tạo các máy ảo từ các hình ảnh sẵn cho nhiều bản phân phối Linux phổ biến, chẳng hạn như SUSE, Red Hat và Ubuntu hoặc xây dựng bản phân phối Linux của riêng bạn để chạy trong đám mây.

### **Tạo máy ảo Azure VM**

Máy ảo có thể được xác định và triển khai trên Azure theo nhiều cách: cổng thông tin Azure, tập lệnh (sử dụng Azure CLI hoặc Azure PowerShell) hoặc mẫu Trình quản lý tài nguyên Azure. Trong mọi trường hợp, bạn sẽ cần cung cấp một số thông tin mà chúng tôi sẽ đề cập ngay sau đây.

Azure Marketplace cũng cung cấp các hình ảnh được cấu hình sẵn bao gồm cả hệ điều hành và các công cụ phần mềm yêu thích được cài đặt cho các tình huống cụ thể.

Compute

Filter

Search Compute

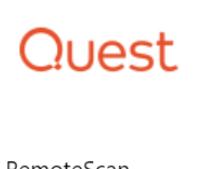
Recommended

More

					
Windows Server Microsoft	Red Hat Enterprise Linux RedHat	Ubuntu Server Canonical	SQL Server 2017 Enterprise Microsoft	Virtual machine scale set Microsoft	Container Service Microsoft

Virtual Machine Images

More

					
Unified Communications Quest Software 	RemoteScan Enterprise Quest Software 	Pivotal Cloud Foundry on Pivotal Software... 	Aqua Container Security Platform Aqua Security 	Commvault Trial Commvault 	BPM - Document Management AuraPortal 

## Các tài nguyên cần sử dụng cho Linux VM

Khi tạo máy ảo Linux trong Azure, bạn cũng tạo tài nguyên để lưu trữ máy ảo. Các tài nguyên này phối hợp với nhau để ảo hóa máy tính và chạy hệ điều hành Linux. Chúng phải tồn tại (và được chọn trong quá trình tạo VM) hoặc chúng sẽ được tạo bằng VM:

- Một máy ảo cung cấp tài nguyên CPU và bộ nhớ
- Tài khoản lưu trữ Azure để giữ các đĩa cứng ảo
- Đĩa ảo để chứa HDH, ứng dụng và dữ liệu
- Mạng ảo (VNet) để kết nối VM với các dịch vụ Azure khác hoặc phần cứng tại chỗ của bạn
- Giao diện mạng để liên lạc với VNet
- Địa chỉ IP công cộng tùy chọn để bạn có thể truy cập VM

Giống như các dịch vụ Azure khác, bạn sẽ cần Nhóm tài nguyên để chứa VM (và tùy ý nhóm các tài nguyên này để quản trị). Khi bạn tạo một VM mới, bạn có thể sử dụng một nhóm tài nguyên hiện có hoặc tạo một VM mới.

## Lựa chọn ảnh máy ảo VM Image

Chọn một hình ảnh là một trong những quyết định đầu tiên và quan trọng nhất bạn sẽ đưa ra khi tạo VM. Hình ảnh là một mẫu được sử dụng để tạo VM. Các mẫu này bao gồm HĐH và thường là các phần mềm khác, chẳng hạn như các công cụ phát triển hoặc môi trường lưu trữ web.

Bất cứ thứ gì máy tính có thể cài đặt đều có thể được đưa vào một hình ảnh. Bạn có thể tạo một VM từ một hình ảnh được cấu hình sẵn để chính xác các tác vụ bạn cần, chẳng hạn như lưu trữ một ứng dụng web trên Máy chủ HTTP Apache.

Bạn cũng có thể tạo và tải lên hình ảnh đĩa tùy chỉnh

## Thay đổi kích thước máy ảo VM

Giống như một máy vật lý có một lượng bộ nhớ và sức mạnh CPU nhất định, một máy ảo cũng vậy. Azure cung cấp một loạt các VM có kích cỡ khác nhau ở các mức giá khác nhau. Kích thước bạn chọn sẽ xác định khả năng xử lý, bộ nhớ và dung lượng lưu trữ tối đa của VM.

## Cảnh báo

Có giới hạn hạn ngạch trên mỗi thuê bao có thể ảnh hưởng đến việc tạo VM. Nếu bạn gặp phải các giới hạn hạn ngạch này, bạn có thể mở yêu cầu hỗ trợ khách hàng trực tuyến để tăng giới hạn của mình.

Các kích thước VM được nhóm thành các danh mục, bắt đầu với Sê-ri B để thử nghiệm cơ bản và chạy đến sê-ri H cho các tác vụ điện toán lớn. Bạn nên chọn kích thước của VM dựa trên khối lượng công việc bạn muốn thực hiện. Có thể thay đổi kích thước của VM sau khi được tạo, nhưng trước tiên VM phải được dừng lại. Vì vậy, tốt nhất là kích thước nó một cách thích hợp ngay từ đầu nếu có thể.

Dưới đây là một số hướng dẫn dựa trên kịch bản bạn đang nhắm mục tiêu

Bạn cần làm gì ?	Kích thước
<b>Sử dụng chung (máy tính/web):</b> Thủ nghiệm và phát triển, cơ sở dữ liệu vừa và nhỏ, máy chủ web lưu lượng truy cập thấp đến trung bình.	B, Dsv3, Dv3, DSv2, Dv2
<b>Nhiệm vụ tính toán nặng nề:</b> Máy chủ web lưu lượng trung bình, thiết bị mạng, quy trình xử lý lô và máy chủ ứng dụng.	Fsv2, Fs, F

Bạn cần làm gì ?	Kích thước
<b>Sử dụng bộ nhớ lớn:</b> Máy chủ cơ sở dữ liệu quan hệ, bộ nhớ cache trung bình đến lớn và phân tích trong bộ nhớ.	Esv3, Ev3, M, GS, G, DSv2, Dv2
<b>Lưu trữ và xử lý dữ liệu:</b> Cơ sở dữ liệu lớn, cơ sở dữ liệu SQL và NoQuery cần thông lượng đĩa và I / O cao.	Ls
<b>Kết xuất đồ họa nặng:</b> chỉnh sửa video, cũng như đào tạo mô hình và suy luận (ND) với học tập sâu.	NV, NC, NCv2, NCv3, ND
<b>Tính toán hiệu năng cao (HPC):</b> Khối lượng công việc của bạn cần các máy ảo CPU mạnh nhất và nhanh nhất với các giao diện mạng thông lượng cao tùy chọn.	H

### Lựa chọn lưu trữ cho máy ảo

Các quyết định tiếp theo xoay quanh việc lưu trữ. Đầu tiên, bạn có thể chọn công nghệ đĩa. Các tùy chọn bao gồm ổ đĩa cứng dựa trên đĩa truyền thống (HDD) hoặc ổ cứng thể rắn (SSD) hiện đại hơn. Giống như phần cứng bạn mua, bộ lưu trữ SSD có giá cao hơn nhưng cung cấp hiệu suất tốt hơn.

### Mẹo

Có hai cấp lưu trữ SSD: tiêu chuẩn và cao cấp. Chọn đĩa SSD tiêu chuẩn nếu bạn có khối lượng công việc bình thường nhưng muốn hiệu suất tốt hơn. Chọn đĩa SSD cao cấp nếu bạn có khối lượng công việc lớn I / O hoặc các hệ thống quan trọng cần xử lý dữ liệu rất nhanh.

### Ánh xạ lưu trữ vào đĩa

Azure sử dụng đĩa cứng ảo (VHD) để thể hiện các đĩa vật lý cho VM. VHD sao chép định dạng logic và dữ liệu của ổ đĩa nhưng được lưu trữ dưới dạng các đĩm trang trong tài khoản Azure Storage. Bạn có thể chọn trên cơ sở mỗi đĩa nên sử dụng loại lưu trữ nào (SSD hoặc HDD). Điều này cho phép bạn kiểm soát hiệu suất của từng đĩa, có thể dựa trên I / O mà bạn dự định thực hiện trên đó.

Theo mặc định, hai đĩa cứng ảo (VHD) sẽ được tạo cho máy ảo Linux của bạn:

- Đĩa hệ điều hành: Đây là ổ đĩa chính của bạn và có dung lượng tối đa 2048 GB. Nó sẽ được gắn nhãn là / dev / sda theo mặc định.
- Đĩa tạm thời: Điều này cung cấp lưu trữ tạm thời cho HĐH hoặc bất kỳ ứng dụng nào. Trên các máy ảo Linux, đĩa là / dev / sdb và được định dạng và gắn kết với / mnt bởi Azure Linux Agent. Nó có kích thước dựa trên kích thước VM và được sử dụng để lưu trữ tệp hoán đổi.

## **Cảnh báo**

Đĩa tạm thời không bền. Bạn chỉ nên ghi dữ liệu vào đĩa này không quan trọng đối với hệ thống

## **Những điều cần biết về dữ liệu**

Bạn có thể lưu trữ dữ liệu trên ổ đĩa chính cùng với HDH, nhưng cách tiếp cận tốt hơn là tạo các đĩa dữ liệu chuyên dụng. Bạn có thể tạo và đính kèm các đĩa bổ sung vào VM. Mỗi đĩa có thể chứa tới 32.767 gibibytes (GiB) dữ liệu, với dung lượng lưu trữ tối đa được xác định bởi kích thước VM bạn chọn.

## **Lưu ý**

Một khả năng thú vị là tạo hình ảnh VHD từ đĩa thật. Điều này cho phép bạn dễ dàng di chuyển thông tin hiện có từ máy tính tại chỗ sang đám mây.

## **Các ổ đĩa quản trị được và không quản trị được**

Lựa chọn lưu trữ cuối cùng bạn sẽ thực hiện là sử dụng đĩa không được quản lý hoặc được quản lý.

Với các đĩa không được quản lý, bạn chịu trách nhiệm cho các tài khoản lưu trữ được sử dụng để giữ các VHD tương ứng với các đĩa VM của bạn. Bạn trả giá tài khoản lưu trữ cho số lượng không gian bạn sử dụng. Một tài khoản lưu trữ duy nhất có giới hạn tốc độ cố định là 20.000 thao tác I / O / giây. Điều này có nghĩa là một tài khoản lưu trữ duy nhất có khả năng hỗ trợ 40 đĩa cứng ảo tiêu chuẩn ở tốc độ tối đa. Nếu bạn cần mở rộng quy mô, thì bạn cần nhiều hơn một tài khoản lưu trữ, điều này có thể trở nên phức tạp.

Các đĩa được quản lý là mô hình lưu trữ đĩa mới hơn và được đề xuất. Họ giải quyết một cách tao nhã sự phức tạp này bằng cách đặt gánh nặng quản lý tài khoản lưu trữ lên Azure. Bạn chỉ định loại đĩa (Premium hoặc Standard) và kích thước của đĩa, và Azure tạo và quản lý cả đĩa và bộ lưu trữ mà nó sử dụng. Bạn không phải lo lắng về giới hạn tài khoản lưu trữ, điều này giúp chúng dễ dàng mở rộng hơn. Họ cũng cung cấp một số lợi ích khác:

- Tăng độ tin cậy: Azure đảm bảo rằng các VHD được liên kết với các máy ảo có độ tin cậy cao sẽ được đặt trong các phần khác nhau của Bộ lưu trữ Azure để cung cấp mức độ phục hồi tương tự.
- Bảo mật tốt hơn: Đĩa được quản lý là tài nguyên được quản lý thực sự trong nhóm tài nguyên. Điều này có nghĩa là họ có thể sử dụng kiểm soát truy cập dựa trên vai trò để hạn chế những người có thể làm việc với dữ liệu VHD.

- Hỗ trợ ảnh chụp nhanh: Ảnh chụp nhanh có thể được sử dụng để tạo bản sao chỉ đọc của VHD. Bạn phải tắt VM sở hữu, nhưng việc tạo ảnh chụp nhanh chỉ mất vài giây. Khi đã xong, bạn có thể bật máy ảo và sử dụng ảnh chụp nhanh để tạo một máy ảo trùng lặp để khắc phục sự cố sản xuất hoặc khôi phục VM đến thời điểm chụp ảnh nhanh.
- Hỗ trợ sao lưu: Các đĩa được quản lý có thể được tự động sao lưu vào các vùng khác nhau để khắc phục thảm họa với Azure Backup mà không ảnh hưởng đến dịch vụ của VM.

## Kết nối mạng

Máy ảo giao tiếp với các tài nguyên bên ngoài bằng mạng ảo (VNet). VNet đại diện cho một mạng riêng trong một khu vực duy nhất mà tài nguyên của bạn giao tiếp. Mạng ảo giống như các mạng bạn quản lý tại chỗ. Bạn có thể chia chúng thành các mạng con để cài đặt tài nguyên, kết nối chúng với các mạng khác (bao gồm cả mạng tại chỗ của bạn) và áp dụng quy tắc lưu lượng để điều chỉnh các kết nối trong và ngoài nước.

## Quy hoạch mạng

Khi bạn tạo một VM mới, bạn sẽ có tùy chọn tạo một mạng ảo mới hoặc sử dụng một VNet hiện có trong khu vực của bạn.

Có Azure tạo mạng cùng với VM rất đơn giản, nhưng có thể không lý tưởng cho hầu hết các kịch bản. Tốt hơn hết là lên kế hoạch cho các yêu cầu mạng của bạn trước cho tất cả các thành phần trong kiến trúc của bạn và tạo cấu trúc VNet riêng. Sau đó, tạo các VM và đặt chúng vào các VNets đã được tạo. Chúng ta sẽ xem xét thêm về các mạng ảo sau này trong mô-đun này.

Trước khi tạo một máy ảo, chúng ta cần quyết định cách chúng ta muốn quản trị VM. Hãy xem xét các lựa chọn của chúng tôi.

## Sử dụng SSH để quản trị máy ảo Linux VM

Trước khi chúng ta có thể tạo một máy ảo Linux trong Azure, chúng ta sẽ cần suy nghĩ về việc truy cập từ xa. Chúng tôi muốn có thể đăng nhập vào máy chủ web Linux của mình để định cấu hình phần mềm và thực hiện bảo trì. Cách tiếp cận mặc định để quản trị máy ảo Linux được lưu trữ trong Azure là SSH.

## SSH là gì?

Secure Shell (SSH) là một giao thức kết nối được mã hóa cho phép đăng nhập an toàn trên các kết nối không bảo mật. SSH cho phép bạn kết nối với thiết bị đầu cuối từ một vị trí từ xa bằng kết nối mạng.

Có hai cách tiếp cận chúng ta có thể sử dụng để xác thực kết nối SSH: tên người dùng và mật khẩu hoặc cặp khóa SSH.

### **Mẹo**

Mặc dù SSH cung cấp kết nối được mã hóa, nhưng việc sử dụng mật khẩu với kết nối SSH khiến VM dễ bị tấn công bằng mật khẩu. Một phương thức an toàn và ưa thích hơn để kết nối với máy ảo Linux với SSH là cặp khóa công khai, còn được gọi là khóa SSH.

Với cặp khóa SSH, bạn có thể đăng nhập vào các máy ảo Azure dựa trên Linux mà không cần mật khẩu. Đây là một cách tiếp cận an toàn hơn nếu bạn chỉ có kế hoạch đăng nhập vào VM từ một vài máy tính. Nếu bạn cần có thể truy cập Linux VM từ nhiều địa điểm khác nhau, kết hợp tên người dùng và mật khẩu có thể là một cách tiếp cận tốt hơn. Có hai phần cho cặp khóa SSH: khóa chung và khóa riêng.

- Khóa chung được đặt trên máy ảo Linux của bạn hoặc bất kỳ dịch vụ nào khác mà bạn muốn sử dụng với mật mã khóa công khai. Điều này có thể được chia sẻ với bất cứ ai.
- Khóa riêng là những gì bạn trình bày để xác minh danh tính của bạn với máy ảo Linux khi bạn thực hiện kết nối SSH. Xem xét thông tin bí mật này và bảo vệ thông tin này giống như bạn sẽ nhập mật khẩu hoặc bất kỳ dữ liệu riêng tư nào khác.

Bạn có thể sử dụng cùng một cặp khóa công khai riêng lẻ để truy cập nhiều dịch vụ và máy ảo Azure.

### **Tạo cặp khóa SSH**

Trên Linux, Windows 10 và MacOS, bạn có thể sử dụng lệnh ssh-keygen tích hợp để tạo các tệp khóa công khai và riêng tư SSH.

### **Mẹo**

Windows 10 bao gồm ứng dụng khách SSH với Bản cập nhật Fall Creators Update. Các phiên bản trước của Windows yêu cầu phần mềm bổ sung để sử dụng SSH; kiểm tra tài liệu để biết chi tiết đầy đủ Ngoài ra, bạn có thể cài đặt hệ thống con Linux cho Windows và có cùng chức năng.

Chúng tôi sẽ sử dụng Azure Cloud Shell, sẽ lưu trữ các khóa được tạo trong Azure trong tài khoản lưu trữ riêng của bạn. Bạn cũng có thể nhập các lệnh này trực tiếp vào shell cục bộ nếu muốn. Bạn sẽ cần điều chỉnh các hướng dẫn trong suốt mô-đun này để phản ánh một phiên cục bộ nếu bạn thực hiện phương pháp này.

Đây là lệnh tối thiểu cần thiết để tạo cặp khóa cho máy ảo Azure. Điều này sẽ tạo ra cặp khóa công khai RSA 2 (SSH-2) RSA. Độ dài tối thiểu là 2048, nhưng vì lợi ích của mô-đun học tập này, chúng tôi sẽ sử dụng 4096.

Nhập lệnh này vào Cloud Shell

## **ssh-keygen -t rsa -b 4096**

Công cụ sẽ nhắc tên tệp và cụm mật khẩu tùy chọn. Đối với bài tập này, chỉ cần mặc định. Nó sẽ tạo hai tệp: id\_rsa và id\_rsa.pub trong thư mục ~ / .ssh. Các tập tin sẽ được ghi đè nếu chúng tồn tại. Có nhiều tùy chọn khác nhau mà bạn có thể sử dụng để cung cấp tên tệp hoặc cụm mật khẩu để tránh lời nhắc.

### **Cụm mật khẩu khóa riêng**

Bạn có thể tùy ý cung cấp cụm mật khẩu trong khi tạo khóa riêng của mình. Đây là mật khẩu bạn phải nhập khi sử dụng khóa. Cụm mật khẩu này được sử dụng để truy cập tệp khóa SSH riêng tư và không phải là mật khẩu tài khoản người dùng.

Khi bạn thêm cụm mật khẩu vào khóa SSH, nó sẽ mã hóa khóa riêng bằng AES 128 bit để khóa riêng là vô dụng nếu không có cụm mật khẩu để giải mã.

### **Lưu ý**

Chúng tôi khuyên bạn nên thêm một cụm mật khẩu. Nếu kẻ tấn công đã đánh cắp khóa riêng của bạn và khóa đó không có cụm mật khẩu, họ có thể sử dụng khóa riêng đó để đăng nhập vào bất kỳ máy chủ nào có khóa chung tương ứng. Nếu cụm mật khẩu bảo vệ khóa riêng, kẻ tấn công đó không thể sử dụng nó. Điều này cung cấp một lớp bảo mật bổ sung cho cơ sở hạ tầng của bạn trên Azure.

### **Sử dụng cặp khóa SSH với máy ảo Azure Linux**

Khi bạn đã tạo cặp khóa, bạn có thể sử dụng nó với máy ảo Linux trong Azure. Bạn có thể cung cấp khóa chung trong quá trình tạo VM hoặc thêm nó sau khi VM được tạo. Bạn có thể xem nội dung của tệp trong Azure Cloud Shell bằng lệnh sau:

```
cat ~/.ssh/id_rsa.pub
```

Kết xuất lệnh này sẽ là một dòng duy nhất và trông giống như:

*ssh-rsa*

```
XXXXXXXXXXc2EAAAADAXABAAABAXC5Am7+fGZ+5zXBGgXS6GUvmsXCLGc7tX  
7/rViXk3+eShZzaXnt75gUmT1I2f75zFn2hIAIDGKWF4g12KWcZxy81TniUOTjUsVlwPy  
mXUXxESL/UfJKfbdstBhTOdy5EG9rYWA0K43SJmwPhH28BpoLfXXXXGX/ilsXXXX  
KgRLiJ2W19MzXHp8z3Lxw7r9wx3HaVIP4XiFv9U4hGcp8RMIIIMP1nNesFlOBpG4pV  
2bJRBTXNXeY4l6F8WZ3C4kuf8XxOo08mXaTpVZ3T1841altnNTZCcPkXuMrBjYSJbA8  
npoXAXNwiivyo3X2KMXXXXdXXXXXXXXXXXXCXXXX/ azureuser@myserver
```

Sao chép giá trị này, vì vậy bạn có thể sử dụng nó trong bài tập tiếp theo.

### **Sử dụng khóa SSH khi tạo Linux VM**

Để áp dụng khóa SSH trong khi tạo Linux VM mới, bạn sẽ cần sao chép nội dung của khóa chung và cung cấp nó cho cổng thông tin Azure hoặc cung cấp tệp khóa chung cho lệnh Azure CLI hoặc Azure PowerShell. Chúng tôi sẽ sử dụng phương pháp này khi chúng tôi tạo Linux VM của mình.

## Thêm khóa SSH vào máy ảo Linux hiện có

Nếu bạn đã tạo VM, bạn có thể cài đặt khóa chung vào máy ảo Linux của mình bằng lệnh ssh-copy-id. Khi khóa đã được cấp phép cho SSH, nó sẽ cấp quyền truy cập vào máy chủ mà không cần mật khẩu, mặc dù vậy bạn vẫn sẽ được nhắc nhập cụm mật khẩu trên khóa nếu bạn đặt mật khẩu.

Ví dụ: nếu chúng ta có một máy ảo Linux có tên myserver với người dùng azureuser, chúng ta có thể sử dụng lệnh sau để cài đặt tệp khóa công khai và ủy quyền cho người dùng bằng khóa:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub azureuser@myserver
```

Bây giờ chúng ta đã có khóa công khai, hãy chuyển sang cổng thông tin Azure và tạo một máy ảo Linux.

## Tạo máy ảo Linux với Azure Portal

Ta có thể tạo máy ảo Linux bằng cổng thông tin Azure, Azure CLI hoặc Azure PowerShell. Cách tiếp cận đơn giản nhất khi bạn bắt đầu với Azure là sử dụng cổng thông tin vì nó sẽ hướng dẫn bạn thông tin cần thiết và cung cấp các gợi ý và thông báo hữu ích trong quá trình tạo:

1. Đăng nhập vào cổng thông tin Azure bằng chính tài khoản bạn đã kích hoạt hộp cát.
2. Trên menu cổng thông tin Azure hoặc từ Trang chủ, chọn Tạo tài nguyên.
3. Trong hộp tìm kiếm, nhập Ubuntu Server.
4. Trong kết quả Thị trường, chọn liên kết Tất cả kết quả ở phía bên phải để xem các phiên bản khác nhau có sẵn.
5. Chọn Ubuntu Server 18.04 LTS Canonical từ danh sách được trình bày.
6. Nhấp vào nút Tạo để bắt đầu định cấu hình VM.

## Định cấu hình cài đặt VM

Trải nghiệm tạo VM trong cổng thông tin được trình bày theo định dạng trình hướng dẫn để đưa bạn qua tất cả các khu vực cấu hình cho VM. Nhấp vào nút Tiếp theo sẽ đưa bạn đến phần cấu hình tiếp theo. Tuy nhiên, bạn có thể di chuyển giữa các phần theo ý muốn với các tab chạy trên đầu xác định từng phần.

Khi bạn điền vào tất cả các tùy chọn bắt buộc (được xác định bằng dấu sao màu đỏ), bạn có thể bỏ qua phần còn lại của trải nghiệm trình hướng dẫn và bắt đầu tạo VM thông qua nút Xem lại + Tạo ở phía dưới.

Chúng ta sẽ bắt đầu với phần Cơ bản. Các hướng dẫn này dành cho công thông tin Sandbox. Nếu bạn đang sử dụng một tài khoản công thông tin Azure khác, bạn có thể cần điều chỉnh một số chi tiết cho phù hợp.

### **Định cấu hình cài đặt VM cơ bản**

1. Đối với Đăng ký, đăng ký hộp cát nên được chọn cho bạn theo mặc định.
2. Đối với nhóm Tài nguyên, nhóm tài nguyên có tên [tên nhóm tài nguyên hộp cát] nên được chọn cho bạn theo mặc định.
3. Trong phần Chi tiết sơ thẩm, nhập tên cho VM máy chủ web của bạn, chẳng hạn như test-web-eus-vm1. Điều này cho biết môi trường (thử nghiệm), vai trò (web), địa điểm (Đông Hoa Kỳ), dịch vụ (vm) và số hiệu (1).

Nó được coi là thực tiễn tốt nhất để chuẩn hóa tên tài nguyên của bạn, vì vậy bạn có thể nhanh chóng xác định mục đích của chúng. Tên máy ảo Linux phải có từ 1 đến 64 ký tự và bao gồm các số, chữ cái và dấu gạch ngang.

### **Lưu ý**

Khi bạn thay đổi cài đặt và tab ra khỏi từng trường văn bản tự do, Azure sẽ tự động xác thực từng giá trị và đặt dấu kiểm màu xanh lục bên cạnh nó khi nó tốt. Bạn có thể di con trỏ chuột lên các chi tiết lỗi để có thêm thông tin về các vấn đề mà nó phát hiện ra.

### **Chọn một địa điểm.**

Chọn một khu vực từ danh sách sau đây khi bạn tạo tài nguyên:

- + Tây Mỹ 2
- + Nam Trung Mỹ
- + Trung Mỹ
- + Đông Mỹ
- + Tây Âu
- + Đông Nam Á**
- + East Nhật Bản
- + Nam Brazil
- + Đông Nam Úc

## + Trung Ân

1. Đặt tùy chọn sẵn có thành Không cần dự phòng cơ sở hạ tầng. Tùy chọn này có thể được sử dụng để đảm bảo VM có tính sẵn sàng cao bằng cách nhóm nhiều VM lại với nhau như một tập hợp để đối phó với các sự kiện hoặc sự cố bảo trì theo kế hoạch hoặc không có kế hoạch. Đối với bài tập này, chúng tôi sẽ không cần dịch vụ này.
2. Đảm bảo rằng hình ảnh được đặt thành Ubuntu Server 18.04 LTS. Bạn có thể mở danh sách thả xuống để xem tất cả các tùy chọn có sẵn.
3. Để trường Kích thước với mặc định của lựa chọn D2s v3, cung cấp cho bạn hai vCPU với 8 GB RAM.
4. Chuyển sang phần tài khoản Quản trị viên, đối với loại Xác thực, chọn tùy chọn Khóa chung SSH.
5. Nhập tên người dùng bạn sẽ sử dụng để đăng nhập bằng SSH. Chọn một cái gì đó bạn có thể nhớ hoặc viết nó xuống.
6. Sao chép khóa SSH từ tệp khóa công khai mà bạn đã tạo ở đơn vị trước đó và dán nó vào trường khóa công khai SSH.

### **Lưu ý quan trọng**

1. Khi bạn sao chép khóa chung vào cổng thông tin Azure, hãy đảm bảo không thêm bất kỳ ký tự khoảng trắng hoặc nguồn cấp dữ liệu bổ sung nào.
2. Trong phần RULES PORT INBOUND, trước tiên, chọn Cho phép các cổng được chọn. Vì đây là máy ảo Linux, chúng tôi muốn có thể truy cập VM bằng SSH từ xa. Cuộn danh sách Chọn cổng vào nếu cần cho đến khi bạn tìm thấy SSH (22) và bật nó.

**INBOUND PORT RULES**

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

\* Public inbound ports  None  Allow selected ports

\* Select inbound ports

**⚠️** These ports will be exposed to the internet. Use the Advanced controls to limit inbound traffic to known IP addresses. You can also update inbound traffic rules later.

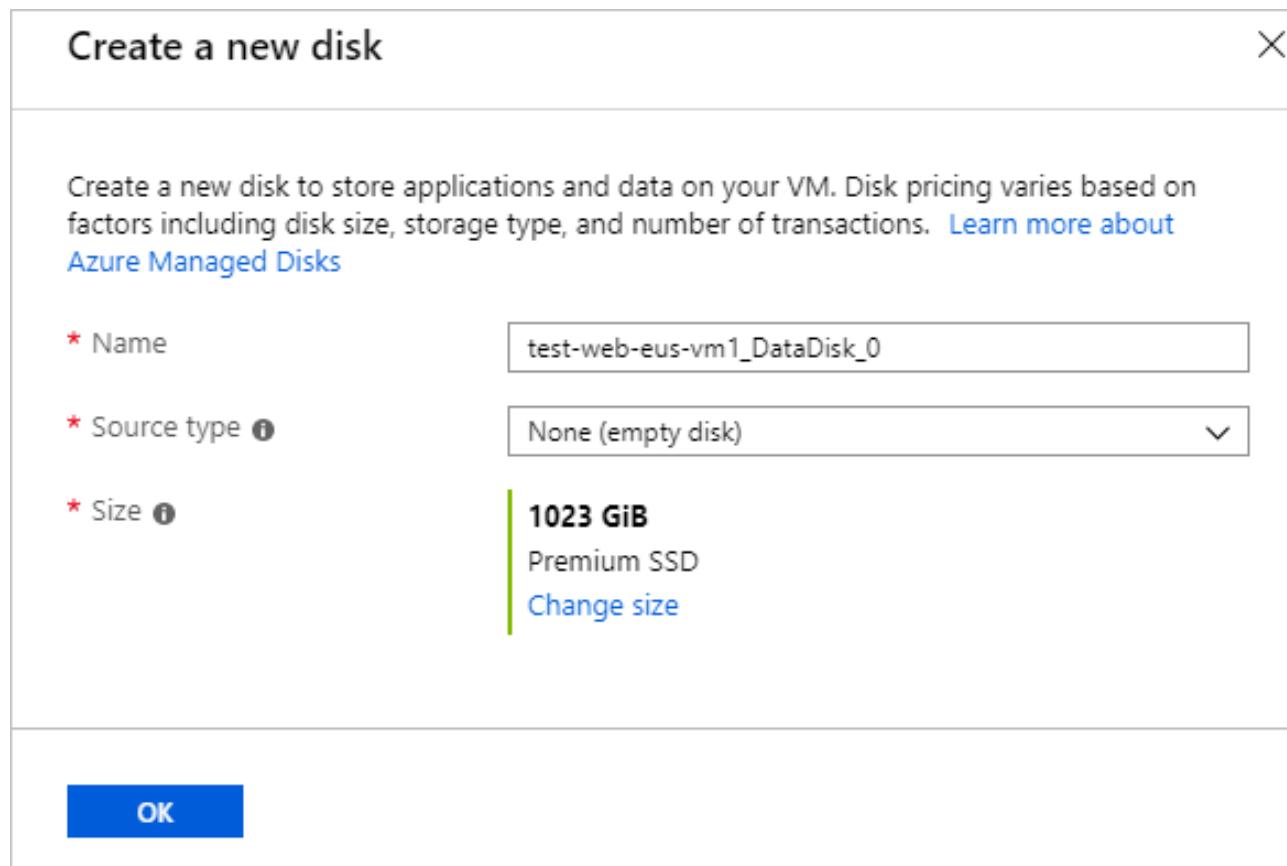
## Cấu hình đĩa cho VM

1. Nhấp Tiếp theo: Đĩa > để di chuyển đến phần Đĩa.
2. Chọn SSD cao cấp cho loại đĩa hệ điều hành.

## Tạo một đĩa dữ liệu

Hãy nhớ rằng chúng ta sẽ nhận được một đĩa hệ điều hành (/dev/sda) và một đĩa tạm thời (/dev/sdb). Hãy thêm một đĩa dữ liệu:

- Nhập vào Tạo và đính kèm một liên kết đĩa mới trong phần Đĩa dữ liệu.



- Bạn có thể lấy tất cả các giá trị mặc định: SSD cao cấp, tên được tạo tự động, kích thước 1023 GiB và Không có (đĩa trống) cho loại Nguồn, mặc dù lưu ý rằng loại nguồn là nơi bạn có thể sử dụng bộ lưu trữ snapshot hoặc Azure Blob để tạo một VHD.
- Nhập OK để tạo đĩa và quay lại phần Đĩa dữ liệu.
- Bây giờ sẽ có một đĩa mới trong hàng đầu tiên.

The screenshot shows the 'DATA DISKS' configuration page. It lists a single disk entry: LUN 0, NAME test-web-eus-vm1\_DataDisk\_0, SIZE (GiB) 1023, DISK TYPE Premium SSD, and HOST CACHING None. Below the table are links for 'Create and attach a new disk' and 'Attach an existing disk'.

## Cấu hình mạng (Network)

1. Nhập Next: Network > để chuyển đến phần Network.
2. Trong môi trường sản xuất nơi chúng tôi đã có các thành phần khác, bạn muốn sử dụng một mạng ảo hiện có. Bằng cách đó, VM của bạn có thể giao tiếp với các dịch vụ đám mây khác trong giải pháp của bạn. Nếu chưa có một định nghĩa nào ở vị trí này, bạn có thể tạo nó ở đây và định cấu hình:

Không gian địa chỉ: Không gian IPV4 tổng thể có sẵn cho mạng này.

Mạng con: Mạng con đầu tiên chia nhỏ không gian địa chỉ - nó phải nằm trong không gian địa chỉ được xác định. Khi VNet được tạo, bạn có thể thêm các mạng con bổ sung.

### Lưu ý

Theo mặc định, Azure sẽ tạo một mạng ảo, giao diện mạng và IP công cộng cho VM của bạn. Việc thay đổi các tùy chọn kết nối mạng sau khi VM được tạo không phải là chuyện nhỏ, vì vậy hãy luôn kiểm tra kỹ các bài tập mạng trên các dịch vụ bạn tạo trong Azure. Đối với bài tập này, mặc định sẽ hoạt động tốt.

### Kết thúc cấu hình VM và tạo ảnh

Phần còn lại của các tùy chọn có mặc định hợp lý và không cần thay đổi bất kỳ tùy chọn nào. Bạn có thể khám phá các tab khác nếu bạn thích. Các tùy chọn riêng lẻ có biểu tượng (i) bên cạnh chúng sẽ hiển thị mẹo trợ giúp để giải thích tùy chọn. Đây là một cách tuyệt vời để tìm hiểu về các tùy chọn khác nhau mà bạn có thể sử dụng để định cấu hình VM:

1. Nhấp vào nút Xem lại + tạo ở cuối bảng điều khiển.
2. Hệ thống sẽ xác thực các tùy chọn của bạn và cung cấp cho bạn thông tin chi tiết về VM được tạo.
3. Nhấp vào Tạo để tạo và triển khai VM. Bảng điều khiển Azure sẽ hiển thị VM đang được triển khai. Việc này có thể mất vài phút.

Trong khi triển khai, chúng ta hãy xem những gì chúng ta có thể làm với VM này.

### Azure máy ảo Địa chỉ IP và tùy chọn SSH

Bạn đã tạo một máy ảo Linux trong Azure. Việc tiếp theo bạn sẽ làm là cấu hình nó cho các tác vụ chúng ta muốn chuyển sang Azure.

Trừ khi bạn đã thiết lập VPN site-to-site sang Azure, các máy ảo Azure của bạn sẽ giành được truy cập mạng từ mạng cục bộ của bạn. Nếu bạn chỉ mới bắt đầu với Azure, thì bạn không chắc rằng bạn có VPN site-to-site hoạt động. Vậy làm thế nào bạn có thể kết nối với máy ảo?

### Địa chỉ IP Azure VM

Như chúng ta đã thấy trước đây, máy ảo Azure giao tiếp trên mạng ảo. Họ cũng có thể có một địa chỉ IP công cộng tùy chọn được gán cho họ. Với IP công cộng, chúng ta có thể tương tác với VM qua Internet. Ngoài ra, chúng tôi có thể thiết lập mạng riêng ảo (VPN) kết nối mạng tại chỗ của chúng tôi với Azure - cho phép chúng tôi kết nối an toàn với VM mà không cần lộ IP công khai. Cách tiếp cận này được đề cập trong một mô-đun khác và được ghi lại đầy đủ nếu bạn muốn khám phá tùy chọn đó.

Địa chỉ IP công cộng trong Azure được phân bổ động theo mặc định. Điều đó có nghĩa là địa chỉ IP có thể thay đổi theo thời gian - đối với VM, việc gán địa chỉ IP xảy ra khi VM được khởi động lại. Bạn có thể trả nhiều tiền hơn để gán địa chỉ tĩnh, nếu bạn muốn kết nối trực tiếp với địa chỉ IP và cần đảm bảo rằng địa chỉ IP sẽ không thay đổi.

Thừa nhận những hạn chế này và các lựa chọn thay thế được mô tả ở trên, chúng tôi sẽ sử dụng địa chỉ IP công cộng của VM trong mô-đun này.

### Kết nối với VM bằng SSH

Để kết nối với VM thông qua SSH, bạn cần:

- địa chỉ IP công cộng của VM
- tên người dùng của tài khoản cục bộ trên VM
- khóa công khai được định cấu hình trong tài khoản đó
- truy cập vào khóa riêng tương ứng
- cổng 22 mở trên VM

Trước đây, bạn đã tạo một cặp khóa SSH và thêm khóa chung vào cấu hình VM và đảm bảo rằng cổng 22 đã mở.

Trong đơn vị tiếp theo, bạn sẽ sử dụng thông tin này để mở một thiết bị đầu cuối an toàn trên VM bằng SSH.

Khi thiết bị đầu cuối được mở, bạn có quyền truy cập vào tất cả các công cụ dòng lệnh Linux tiêu chuẩn.

## 1.11 Câu hỏi ôn tập Chương 1

1. Hãy cho biết triết lý phát triển của Hệ điều hành Linux và liệt kê các bản phân phối chính của Linux.
2. Bản phân phối của hệ điều hành Linux Ubuntu có những đặc điểm gì trong hỗ trợ người dùng, cập nhật và nâng cấp các phiên bản phần mềm.
3. Vì lý do bảo mật, bạn phải sử dụng ảnh từ Azure Marketplace chính thức khi tạo một máy ảo mới. (Đúng hay sai?)

**4.** Hiệu ứng của cài đặt bảo mật mạng mặc định cho máy ảo mới là gì?

- A.** Không được phép yêu cầu gửi đi hay gửi đi.
- B.** Yêu cầu ra nước ngoài được cho phép. Lưu lượng truy cập trong nước chỉ được phép từ trong mạng ảo.

**C.** Không có hạn chế: tất cả các yêu cầu gửi đi và gửi đi đều được cho phép.

**5.** Giả sử bạn có một số máy ảo Linux được lưu trữ trong Azure. Bạn sẽ quản trị các máy ảo này từ xa qua SSH từ ba máy chuyên dụng trong trụ sở công ty của bạn. Phương pháp xác thực nào sau đây thường được coi là thực hành tốt nhất cho tình huống này

**A.** Username and password

**B.** Private key

**C.** Private key with passphrase

**6.** Cài đặt Linux VM, kết nối bằng SSH và cấu hình máy chủ web Apache Server:

+ Tạo Linux VM trong cổng thông tin Azure, kết nối với địa chỉ IP công cộng của VM và quản lý nó bằng kết nối SSH. SSH cho phép tương tác với hệ điều hành và phần mềm trên máy ảo Linux VM, cổng thông tin Azure Portal cho phép định cấu hình phần cứng và các kết nối ảo. (*Cũng có thể sử dụng PowerShell hoặc Azure CLI, nếu môi trường dòng lệnh hoặc tập lệnh cho phép*).

**Connect to virtual machine**

test-web-eus-vm1

RDP    SSH

To connect to your virtual machine via SSH, select an IP address, optionally change the port number, and use one of the following commands:

\* IP address  
Public IP address (137.117.101.249)

\* Port number  
22

Login using VM local account  
ssh jim@137.117.101.249 

 Inbound traffic to the Public IP address may be blocked. You can update inbound port rules in the **VM Networking** page.

 You can troubleshoot VM connection issues by opening the **Diagnose and solve problems** page.

**Lấy địa chỉ IP công cộng của VM:** Trong cảng thông tin Azure, đảm bảo bảng Tổng quan cho máy ảo mà bạn đã tạo trước đó được mở. Bạn có thể tìm thấy VM trong Tất cả tài nguyên nếu bạn cần mở nó. Bảng tổng quan cho phép bạn:

- Xem nếu VM đang chạy
  - Dừng hoặc khởi động lại VM
  - Lấy địa chỉ IP công cộng của VM
  - Xem hoạt động của CPU, đĩa và mạng
2. Nhấp vào nút Kết nối ở đầu ngăn.

3. Trong bảng Kết nối với máy ảo, lưu ý cài đặt địa chỉ IP và số cổng. Trên tab SSH, bạn cũng sẽ tìm thấy lệnh bạn cần thực thi cục bộ để kết nối với VM. Sao chép lệnh vào clipboard.

## Kết nối với SSH

1. Dán lệnh từ bảng ghi tạm của bạn vào Azure Cloud Shell. Nó sẽ trông giống như mẫu dưới đây; tuy nhiên, nó sẽ có một địa chỉ IP khác (và có lẽ tên người dùng khác nếu bạn không sử dụng k24it):

```
ssh -p 22 k24it@<public-IPaddress-of-Server>
```

Lần đầu tiên ta kết nối, SSH sẽ hỏi về việc xác thực với một máy chủ không xác định. SSH đang nói với bạn rằng bạn chưa bao giờ kết nối với máy chủ này trước đây. Nếu đó là sự thật thì điều đó hoàn toàn bình thường và bạn có thể trả lời bằng cách lưu dấu vân tay của máy chủ trong tệp máy chủ đã biết:

```
The authenticity of host '137.117.101.249 (137.117.101.249)' can't be established.  
ECDSA key fingerprint is  
SHA256:w1h08h4ieIiMq7ibIVSQM/PhcXFV7O7EEhjEqhPYMWY.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '137.117.101.249' (ECDSA) to the list of known hosts.
```

1. Lệnh này sẽ mở kết nối SSH và đặt bạn tại dấu nhắc lệnh shell cho Linux.

2. Thử thực hiện một vài lệnh Linux

- ls -la / để hiển thị thư mục gốc của đĩa
- ps -l để hiển thị tất cả các quy trình đang chạy

Dmesg để liệt kê tất cả các thông điệp kernel

Lsblk để liệt kê tất cả các thiết bị chặn - ở đây bạn sẽ thấy các ổ đĩa của mình

Điều thú vị hơn để quan sát trong danh sách các ổ đĩa là những gì còn thiếu. Lưu ý rằng ổ dữ liệu của chúng tôi (sdc) có mặt nhưng không được gắn vào hệ thống tệp. Azure đã thêm một VHD nhưng không khởi tạo nó.

## Khởi tạo đĩa dữ liệu

Bất kỳ ổ đĩa bổ sung nào bạn tạo từ đầu cần phải được khởi tạo và định dạng. Quá trình khởi tạo giống hệt với đĩa vật lý:

1. Đầu tiên, xác định đĩa. Chúng tôi đã làm điều đó ở trên. Bạn cũng có thể sử dụng dmesg | grep SCSI, sẽ liệt kê tất cả các thông báo từ kernel cho các thiết bị SCSI.

2. Khi bạn biết ổ đĩa (sdc) bạn cần khởi tạo, bạn có thể sử dụng fdisk để làm điều đó. Bạn sẽ cần chạy lệnh với sudo và cung cấp đĩa bạn muốn phân vùng. Chúng ta có thể sử dụng lệnh sau để tạo phân vùng chính mới:

```
sudo fdisk /dev/sdc
```

Tiếp theo, chúng ta cần viết một hệ thống tệp vào phân vùng bằng lệnh mkfs.

```
sudo mkfs -t ext4 /dev/sdc1
```

Cuối cùng, chúng ta cần gắn ổ đĩa vào hệ thống tập tin. Giả sử chúng ta sẽ có một thư mục dữ liệu. Hãy tạo thư mục điểm gắn kết và gắn ổ đĩa.

```
sudo mkdir /data & sudo mount /dev/sdc1 /data
```

Chúng tôi đã khởi tạo đĩa và gắn nó. Nếu bạn quan tâm đến nhiều chi tiết hơn về quy trình này, hãy xem qua các đĩa Thêm và kích thước trong mô-đun máy ảo Azure. Nhiệm vụ này được đề cập chi tiết hơn ở đó.

### **Cài đặt phần mềm máy chủ apache webserver lên Linux VM**

Như bạn có thể thấy, SSH cho phép bạn làm việc với Linux VM giống như một máy tính cục bộ. Bạn có thể quản trị VM này như bất kỳ máy tính Linux nào khác: cài đặt phần mềm, định cấu hình vai trò, điều chỉnh các tính năng và các tác vụ hàng ngày khác. Hãy tập trung vào việc cài đặt phần mềm một lát.

Bạn cũng có thể cài đặt phần mềm từ internet khi bạn được kết nối với VM thông qua SSH. Theo mặc định, các máy Azure được kết nối internet. Bạn có thể sử dụng các lệnh tiêu chuẩn để cài đặt các gói phần mềm phổ biến trực tiếp từ kho lưu trữ tiêu chuẩn. Hãy sử dụng phương pháp này để cài đặt Apache.

### **Cài đặt máy chủ web Apache**

Phần mềm máy chủ web Apache có sẵn trong kho phần mềm mặc định của Ubuntu, vì vậy chúng ta sẽ cài đặt nó bằng các công cụ quản lý gói thông thường:

1. Bắt đầu bằng cách cập nhật chỉ mục gói cục bộ để phản ánh các thay đổi ngược dòng mới nhất:

```
sudo apt-get update
```

2. Sau đó, cài đặt gói phần mềm apache2

```
sudo apt-get install apache2
```

3. Quá trình cài đặt sẽ diễn ra tự động. Sau đó chúng ta có thể kiểm tra bằng lệnh

```
sudo systemctl status apache2 -no-pager
```

Lệnh systemctl sẽ trả lại kết quả tương tự như sau:

```
apache2.service - The Apache HTTP Server
```

```
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
```

```
  Drop-In: /lib/systemd/system/apache2.service.d
```

```
    └─apache2-systemd.conf
```

```
  Active: active (running) since Mon 2018-09-03 21:00:03  
  UTC; 1min 34s ago
```

```
  Main PID: 11156 (apache2)
```

```
  Tasks: 55 (limit: 4915)
```

```
  CGroup: /system.slice/apache2.service
```

```
    ├─11156 /usr/sbin/apache2 -k start
```

```
    ├─11158 /usr/sbin/apache2 -k start
```

```
    └─11159 /usr/sbin/apache2 -k start
```

```
test-web-eus-vm1 systemd[1]: Starting The Apache HTTP Server...
```

```
test-web-eus-vm1 systemd[1]: Started The Apache HTTP Server.
```

### **Lưu ý**

Việc thực hiện các lệnh như thế này là không quan trọng, tuy nhiên đó là một quy trình thủ công - nếu chúng ta luôn cần cài đặt một số phần mềm, bạn có thể xem xét tự động hóa quy trình bằng cách sử dụng tập lệnh.

Cuối cùng, chúng tôi có thể thử truy xuất trang mặc định thông qua địa chỉ IP công cộng. Tuy nhiên, mặc dù máy chủ web đang chạy trên VM, bạn sẽ không nhận được kết nối hoặc phản hồi hợp lệ. Bạn có biết tại sao?

Chúng ta cần thực hiện thêm một bước nữa để có thể tương tác với máy chủ web. Mạng ảo của chúng tôi đang chặn yêu cầu gửi đến. Chúng ta có thể thay đổi điều đó thông qua cấu hình. Hãy xem xét cho phép yêu cầu gửi đến tiếp theo.

Việc điều chỉnh cấu hình máy chủ thường được thực hiện với thiết bị trong môi trường tại chỗ của bạn. Theo nghĩa này, bạn có thể coi máy ảo Azure là một phần mở rộng của môi trường đó. Bạn có thể thay đổi cấu hình, quản lý mạng, mở hoặc chặn lưu lượng truy cập và hơn thế nữa thông qua cổng thông tin Azure, công cụ Azure CLI hoặc Azure PowerShell.

Chúng tôi đã chạy máy chủ của chúng tôi và Apache được cài đặt và phục vụ các trang. Nhóm bảo mật của chúng tôi yêu cầu chúng tôi khóa tất cả các máy chủ của mình và chúng tôi chưa làm gì với VM này. Chúng tôi không làm gì cả và nó cho phép Apache nghe trên cổng 80. Hãy khám phá cấu hình mạng Azure để xem cách sử dụng hỗ trợ bảo mật tích hợp để làm cứng máy chủ của chúng tôi.

### **Mở cổng trong máy ảo Azure**

Theo mặc định, các máy ảo mới bị khóa.

Ứng dụng có thể thực hiện các yêu cầu gửi đi, nhưng lưu lượng truy cập trong nước duy nhất được phép là từ mạng ảo (ví dụ: các tài nguyên khác trên cùng một mạng cục bộ) và từ Azure Load Balancer (kiểm tra thăm dò).

Có hai bước để điều chỉnh cấu hình để hỗ trợ các giao thức khác nhau trên mạng. Khi bạn tạo một VM mới, bạn có cơ hội mở một vài cổng phổ biến (RDP, HTTP, HTTPS và SSH). Tuy nhiên, nếu bạn yêu cầu các thay đổi khác đối với tường lửa, bạn sẽ cần điều chỉnh chúng theo cách thủ công.

Quá trình này bao gồm hai bước:

1. Tạo một nhóm bảo mật mạng.
2. Tạo quy tắc gửi đến cho phép lưu lượng truy cập trên các cổng bạn cần.

### **Nhóm an ninh mạng (NSG) là gì?**

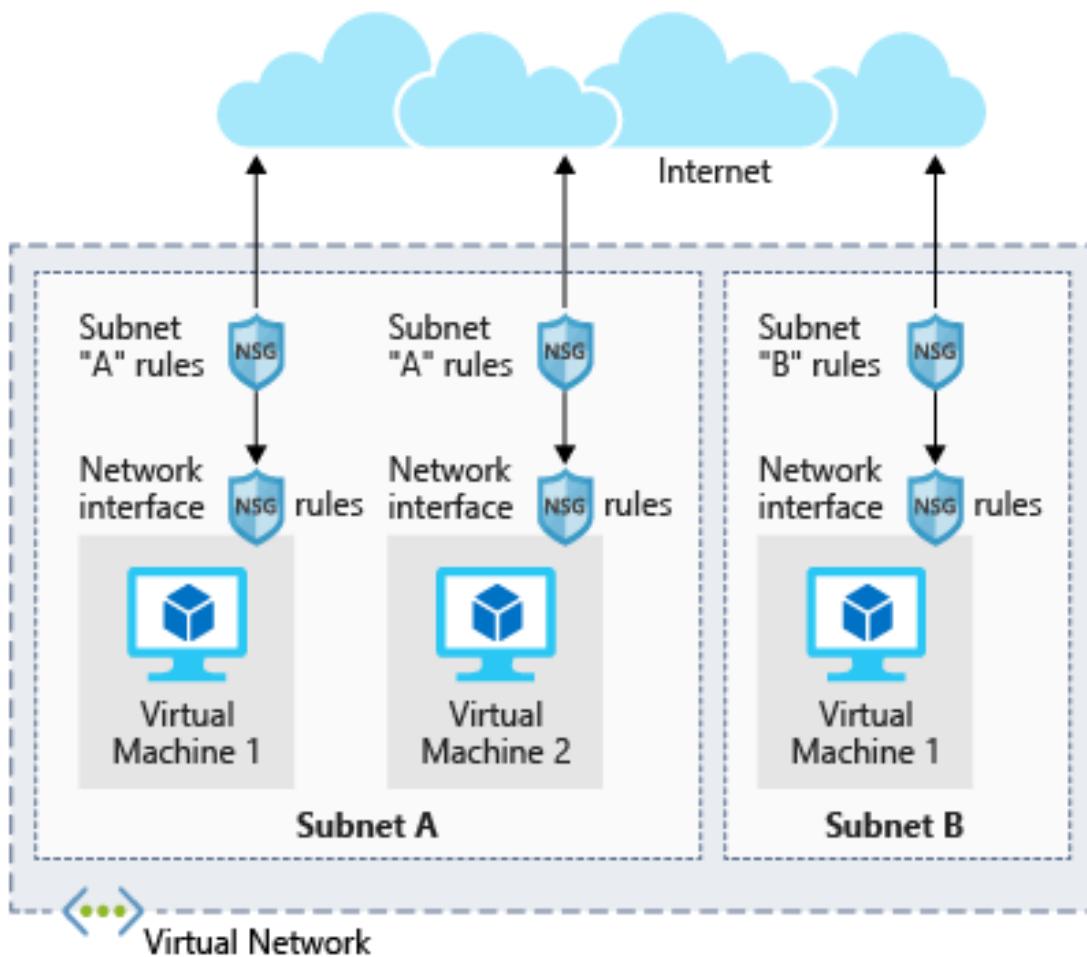
Mạng ảo (VNets) là nền tảng của mô hình mạng Azure và cung cấp sự cách ly và bảo vệ. Các nhóm bảo mật mạng (NSG) là công cụ chính bạn sử dụng để thực thi và kiểm soát các quy tắc lưu lượng mạng ở cấp độ mạng. NSG là một lớp bảo mật tùy chọn cung cấp tường lửa phần mềm bằng cách lọc lưu lượng truy cập vào và ra trên VNet.

Các nhóm bảo mật có thể được liên kết với giao diện mạng (cho mỗi quy tắc máy chủ), mạng con trong mạng ảo (để áp dụng cho nhiều tài nguyên) hoặc cả hai cấp.

### Cài đặt mạng và bảo mật trong máy ảo Linux VM

Quy tắc nhóm bảo mật

NSG sử dụng các quy tắc để cho phép hoặc từ chối lưu lượng di chuyển qua mạng. Mỗi quy tắc xác định địa chỉ nguồn và đích (hoặc phạm vi), giao thức, cổng (hoặc phạm vi), hướng (trong hoặc ngoài), mức độ ưu tiên số và cho phép hoặc từ chối lưu lượng phù hợp với quy tắc.



Mỗi nhóm bảo mật có một bộ quy tắc bảo mật mặc định để áp dụng các quy tắc mạng mặc định được mô tả ở trên. Các quy tắc mặc định này không thể được sửa đổi nhưng có thể bị ghi đè.

## Cách Azure sử dụng quy tắc mạng

Đối với lưu lượng truy cập trong nước, Azure xử lý nhóm bảo mật được liên kết với mạng con và sau đó nhóm bảo mật được áp dụng cho giao diện mạng. Lưu lượng ra được xử lý theo thứ tự ngược lại (giao diện mạng trước, sau đó là mạng con).

## Cảnh báo

Hãy nhớ rằng các nhóm bảo mật là tùy chọn ở cả hai cấp độ. Nếu không có nhóm bảo mật nào được áp dụng, thì tất cả lưu lượng truy cập được Azure cho phép. Nếu VM có IP công cộng, điều này có thể là một rủi ro nghiêm trọng, đặc biệt nếu HĐH không cung cấp tường lửa tích hợp.

Các quy tắc được đánh giá theo thứ tự ưu tiên, bắt đầu với quy tắc ưu tiên thấp nhất. Quy tắc từ chối luôn dừng đánh giá. Ví dụ: nếu quy tắc giao diện mạng chặn yêu cầu gửi đi, mọi quy tắc được áp dụng cho mạng con sẽ không được kiểm tra. Để lưu lượng được phép thông qua nhóm bảo mật, nó phải đi qua tất cả các nhóm được áp dụng.

Quy tắc cuối cùng luôn luôn là quy tắc Từ chối Tất cả. Đây là quy tắc mặc định được thêm vào mọi nhóm bảo mật cho cả lưu lượng truy cập trong và ngoài nước với mức độ ưu tiên là 65500. Điều đó có nghĩa là để lưu lượng truy cập đi qua nhóm bảo mật, bạn phải có quy tắc cho phép hoặc quy tắc mặc định cuối cùng sẽ chặn quy tắc này.

## Lưu ý

SMTP (cổng 25) là một trường hợp đặc biệt. Tùy thuộc vào cấp độ đăng ký của bạn và khi tài khoản của bạn được tạo, lưu lượng truy cập SMTP đi có thể bị chặn. Bạn có thể yêu cầu loại bỏ hạn chế này với lý do kinh doanh.

## Tạo các nhóm bảo mật mạng

Các nhóm bảo mật được quản lý tài nguyên như hầu hết mọi thứ trong Azure. Bạn có thể tạo chúng trong cổng thông tin Azure hoặc thông qua các công cụ kịch bản dòng lệnh. Thách thức là trong việc xác định các quy tắc. Hãy xem xét việc xác định quy tắc mới để cho phép truy cập HTTP và chặn mọi thứ khác.

## Cấu hình cài đặt mạng

Khi tạo máy ảo (VM), ta đã chọn SSH cổng vào để có thể kết nối với VM. Điều này tạo ra một NSG được gắn vào giao diện mạng của VM. NSG đó đang chặn lưu lượng HTTP. Hãy cập nhật NSG này để cho phép lưu lượng HTTP gửi đến trên cổng 80.

## Cập nhật NSG trên giao diện mạng

Cổng 80 được mở trên NSG được áp dụng cho mạng con. Nhưng cổng 80 bị chặn bởi NSG áp dụng cho giao diện mạng. Hãy sửa nó để chúng ta có thể kết nối với trang web.

- Quay trở lại bảng Tổng quan cho máy ảo. Bạn có thể tìm thấy VM trong Tất cả tài nguyên.
- Trong phần Cài đặt, chọn mục Mạng.
- Bạn sẽ thấy các quy tắc NSG cho mạng con trong phần trên cùng và quy tắc NSG cho giao diện mạng trong phần dưới cùng của tab. Trong phần dưới cùng, đối với quy tắc NSG cho giao diện mạng, chọn Thêm quy tắc cổng vào.

**Network Interface: test-web-eus-vm1268** Effective security rules Topology  
Virtual network/subnet: Learn-f0d11fc9-d764-44a1-b9ec-c5aabb999d6b-vnet/default NIC Public IP: **40.118.131.182** NIC Private IP: **10.0.0.4** Accelerated networking: **Disabled**

Inbound port rules Outbound port rules Application security groups Load balancing

Network security group **NSG-westus** (attached to subnet: default)  
Impacts 1 subnets, 0 network interfaces Add inbound port rule

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
100	Allow-VirtualHostIP-Inbound	Any	Any	168.63.129.16,169.2...	Any	Allow
500	Allow-HTTP-Inbound	80,443,8080	Any	Any	Any	Allow
501	Allow-SSH-Inbound	22	TCP	Any	Any	Allow
503	Allow-RPD-Inbound	3389	TCP	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

Network security group **test-web-eus-vm1-nsg** (attached to network interface: test-web-eus-vm1268)  
Impacts 0 subnets, 1 network interfaces Add inbound port rule

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
300	SSH	22	TCP	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

1. Chuyển sang chế độ Cơ bản.

2. Thêm thông tin cho quy tắc HTTP của chúng tôi:

- + Đặt Dịch vụ thành HTTP. Điều này thiết lập phạm vi cổng của bạn.
- + Đặt mức độ ưu tiên thành 310.
- + Đặt tên cho quy tắc; sử dụng cho phép lưu lượng truy cập http.
- + Dưa ra quy tắc mô tả.

3. Nhấp vào Thêm để tạo quy tắc.

Mở trang web mặc định

Sử dụng địa chỉ IP của máy chủ để thực hiện yêu cầu HTTP. Nó sẽ hoạt động.

The screenshot shows a Microsoft Edge browser window with the address bar set to 137.117.101.249. The main content is the Apache2 Ubuntu Default Page. At the top left is the Ubuntu logo. The title "Apache2 Ubuntu Default Page" is centered above a red banner with the text "It works!". Below the banner, a paragraph explains the purpose of the page. Another paragraph below it states: "If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator." A section titled "Configuration Overview" follows, containing text about the configuration system. To the right of this text is a code block showing the directory structure of /etc/apache2:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   '-- *.Load
|   '-- *.conf
|-- conf-enabled
|   '-- *.conf
|-- sites-enabled
|   '-- *.conf
```

Below the code block is a list of bullet points explaining the structure:

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default

## Lưu ý thêm

Luôn đảm bảo khóa các cổng được sử dụng để truy cập quản trị. Một cách tiếp cận thậm chí tốt hơn là tạo VPN để liên kết mạng ảo với mạng riêng của bạn và chỉ cho phép các yêu cầu RDP hoặc SSH từ phạm vi địa chỉ đó. Bạn cũng có thể thay đổi cổng được sử dụng bởi SSH thành một cái gì đó khác với mặc định. Hãy nhớ rằng việc thay đổi cổng là không đủ để ngăn chặn các cuộc tấn công. Nó chỉ đơn giản là làm cho nó khó khăn hơn một chút để khám phá

## Chương 2 Sử dụng Hệ điều hành Linux

### 2.1 Khởi động Linux

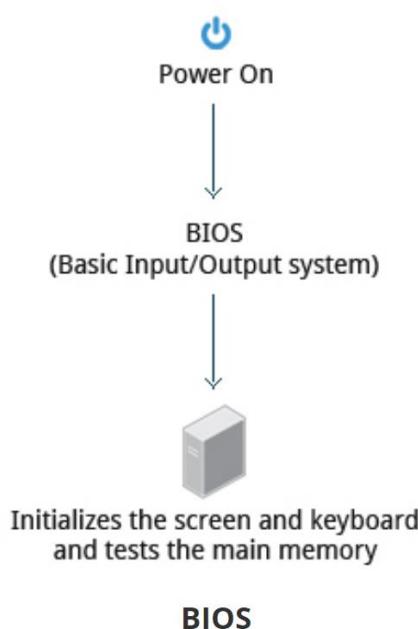
Quá trình khởi động Linux là thủ tục khởi tạo hệ thống. Nó bao gồm mọi thứ xảy ra từ khi bật nguồn máy tính lần đầu tiên cho đến khi giao diện người dùng hoạt động đầy đủ.

Hiểu rõ các bước trong quy trình khởi động có thể giúp bạn khắc phục các sự cố cũng như điều chỉnh hiệu năng của máy tính theo nhu cầu của bạn.

Mặt khác, quá trình khởi động có thể khá kỹ thuật và bạn có thể bắt đầu sử dụng Linux mà không cần biết tất cả các chi tiết.

### 2.2 Cấu hình BIOS

Bắt đầu một hệ thống Linux dựa trên x86 bao gồm một số bước. Khi máy tính được bật nguồn, Hệ thống đầu vào / đầu ra cơ bản (BIOS) sẽ khởi tạo phần cứng, bao gồm màn hình và bàn phím và kiểm tra bộ nhớ chính. Quá trình này cũng được gọi là POST (Tự kiểm tra khi bật nguồn).

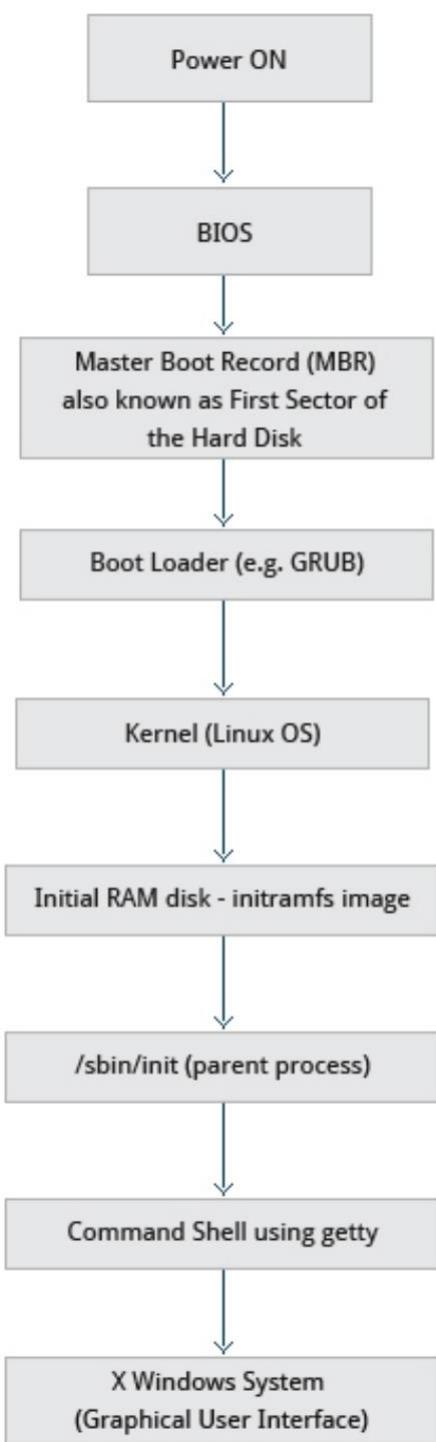


Phần mềm BIOS được lưu trữ trên chip ROM trên bo mạch chủ. Sau này, phần còn lại của quá trình khởi động được điều khiển bởi hệ điều hành (HĐH).

### 2.3 Master Boot Record (MBR) và Boot Loader

Khi POST hoàn tất, điều khiển hệ thống chuyển từ BIOS sang bộ tải khởi động. Bộ tải khởi động thường được lưu trữ trên một trong các đĩa cứng

trong hệ thống, trong khu vực khởi động (đối với hệ thống BIOS / MBR truyền thống)



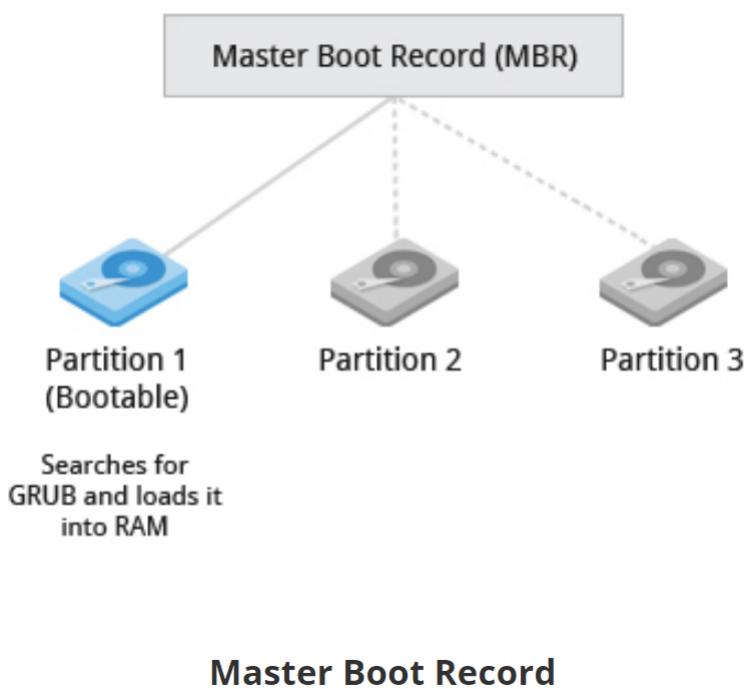
### Quá trình khởi động

hoặc phân vùng EFI (đối với Giao diện phần mềm mở rộng (Thống nhất) gần đây hoặc hệ thống EFI / UEFI). Cho đến giai đoạn này, máy không truy cập bất kỳ phương tiện lưu trữ lớn. Sau đó, thông tin về ngày, giờ và các thiết bị ngoại vi quan trọng nhất được tải từ các giá trị CMOS (sau khi công nghệ được sử dụng cho bộ nhớ lưu trữ chạy bằng pin cho phép hệ thống theo dõi ngày và giờ ngay cả khi tắt nguồn) .

Một số bộ tải khởi động tồn tại cho Linux; những cái phổ biến nhất là GRUB (cho bộ tải khởi động thống nhất GRand), ISOLINUX (để khởi động từ phương tiện di động) và DAS U-Boot (để khởi động trên các thiết bị / thiết bị nhúng). Hầu hết các bộ tải khởi động Linux có thể hiển thị giao diện người dùng để chọn các tùy chọn thay thế để khởi động Linux và thậm chí các hệ điều hành khác có thể được cài đặt. Khi khởi động Linux, bộ tải khởi động có trách nhiệm tải hình ảnh hạt nhân và đĩa RAM hoặc hệ thống tập tin ban đầu (chứa một số tệp quan trọng và trình điều khiển thiết bị cần thiết để khởi động hệ thống) vào bộ nhớ.

## 2.4 Hoạt động của Boot Loader

Bộ tải khởi động có hai giai đoạn riêng biệt:



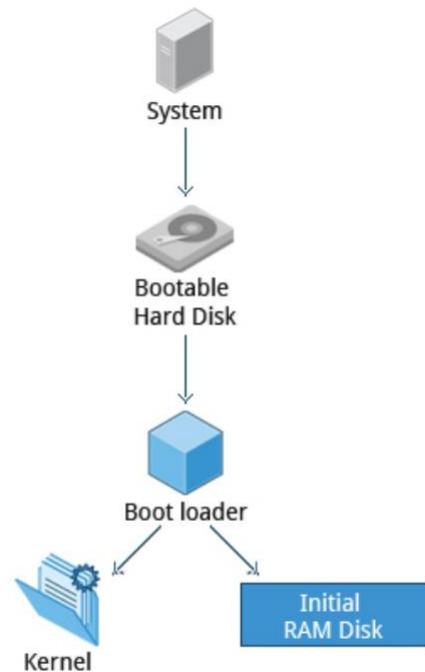
Đối với các hệ thống sử dụng phương pháp BIOS / MBR, bộ tải khởi động nằm ở khu vực đầu tiên của đĩa cứng, còn được gọi là Bản ghi khởi động chính (MBR). Kích thước của MBR chỉ là 512 byte. Trong giai đoạn này, bộ tải khởi động kiểm tra bảng phân vùng và tìm phân vùng khởi động. Khi tìm thấy phân vùng có thể khởi động, nó sẽ tìm kiếm bộ tải khởi động giai đoạn thứ hai, ví dụ GRUB và tải nó vào RAM (Bộ nhớ truy cập ngẫu nhiên). Đối với

các hệ thống sử dụng phương pháp EFI / UEFI, phần sụn UEFI đọc dữ liệu Trình quản lý khởi động của nó để xác định ứng dụng UEFI nào sẽ được khởi chạy và từ đó (nghĩa là có thể tìm thấy phân vùng EFI từ đĩa nào và phân vùng). Phần sụn sau đó khởi chạy ứng dụng UEFI, ví dụ GRUB, như được xác định trong mục khởi động trong trình quản lý khởi động của phần sụn. Quy trình này phức tạp hơn, nhưng linh hoạt hơn các phương pháp MBR cũ.

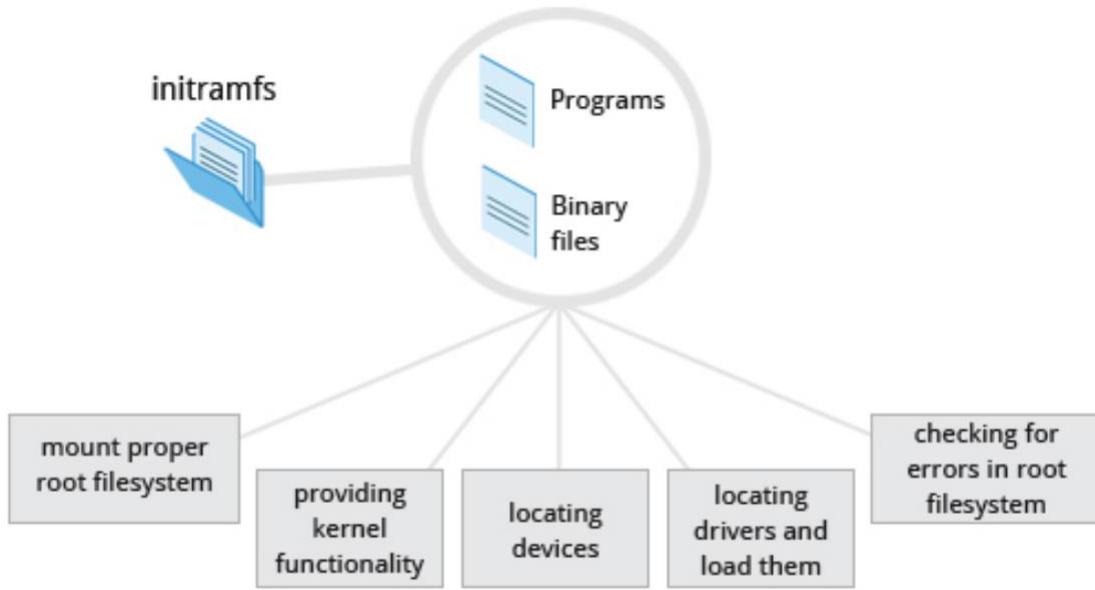
Trình tải khởi động giai đoạn thứ hai nằm dưới / boot. Một màn hình giật gân được hiển thị, cho phép chúng ta chọn hệ điều hành (HĐH) nào để khởi động. Sau khi chọn HĐH, bộ tải khởi động sẽ tải kernel của hệ điều hành đã chọn vào RAM và chuyển điều khiển cho nó. Bộ tải khởi động tải hình ảnh hạt nhân đã chọn và chuyển điều khiển cho nó. Hạt nhân hầu như luôn được nén, vì vậy công việc đầu tiên của nó là tự giải nén. Sau này, nó sẽ kiểm tra và phân tích phần cứng hệ thống và khởi tạo bất kỳ trình điều khiển thiết bị phần cứng nào được tích hợp trong kernel.

## 2.5 Ổ đĩa Ram khởi động (Initial RAM Disk)

Hình ảnh hệ thống tập tin initramfs chứa các chương trình và tệp nhị phân thực hiện tất cả các hành động cần thiết để gắn hệ thống tập tin gốc thích hợp, như cung cấp chức năng kernel cho trình điều khiển thiết bị và hệ thống tập tin cần thiết cho bộ điều khiển lưu trữ lớn với một cơ sở gọi là udev (cho thiết bị người dùng), chịu trách nhiệm để tìm ra thiết bị nào có mặt, định vị trình điều khiển thiết bị mà họ cần để hoạt động đúng và tải chúng. Sau khi tìm thấy hệ thống tập tin gốc, nó được kiểm tra lỗi và gắn kết.



### Boot Loader in Action



### The Initial RAM Disk

Chương trình gắn kết hướng dẫn hệ điều hành rằng một hệ thống tệp đã sẵn sàng để sử dụng và liên kết nó với một điểm cụ thể trong hệ thống phân cấp tổng thể của hệ thống tệp (điểm gắn kết). Nếu điều này thành công, initramfs sẽ bị xóa khỏi RAM và chương trình init trên hệ thống tập tin gốc (/sbin/init) được thực thi.

init xử lý việc gắn kết và xoay vòng đến hệ thống tập tin gốc thực sự cuối cùng. Nếu cần trình điều khiển phần cứng đặc biệt trước khi có thể truy cập bộ lưu trữ lớn, chúng phải ở trong hình ảnh initramfs.

## 2.6 Đăng nhập chế độ dòng lệnh

Gần cuối quá trình khởi động, init bắt đầu một số lời nhắc đăng nhập ở chế độ văn bản. Những thứ này cho phép bạn nhập tên người dùng, theo sau là mật khẩu của bạn và cuối cùng nhận được một vỏ lệnh. Tuy nhiên, nếu bạn đang chạy một hệ thống có giao diện đăng nhập đồ họa, ban đầu bạn sẽ không nhìn thấy những hệ thống này.

Như bạn sẽ tìm hiểu : Hoạt động dòng lệnh, các thiết bị đầu cuối chạy shell lệnh có thể được truy cập bằng phím ALT cộng với phím chức năng. Hầu hết các bản phân phối bắt đầu sáu thiết bị đầu cuối văn bản và một thiết bị đầu cuối đồ họa bắt đầu bằng F1 hoặc F2. Trong môi trường đồ họa, chuyển sang bảng điều khiển văn bản yêu cầu nhấn CTRL-ALT + phím chức năng phù hợp (với F7 hoặc F1 dẫn đến GUI).

Thông thường, shell lệnh mặc định là bash (GNU Bourne Again Shell), nhưng có một số shell lệnh nâng cao khác có sẵn. Shell in một dấu nhắc văn bản, cho biết nó đã sẵn sàng để chấp nhận các lệnh; sau khi người dùng gõ lệnh và nhấn Enter, lệnh được thực thi và một dấu nhắc khác được hiển thị sau khi lệnh được thực hiện.

## 2.7 Nhân Linux

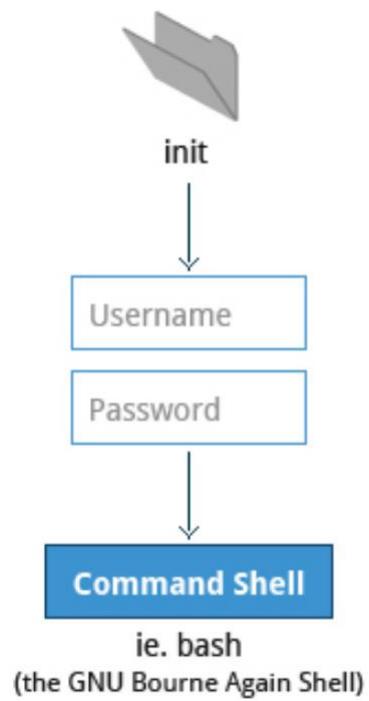
Bộ tải khởi động tải cả kernel và hệ thống tệp dựa trên RAM (initramfs) ban đầu vào bộ nhớ, do đó, nó có thể được sử dụng trực tiếp bởi kernel.

Khi kernel được tải trong RAM, nó sẽ ngay lập tức khởi tạo và cấu hình bộ nhớ máy tính và cũng cấu hình tất cả phần cứng được gắn vào hệ thống. Điều này bao gồm tất cả các bộ xử lý, hệ thống con I / O, thiết bị lưu trữ, v.v. Hạt nhân cũng tải một số ứng dụng không gian người dùng cần thiết.

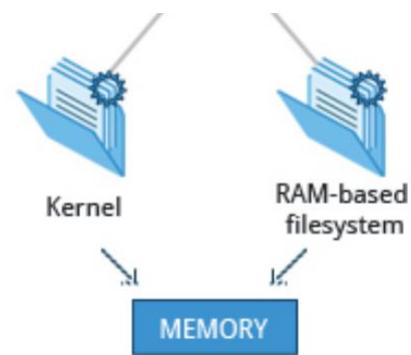
## 2.8 /sbin/init và các dịch vụ Hệ điều hành

Khi kernel đã thiết lập tất cả phần cứng của nó và gắn hệ thống tập tin gốc, kernel sẽ chạy / sbin / init. Điều này sau đó trở thành quy trình ban đầu, sau đó bắt đầu các quy trình khác để hệ thống chạy. Hầu hết các quá trình khác trên hệ thống theo dõi nguồn gốc của chúng cuối cùng là init; các ngoại lệ bao gồm các quá trình nhân được gọi là. Chúng được khởi động bởi kernel trực tiếp và công việc của chúng là quản lý chi tiết hệ điều hành nội bộ.

Bên cạnh việc khởi động hệ thống, init có trách nhiệm giữ cho hệ thống hoạt động và tắt nó sạch sẽ. Một trong những trách nhiệm của nó là hành động khi cần thiết như một người quản lý cho tất cả các quy trình không phải kernel; nó dọn sạch sau khi chúng hoàn thành và khởi động lại dịch vụ đăng nhập của người dùng khi cần thiết khi người dùng đăng nhập và đăng xuất, và thực hiện tương tự cho các dịch vụ hệ thống nền khác.



## Text-Mode Logins



Theo truyền thống, quá trình khởi động này được thực hiện bằng cách sử dụng các quy ước có từ những năm 1980 và hệ thống UNIX của System V. Quá trình nối tiếp này có hệ thống đi qua một chuỗi các đường băng chứa các tập lệnh bắt đầu và dừng dịch vụ. Mỗi runlevel hỗ trợ một chế độ chạy hệ thống khác nhau. Trong mỗi runlevel, các dịch vụ riêng lẻ có thể được đặt để chạy hoặc tắt nếu chạy.

Tuy nhiên, tất cả các bản phân phối chính gần đây đã tránh xa phương thức khởi tạo hệ thống tuần tự này, mặc dù chúng thường hỗ trợ các quy ước System V cho mục đích tương thích. Tiếp theo, chúng tôi thảo luận về các phương thức mới hơn, systemd và Upstart.

## 2.9 Các quy trình khởi động Linux tương tự

SysVinit xem mọi thứ như một quá trình nối tiếp, được chia thành một chuỗi các giai đoạn liên tiếp. Mỗi giai đoạn cần hoàn thành trước khi tiếp theo có thể tiến hành. Do đó, startup không dễ dàng tận dụng lợi thế của xử lý song song có thể được thực hiện trên nhiều bộ xử lý hoặc lỗi.

Hơn nữa, tắt máy và khởi động lại được coi là một sự kiện tương đối hiếm; chính xác mất bao lâu không được coi là quan trọng. Điều này không còn đúng nữa, đặc biệt là với các thiết bị di động và các hệ thống nhúng Linux. Một số phương pháp hiện đại, như sử dụng container, có thể yêu cầu thời gian khởi động gần như tức thời. Do đó, các hệ thống hiện yêu cầu các phương thức với khả năng nhanh hơn và nâng cao. Cuối cùng, các phương thức cũ hơn yêu cầu các tập lệnh khởi động khá phức tạp, rất khó để giữ phổ quát trên các phiên bản phân phối, phiên bản kernel, kiến trúc và các loại hệ thống. Hai lựa chọn chính được phát triển là:

### + *Upstart*

Được phát triển bởi Ubuntu và lần đầu tiên được đưa vào năm 2006

Được thông qua trong Fedora 9 (năm 2008) và trong RHEL 6 và bản sao của nó.

### hệ thống

### + *Systemd*

Được Fedora thông qua đầu tiên (năm 2011)

Được thông qua bởi RHEL 7 và SUSE

Thay thế khởi động trong Ubuntu 16.04.

Mặc dù việc chuyển sang systemd khá gây tranh cãi, nhưng nó đã được các bản phân phối chính chấp nhận và vì vậy chúng tôi sẽ không thảo luận về phương pháp System V cũ hoặc Upstart, đã trở thành ngô cụt. Bất kể người ta cảm thấy thế nào về các tranh cãi hoặc phương pháp kỹ thuật của systemd, việc áp dụng gần như phổ biến đã làm cho việc học

cách làm việc trên các hệ thống Linux trở nên đơn giản hơn, vì có ít sự khác biệt hơn giữa các bản phân phối. Chúng tôi liệt kê các tính năng systemd tiếp theo.

## 2.10 Các tính năng của systemd

Các hệ thống có systemd khởi động nhanh hơn các hệ thống có phương thức init trước đó. Điều này phần lớn là do nó thay thế một tập hợp các bước nối tiếp bằng các kỹ thuật song song tích cực, cho phép nhiều dịch vụ được khởi tạo đồng thời.

Các tập lệnh shell khởi động phức tạp được thay thế bằng các tệp cấu hình đơn giản hơn, liệt kê những gì phải làm trước khi bắt đầu dịch vụ, cách thực hiện khởi động dịch vụ và những điều kiện nào dịch vụ cần chỉ ra khi khởi động kết thúc. Một điều cần lưu ý là /sbin /init bây giờ chỉ trỏ đến /lib /systemd /systemd; tức là systemd tiếp quản quá trình init.

Một lệnh systemd (systemctl) được sử dụng cho hầu hết các tác vụ cơ bản. Mặc dù chúng tôi chưa nói về làm việc tại dòng lệnh, đây là một danh sách ngắn gọn về việc sử dụng nó: Bắt đầu, dừng, khởi động lại dịch vụ (sử dụng nfs làm ví dụ) trên hệ thống hiện đang chạy:

```
$ sudo systemctl start | stop | restart nfs.service
```

Kích hoạt hoặc vô hiệu hóa một dịch vụ hệ thống khởi động khi khởi động hệ thống:

```
$ sudo systemctl enable | vô hiệu hóa nfs.service
```

Trong hầu hết các trường hợp, dịch vụ. Có thể được bỏ qua. Có nhiều sự khác biệt về kỹ thuật với các phương pháp cũ nằm ngoài phạm vi thảo luận của chúng tôi.



## 2.11 Tìm hiểu hệ thống tập tin và thư mục trên Linux

### Linux Filesystem formats

Hãy nghĩ về một tủ lạnh có nhiều kệ có thể được sử dụng để lưu trữ các mặt hàng khác nhau. Những kệ này giúp bạn sắp xếp các mặt hàng tập hợp theo hình dạng, kích thước, loại, v.v ... Khái niệm tương tự áp dụng cho một hệ thống tập tin, là hiện thân của phương pháp lưu trữ và sắp xếp các bộ sưu tập dữ liệu tùy ý dưới dạng có thể sử dụng được.

Các loại hệ thống tập tin khác nhau được Linux hỗ trợ:

Các hệ thống tập tin đĩa thông thường: ext2, ext3, ext4, XFS, Btrfs, JFS, NTFS, v.v.

Hệ thống tập tin lưu trữ flash: ubifs, JFFS2, YAFFS, v.v.

Hệ thống tập tin cơ sở dữ liệu

Các hệ thống tập tin mục đích đặc biệt: Procfs, sysfs, tmpfs, squashfs, debugfs, v.v.

Phần này sẽ mô tả bố cục hệ thống tệp tiêu chuẩn được chia sẻ bởi hầu hết các bản phân phối Linux.



### Partitions and Filesystems

Phân vùng là một phần tiếp giáp vật lý của đĩa hoặc phân đường như là như vậy trong một số thiết lập nâng cao.

Hệ thống tệp là phương pháp lưu trữ / tìm tệp trên đĩa cứng (thường là trong phân vùng).

Người ta có thể nghĩ về một phân vùng như một thùng chứa trong đó một hệ thống tập tin cư trú, mặc dù trong một số trường hợp, một hệ thống tập tin có thể trải rộng hơn một phân vùng nếu sử dụng các liên kết tượng trưng, chúng ta sẽ thảo luận nhiều sau.

Một so sánh giữa các hệ thống tập tin trong Windows và Linux được đưa ra trong bảng đi kèm:

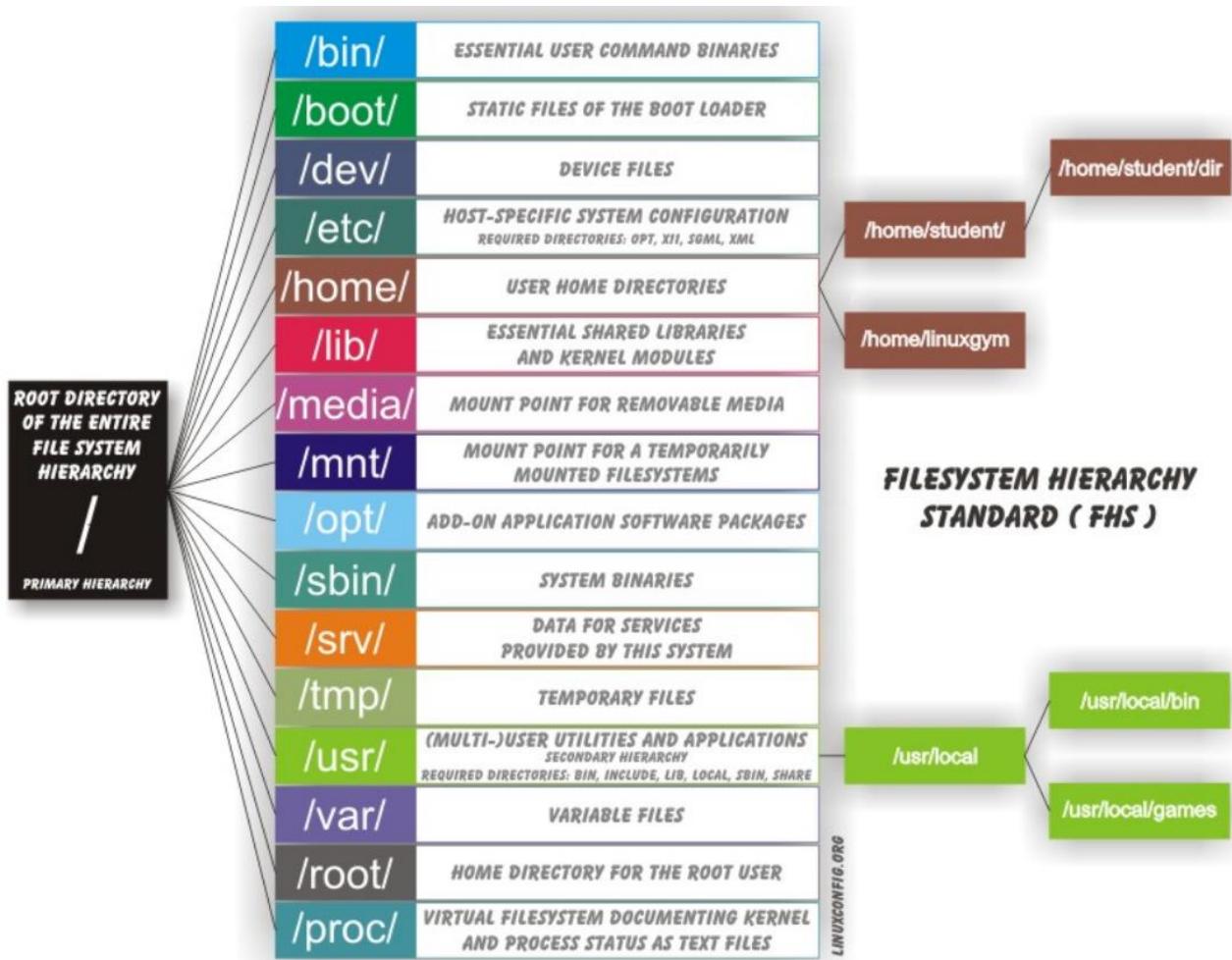
### The Filesystem Hierarchy Standard

Các hệ thống Linux lưu trữ các tệp quan trọng của chúng theo bố cục tiêu chuẩn được gọi là Tiêu chuẩn phân cấp hệ thống tệp (FHS), được duy trì từ lâu bởi Quỹ Linux. Để biết thêm thông tin, hãy xem tài liệu sau: "Tiêu chuẩn phân cấp hệ thống tập tin" được tạo bởi LSB Workgroup. Có một tiêu chuẩn được thiết kế để đảm bảo rằng người dùng, quản trị viên và nhà phát triển có thể di chuyển giữa các bản phân phối mà không phải tìm hiểu lại cách tổ chức hệ thống.

Linux sử dụng ký tự '/' (backslash) để phân tách các đường dẫn (không giống như Windows, sử dụng '\,) và không có ký tự ổ đĩa. Nhiều ổ đĩa và / hoặc phân vùng được gắn kết dưới dạng thư mục trong hệ thống tập tin duy nhất. Các phương tiện lưu động như ổ đĩa USB và CD và DVD sẽ hiển thị dưới dạng được gắn tại / run / media / yourusername / Disclabel cho các hệ thống Linux gần đây hoặc dưới / media cho các bản phân phối cũ hơn. Ví dụ: nếu tên người dùng của bạn là sinh viên, ổ đĩa USB có nhãn FedORA có thể sẽ được tìm thấy tại / run / media / student / FedORA và một tệp README.txt trên đĩa đó sẽ có tại / run / media / student / FedORA / README.txt.

Tất cả các tên hệ thống tập tin Linux đều phân biệt chữ hoa chữ thường, vì vậy / boot, / Boot và / BOOT đại diện cho ba thư mục (hoặc thư mục) khác nhau. Nhiều bản phân phối phân biệt giữa các tiện ích cốt lõi cần thiết cho hoạt động hệ thống phù hợp và các chương trình khác, và đặt cái sau trong các thư mục dưới / usr (nghĩ người dùng). Để hiểu về cách các chương trình khác được tổ chức, hãy tìm thư mục / usr trong sơ đồ từ trang trước và so sánh các thư mục con với các chương trình tồn tại ngay dưới thư mục gốc của hệ thống (/).

## 2.12 Câu hỏi ôn tập Chương 2



## The Filesystem Hierarchy Standard

Hệ thống tập tin của Linux Ubuntu được tổ chức theo một hệ thống phân bậc tương tự cấu trúc của một cây, bao gồm 1 thân thẳng đứng và các cành lớn chia ra. Bậc cao nhất của hệ thống tập tin là thư mục gốc, được ký hiệu bằng vạch chéo “/”. Đối với các hệ điều hành Linux tất cả các thiết bị kết nối vào máy tính đều được nhận ra như các tập tin, kể cả những linh kiện như ổ đĩa cứng, các phân vùng đĩa cứng và các ổ USB. Có nghĩa là tất cả các tập tin và thư mục đều nằm dưới thư mục gốc. Ở đây các phân vùng đĩa cứng được xem như “gắn” vào các thư mục.

Thí dụ, `/home/users/guest/ubuntu.txt` chỉ toàn bộ đường dẫn đến tập tin “ubuntu.txt” có trong thư mục “guest” là thư mục phụ nằm trong thư mục “users”, trong thư mục “home” dưới thư mục gốc “/”.

Nằm dưới thư mục gốc “/” có các thư mục quan trọng của hệ thống tập tin được công nhận ở tất cả các bản phân phối Linux khác nhau. Sau đây là danh sách các thư mục nằm dưới thư mục gốc “/”:

/bin – chứa các ứng dụng quan trọng

/boot – các tập tin cấu hình cho quá trình khởi động hệ thống

/dev – chứa các tập tin là tương ứng cho các thiết bị của hệ thống,

/etc – chứa các tập tin cấu hình của hệ thống

/home – chứa các thư mục của người dùng

/lib – lưu các thư viện chia sẻ của hệ thống

/media – thư mục này được dùng để tạo ra các tập tin gắn (mount) tạm thời được hệ thống tạo ra khi một thiết bị lưu trữ được cắm vào như USB, CD/DVD, etc.

/mnt – thư mục này được dùng để gắn các hệ thống tập tin tạm thời

/opt – thư mục dùng để chứa các phần mềm ứng dụng đã được cài đặt thêm

/proc – lưu các thông tin về tình trạng của hệ thống, về các tiến trình đang hoạt động

/root – đây là thư mục nhà của người quản trị hệ thống (*root*)

/sbin – thư mục này lưu lại các tập tin thực thi của hệ thống

/sys – thư mục này lưu các tập tin của hệ thống

/tmp – thư mục này lưu lại các tập tin được tạo ra tạm thời

/usr – lưu những tập tin của các ứng dụng chính đã được cài đặt cho mọi người dùng

/var – lưu lại các tập tin ghi các số liệu biến đổi như các tập tin dữ liệu và nhật ký,...

### Nhận diện và phân vùng đĩa cứng

Như đã nói ở trên các thiết bị được nhận diện thư một tập tin trong hệ thống Linux. Đối với các đĩa cứng chuẩn **IDE** sẽ được nhận diện là **/dev/hda** (và **/dev/hdb** nếu máy có hai đĩa cứng). Nếu máy dùng đĩa cứng theo chuẩn **SCSI** thì sẽ được nhận diện là **/dev/sda**.

Việc phân vùng đĩa cứng trong Linux có sự khác biệt so với phân vùng đĩa cứng khi các bạn cài đặt hệ điều hành Windows. Đối với hệ điều hành Windows bạn sẽ cài toàn bộ hệ điều hành lên một phân vùng chính (*primary*) và các thư mục được tạo trong các đĩa (C:, D:, E:,...)

Primary Partition 1	Primary Partition 2	Primary Partition 3	Extended Partition		
			Logical 1	Logical 2	Logical 3

[C:]	[D:]	[E:]	[F:]	[G:]	[H:]
------	------	------	------	------	------

Trong hệ thống linux thì các thư mục được gắn trực tiếp vào các phân vùng

Primary Partition 1	Primary Partition 2	Primary Partition 3	Extended Partition		
/dev/hda1	/dev/hda2	/dev/hda3	/dev/hda5	/dev/hda6	/dev/hda7
/	/swap	/usr	/home	/var	/tmp

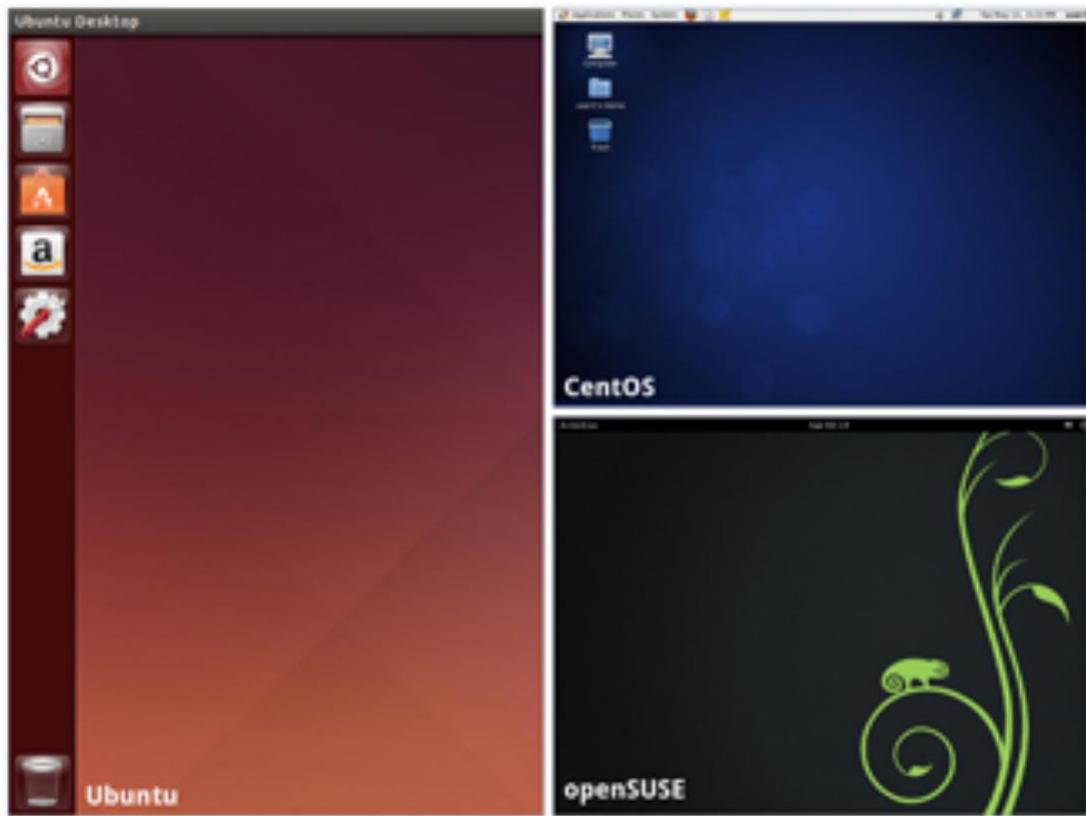
## Chương 3. Làm việc trên môi trường đồ họa Desktop Linux

### 3.1 Giới thiệu

Môi trường làm việc đồ họa (được gọi là Desktop) mặc định của Ubuntu là [GNOME](#), một bộ phần mềm desktop dẫn đầu ở các hệ điều hành Linux. Một môi trường làm việc đồ họa nổi tiếng khác ở các hệ điều hành Linux mang tên [KDE](#). Trong tài liệu này chỉ giới thiệu môi trường làm việc đồ họa GNOME.

Bạn có thể sử dụng Giao diện dòng lệnh (CLI) hoặc Giao diện người dùng đồ họa (GUI) khi sử dụng Linux. Để làm việc tại CLI, bạn phải nhớ các chương trình và lệnh nào được sử dụng để thực hiện các tác vụ và cách nhanh chóng và chính xác để có thêm thông tin về việc sử dụng và các tùy chọn của chúng. Mặt khác, sử dụng GUI thường nhanh chóng và dễ dàng. Nó cho phép bạn tương tác với hệ thống của bạn thông qua các biểu tượng và màn hình đồ họa. Đối với các tác vụ lặp đi lặp lại, CLI thường hiệu quả hơn, trong khi GUI dễ điều hướng hơn nếu bạn không nhớ tất cả các chi tiết hoặc hiếm khi làm điều gì đó.

Chúng tôi sẽ tìm hiểu cách quản lý phiên bằng GUI cho ba họ phân phối Linux mà chúng tôi đề cập nhiều nhất trong khóa học này: Red Hat (CentOS, Fedora), SUSE (openSUSE) và Debian (Ubuntu, Mint). Vì chúng tôi đang sử dụng biến thể openSUSE dựa trên Gnome thay vì dựa trên KDE, tất cả đều thực sự khá giống nhau. Nếu bạn đang sử dụng KDE (hoặc các máy tính để bàn Linux khác như XFCE), trải nghiệm của bạn sẽ thay đổi đôi chút so với những gì được hiển thị, nhưng không theo bất kỳ cách khó khăn nào, vì giao diện người dùng đã hội tụ một số hành vi nổi tiếng nhất định trên các hệ điều hành hiện đại. Trong các phần tiếp theo của khóa học này, chúng tôi sẽ tập trung rất chi tiết vào giao diện dòng lệnh, khá giống nhau trên tất cả các bản phân phối.



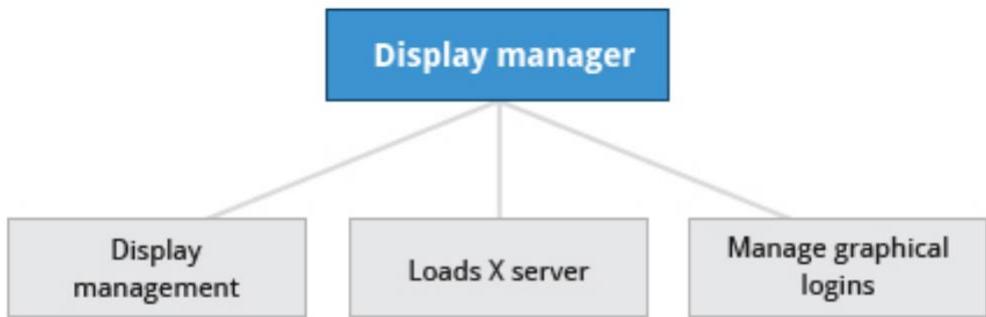
### Ubuntu, CentOS, and openSUSE Desktops

## 3.2 X Window System

Nói chung, trong hệ thống máy tính để bàn Linux, Hệ thống X Window được tải là một trong những bước cuối cùng trong quy trình khởi động.

Một dịch vụ được gọi là trình quản lý hiển thị theo dõi các màn hình được cung cấp và tải máy chủ X (được gọi là vì nó cung cấp dịch vụ đồ họa cho các ứng dụng, đôi khi được gọi là máy khách X). Trình quản lý hiển thị cũng xử lý các thông tin đăng nhập đồ họa và bắt đầu môi trường máy tính để bàn thích hợp sau khi người dùng đăng nhập.

Hệ thống X Window là một hệ thống khá cũ; nó có từ giữa những năm 1980 và, do đó, có một số thiếu sót nhất định trên các hệ thống hiện đại (ví dụ, với bảo mật), vì nó đã được kéo dài khá xa so với mục đích ban đầu của nó. Một hệ thống mới hơn, được gọi là Wayland, đang dần thay thế nó và được sử dụng làm hệ thống hiển thị mặc định trong Fedora 25.

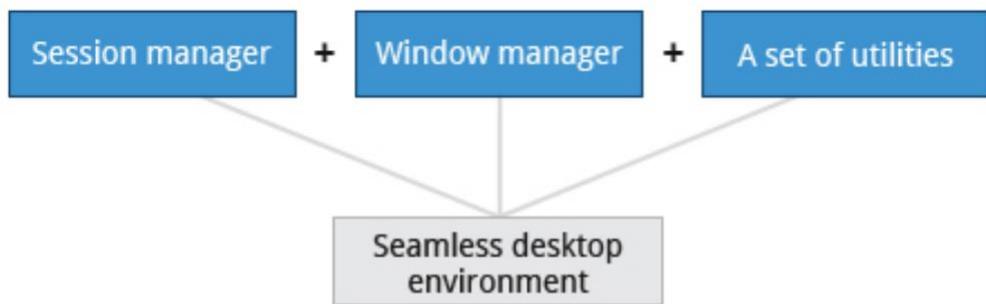


## Display Manager

Môi trường máy tính để bàn bao gồm trình quản lý phiên, khởi động và duy trì các thành phần của phiên đồ họa và trình quản lý cửa sổ, điều khiển vị trí và chuyển động của các cửa sổ, thanh tiêu đề cửa sổ và điều khiển.

Mặc dù chúng có thể được trộn lẫn, nhưng nhìn chung, một tập hợp các tiện ích, trình quản lý phiên và trình quản lý cửa sổ được sử dụng cùng nhau như một đơn vị và cùng nhau cung cấp một môi trường máy tính để bàn liền mạch.

Nếu trình quản lý hiển thị không được khởi động theo mặc định trong runlevel mặc định, bạn có thể khởi động X theo một cách khác, sau khi đăng nhập vào bảng điều khiển chế độ văn bản, bằng cách chạy startx từ dòng lệnh. Hoặc, bạn có thể khởi động trình quản lý hiển thị (gdm, lightdm, kdm, xdm, v.v.) bằng tay từ dòng lệnh. Điều này khác với việc chạy startx vì các trình quản lý hiển thị sẽ chiếm màn hình đăng nhập. Chúng tôi thảo luận về chúng tiếp theo.



## Desktop Environment

### 3.3 Khởi động giao diện đồ họa (GUI)

Khi bạn cài đặt môi trường máy tính để bàn, trình quản lý hiển thị X sẽ bắt đầu ở cuối quá trình khởi động. Nó chịu trách nhiệm khởi động hệ thống đồ họa, đăng nhập người dùng và bắt đầu môi trường máy tính để bàn của người dùng. Bạn thường có thể chọn từ một lựa chọn môi trường máy tính để bàn khi đăng nhập vào hệ thống.

Trình quản lý hiển thị mặc định cho Gnome được gọi là gdm. Các trình quản lý hiển thị phổ biến khác bao gồm lightdm (được sử dụng trên Ubuntu trước phiên bản 18.04 LTS) và kdm (được liên kết với KDE).

#### GNOME Desktop Environment

Gnome là một môi trường máy tính để bàn phổ biến với giao diện người dùng đồ họa dễ sử dụng. Nó được đóng gói như môi trường máy tính để bàn mặc định cho hầu hết các bản phân phối Linux, bao gồm Red Hat Enterprise Linux, Fedora, CentOS, SUSE Linux Enterprise, Ubuntu và Debian. Gnome có điều hướng dựa trên menu và đôi khi là một quá trình chuyển đổi dễ dàng để thực hiện cho người dùng Windows. Tuy nhiên, như bạn sẽ thấy, giao diện có thể khá khác nhau giữa các bản phân phối, ngay cả khi tất cả chúng đều sử dụng Gnome.



Một môi trường máy tính để bàn phổ biến khác rất quan trọng trong lịch sử Linux và cũng được sử dụng rộng rãi là KDE, thường được sử dụng cùng với SUSE và openSUSE. Các lựa chọn thay thế khác cho môi trường máy tính để bàn bao gồm Unity (từ Ubuntu cũ hơn, nhưng vẫn dựa trên Gnome), XFCE và LXDE. Như đã đề cập trước đây, hầu hết các môi trường máy tính để bàn đều có cấu trúc tương tự như Gnome và chúng tôi sẽ hạn chế chú ý vào nó để giữ cho mọi thứ ít phức tạp hơn.

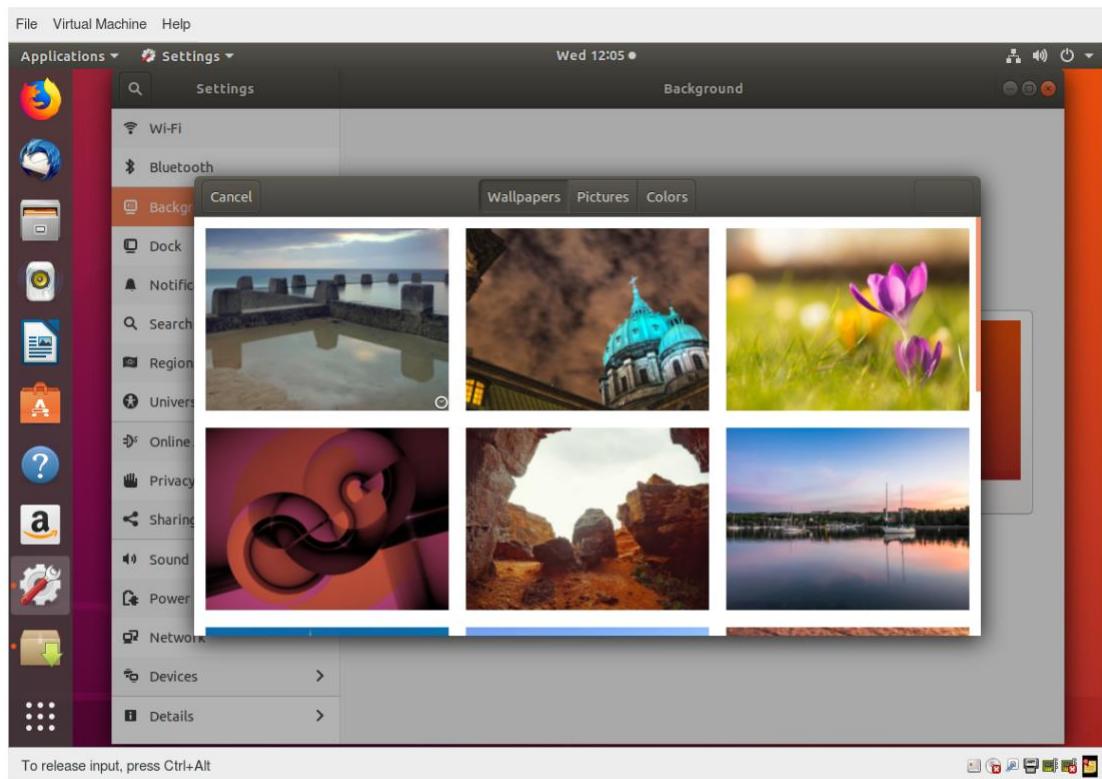
#### Graphical Desktop Background

Mỗi bản phân phối Linux đi kèm với bộ nền máy tính để bàn riêng. Bạn có thể thay đổi mặc định bằng cách chọn hình nền mới hoặc chọn hình ảnh tùy chỉnh để đặt làm hình nền máy tính. Nếu bạn không muốn sử dụng hình ảnh làm nền, bạn có thể chọn màu sẽ được hiển thị trên màn hình nền.

Ngoài ra, bạn cũng có thể thay đổi chủ đề máy tính để bàn, thay đổi giao diện của hệ thống Linux. Chủ đề cũng xác định sự xuất hiện của các cửa sổ ứng dụng.

Chúng ta sẽ học cách thay đổi hình nền và chủ đề của máy tính để bàn.

Để thay đổi nền, bạn có thể nhấp chuột phải vào bất kỳ đâu trên màn hình và chọn Thay đổi nền.



## How to Customize the Desktop Background

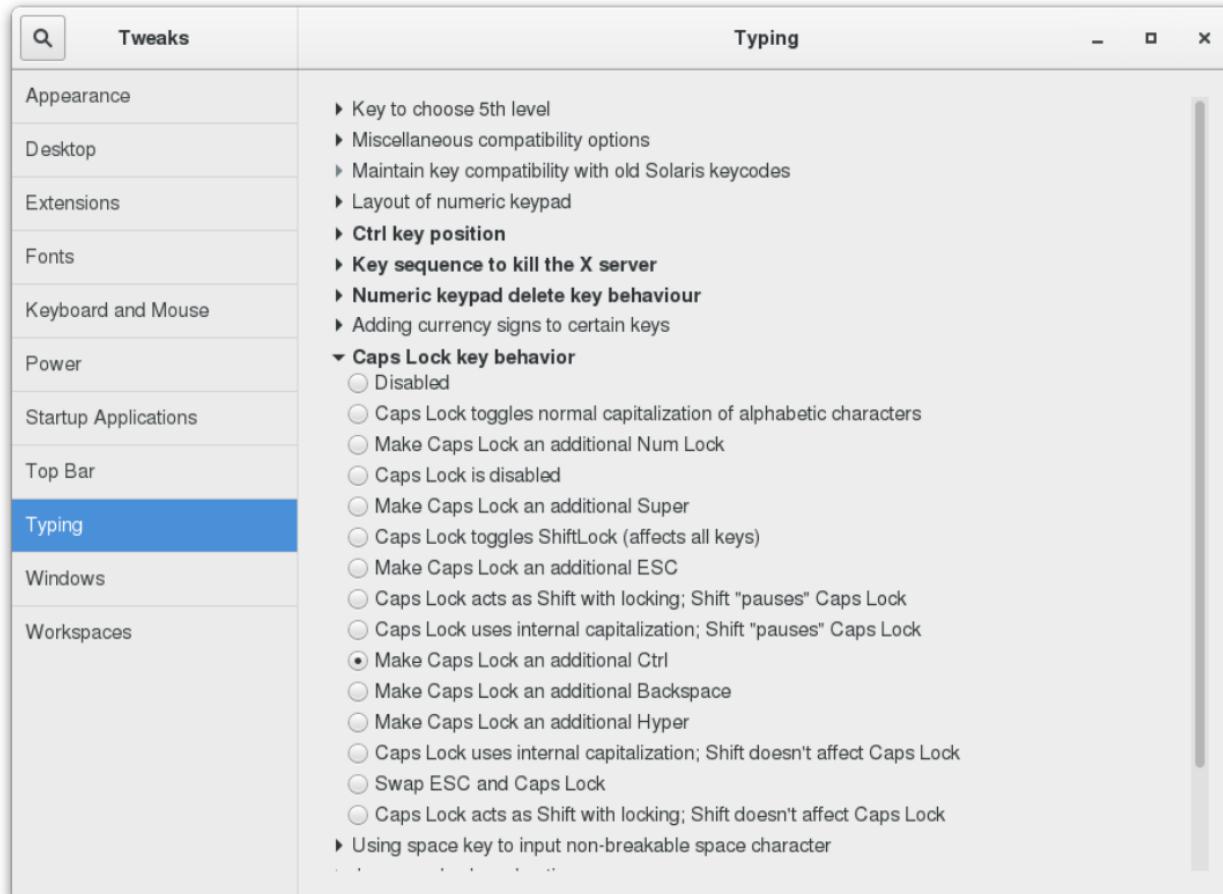
### Công cụ Gnome-tweak-tool

Hầu hết các cài đặt, cả cá nhân và toàn hệ thống, đều được tìm thấy bằng cách nhấp vào góc trên bên phải, trên một bánh răng hoặc biểu tượng rõ ràng khác, tùy thuộc vào bản phân phối Linux của bạn.

Tuy nhiên, có nhiều cài đặt mà nhiều người dùng muốn sửa đổi không thể truy cập được; không may là tiện ích cài đặt mặc định khá hạn chế trong các bản phân phối dựa trên Gnome hiện đại. Mong muốn đơn giản thực sự đã gây khó khăn cho việc thích ứng hệ thống của bạn với thị hiếu và nhu cầu của bạn.

May mắn thay, có một tiện ích tiêu chuẩn, gnome-chỉnh-tool, cho thấy nhiều tùy chọn cài đặt hơn. Nó cũng cho phép bạn dễ dàng cài đặt tiện ích mở rộng của các bên ngoài. Không phải tất cả các bản phân phối Linux đều cài đặt công cụ này theo mặc định, nhưng nó luôn có sẵn. Một số bản phân phối gần đây đã đổi tên công cụ này thành gnome-chỉnh. Bạn có thể phải chạy nó bằng cách nhấn Alt-F2 và sau đó nhập tên. Bạn có thể muốn thêm nó vào danh sách yêu thích của bạn như chúng ta sẽ thảo luận.

Trong ảnh chụp màn hình bên dưới, ánh xạ bàn phím đang được điều chỉnh để phím CapsLock vô dụng có thể được sử dụng làm phím Ctrl bổ sung; điều này giúp người dùng sử dụng Ctrl rất nhiều (chẳng hạn như emacs aficionados) khỏi bị tổn thương về thể chất do chủng Pinkie.



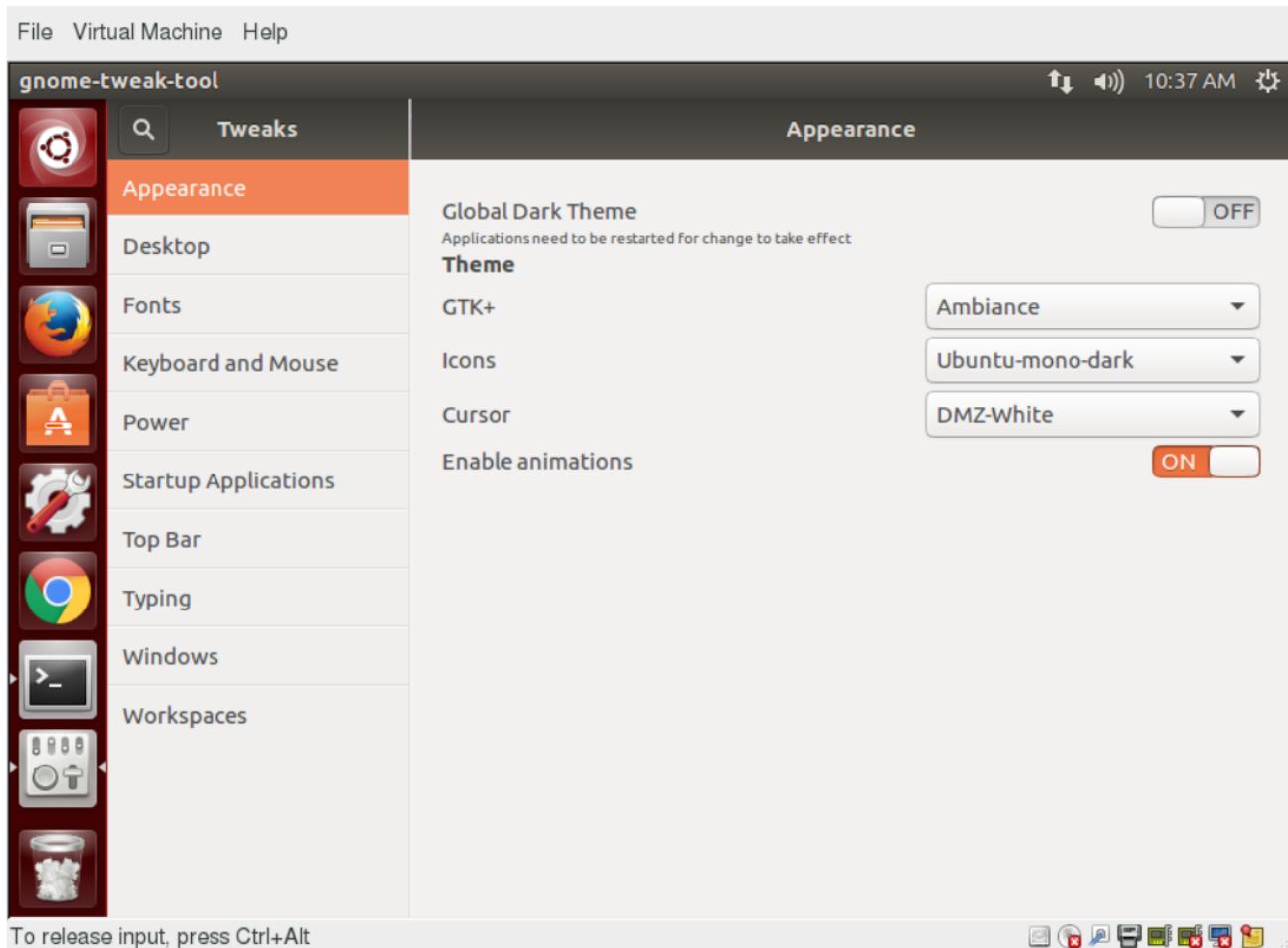
## gnome-tweak-tool

### Thay đổi các chủ đề (Theme)

Giao diện trực quan của các ứng dụng (các nút, thanh cuộn, vật dụng và các thành phần đồ họa khác) được điều khiển bởi một chủ đề. Gnome đi kèm với một tập hợp các chủ đề khác nhau có thể thay đổi giao diện của ứng dụng của bạn.

Phương pháp chính xác để thay đổi chủ đề của bạn có thể phụ thuộc vào phân phối của bạn. Tuy nhiên, đối với tất cả các bản phân phối dựa trên Gnome, bạn chỉ cần chạy gnome-tweak-tool, như trong ảnh chụp màn hình từ Ubuntu.

Có các tùy chọn khác để có được các chủ đề bổ sung ngoài lựa chọn mặc định. Bạn có thể tải xuống và cài đặt các chủ đề từ trang web Wiki của Gnome.



## Changing the Theme

Mặc dù phần này dài, chúng tôi sẽ không thực hiện các bài tập trong phòng thí nghiệm từng bước rất phức tạp, vì sự đa dạng của các bản phân phối và phiên bản Linux, và vì chúng đều tùy chỉnh máy tính để bàn của chúng, ngay cả khi cơ sở mã cơ bản giống nhau. Có gắng đưa ra hướng dẫn chính xác là một bài tập vô ích; không chỉ có nhiều biến thể, chúng còn có khả năng thay đổi mỗi khi phiên bản phân phối Linux mới được phát hành.

Đối với hầu hết các phần, đây không phải là một vấn đề. Giao diện đồ họa được thiết kế để dễ dàng điều hướng và tìm ra, và chúng thực sự không khác nhau nhiều, không chỉ từ phân phối này sang phân phối khác, mà ngay cả giữa các hệ điều hành. Vì vậy, cách duy nhất bạn có thể trở nên lão luyện hơn khi làm việc hiệu quả trên máy tính để bàn của mình là chỉ cần khám phá, chơi và sửa đổi. Những điểm tương tự sẽ được áp dụng cho chương tiếp theo, về cấu hình hệ thống đồ họa.

Linux có thể tùy biến đến mức rất ít người sử dụng nó vẫn giữ được giao diện mặc định của máy tính để bàn. Nay giờ bạn cũng có thể bắt đầu làm cho máy tính để bàn của bạn phản ánh sở thích và tính cách của bạn.

Bắt đầu bằng cách thay đổi nền màn hình thành một cái gì đó phù hợp hơn với sở thích của bạn; có lẽ là một trong những hình nền được cung cấp, màu đơn sắc của sự lựa chọn của bạn hoặc hình ảnh cá nhân mà bạn có thể chuyển sang môi trường Linux của mình.

Tiếp theo, chọn một chủ đề từ các chủ đề có sẵn cho bản phân phối của bạn, một lần nữa, phù hợp với sở thích và tính cách của bạn. Hãy vui vẻ và khám phá với bài tập này.

## **Khóa màn hình**

Thường là một ý tưởng tốt để khóa màn hình của bạn để ngăn người khác truy cập phiên của bạn trong khi bạn rời khỏi máy tính của bạn. Lưu ý: Điều này không định chỉ máy tính; tất cả các ứng dụng và quy trình của bạn tiếp tục chạy trong khi màn hình bị khóa.

Có hai cách để khóa màn hình của bạn:

### *Sử dụng giao diện đồ họa*

Bấm vào góc trên bên phải của màn hình nền, rồi bấm vào biểu tượng khóa.

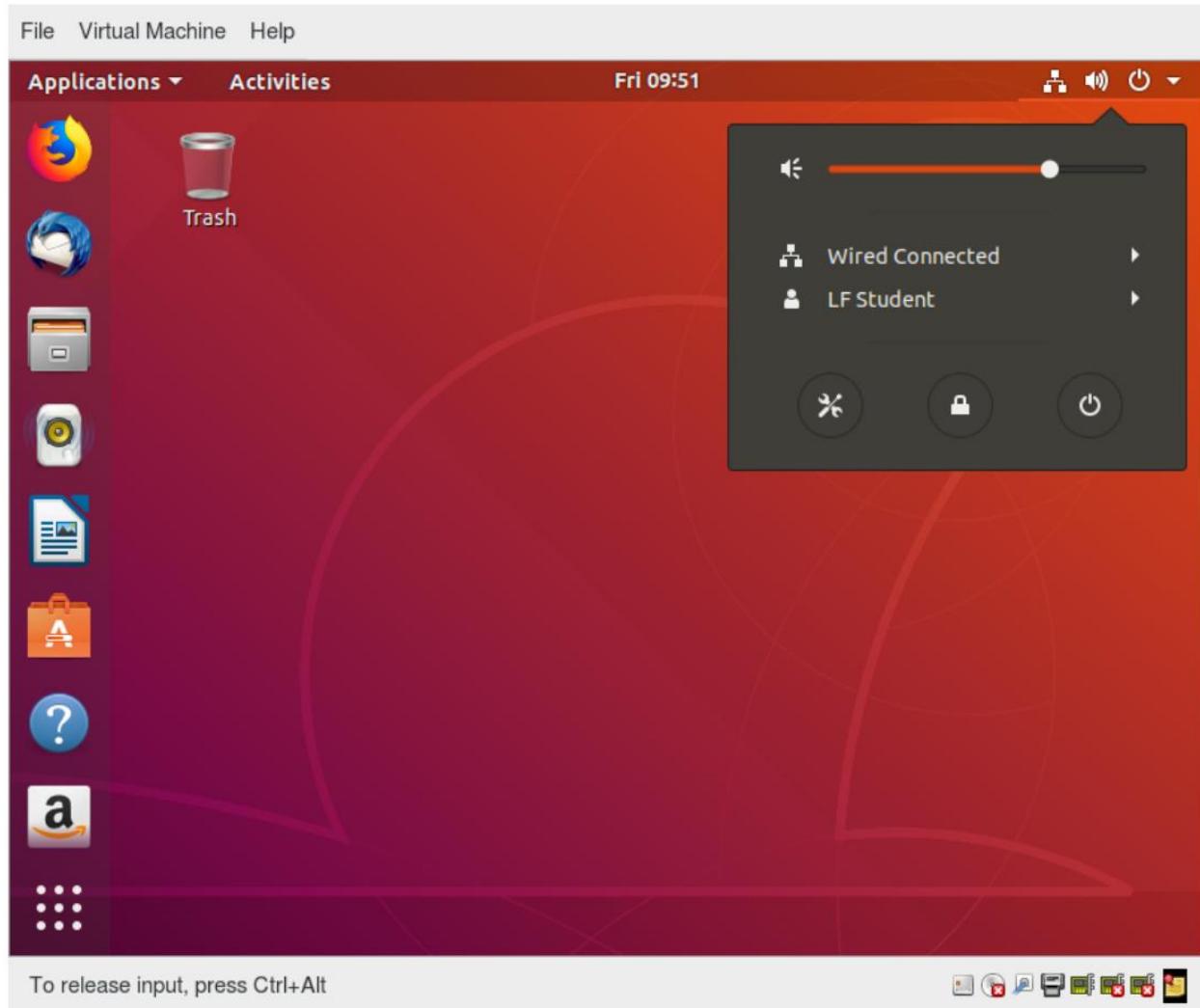
### *Sử dụng phím tắt SUPER-L*

(Phím SUPER còn được gọi là phím Windows).

Phím tắt để khóa màn hình trong ba bản phân phối có thể được sửa đổi bằng cách thay đổi cài đặt bàn phím, đơn thuốc chính xác thay đổi theo phân phối, nhưng không khó để xác định.

Để nhập lại phiên máy tính để bàn, bạn chỉ cần cung cấp lại mật khẩu của mình.

Ảnh chụp màn hình bên dưới cho thấy cách khóa màn hình cho Ubuntu. Các chi tiết khác nhau rất ít trong phân phối hiện đại.



## Locking the Screen for Ubuntu

Để khóa và mở khóa màn hình của bạn trong openSUSE, hãy thực hiện các bước sau:

Nhấp vào biểu tượng nguồn ở góc trên bên phải của màn hình.

Nhấp vào biểu tượng khóa. Màn hình bị khóa ngay lập tức.

Nhấn Enter. Màn hình đăng nhập được hiển thị.

Để mở khóa màn hình, nhập mật khẩu.

Nhấp vào Mở khóa và màn hình máy tính để bàn được hiển thị.

Khóa NTo và mở khóa màn hình của bạn trong openSUSE, thực hiện các bước sau:

Nhấp vào biểu tượng nguồn ở góc trên bên phải của màn hình.

Nhấp vào biểu tượng khóa. Màn hình bị khóa ngay lập tức.

Nhấn Enter. Màn hình đăng nhập được hiển thị.

Để mở khóa màn hình, nhập mật khẩu.

Nhấp vào Mở khóa và màn hình máy tính để bàn được hiển thị.

Lưu ý: Khi bạn khóa màn hình, Gnome sẽ làm trống màn hình hoặc chạy trình bảo vệ màn hình, tùy thuộc vào cài đặt của bạn. Lưu ý: Khi bạn khóa màn hình, Gnome sẽ làm trống màn hình hoặc chạy trình bảo vệ màn hình, tùy thuộc vào cài đặt của bạn.

## Thay đổi người dùng

Linux là một hệ điều hành nhiều người dùng thực sự, cho phép nhiều người dùng đăng nhập đồng thời. Nếu có nhiều người sử dụng hệ thống, tốt nhất là mỗi người nên có tài khoản và mật khẩu riêng. Điều này cho phép cài đặt cá nhân, thư mục chính và các tệp khác. Người dùng có thể thay phiên nhau sử dụng máy, trong khi vẫn duy trì các phiên của mọi người hoặc thậm chí được đăng nhập đồng thời qua mạng.

### 3.4 Tắt máy và khởi động lại Linux với giao diện đồ họa

Bên cạnh việc khởi động và dừng máy tính hàng ngày thông thường, việc khởi động lại hệ thống có thể được yêu cầu như là một phần của một số cập nhật hệ thống chính, thường chỉ là những việc liên quan đến việc cài đặt kernel Linux mới.

Bắt đầu quá trình tắt máy tính từ máy tính để bàn đồ họa khá tầm thường trên tất cả các bản phân phối Linux hiện tại, với rất ít biến thể. Chúng ta sẽ thảo luận sau để làm điều này từ dòng lệnh, sử dụng lệnh tắt máy.

Trong mọi trường hợp, bạn nhấp vào một trong hai cài đặt (bánh răng) hoặc biểu tượng nguồn và làm theo lời nhắc. Chúng tôi sẽ chỉ hiển thị chi tiết cho bản phân phối Ubuntu Linux.

Để tắt máy tính trong Ubuntu, hãy thực hiện các bước sau:

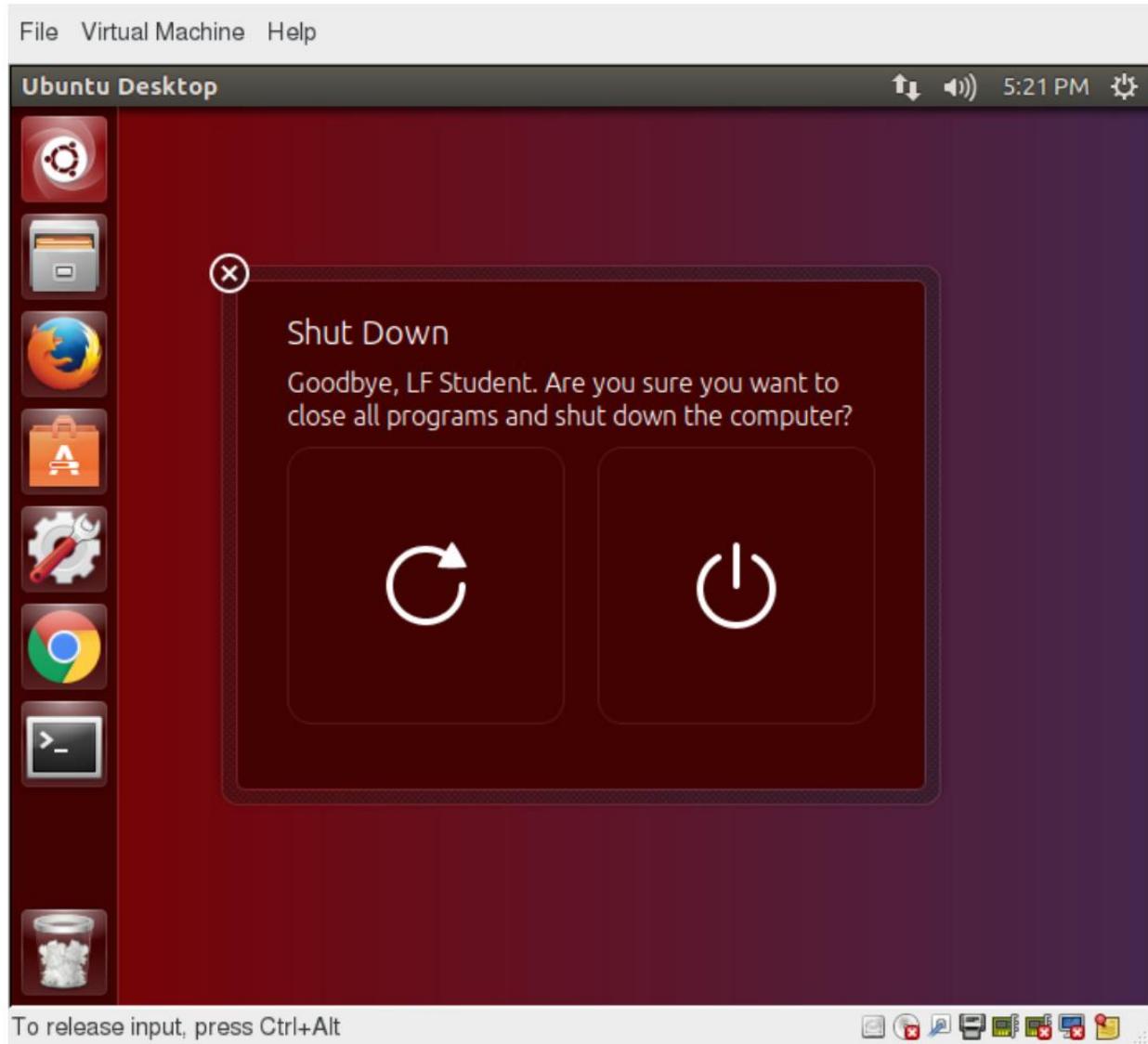
Trên màn hình máy tính để bàn Ubuntu, nhấp vào biểu tượng Nguồn (cho các phiên bản trước) hoặc biểu tượng Bánh răng ở góc trên bên phải màn hình.

Nhấp vào Tắt máy. Các biểu tượng Khởi động lại và Tắt máy được hiển thị.

Nhấp vào biểu tượng Tắt máy ở bên phải để tắt hệ thống và nhấp vào biểu tượng Khởi động lại ở bên trái để khởi động lại hệ thống.

Tắt máy, khởi động lại và các hoạt động đăng xuất sẽ yêu cầu xác nhận trước khi tiếp tục. Điều này là do nhiều ứng dụng sẽ không lưu dữ liệu của họ đúng cách khi bị chấm dứt theo cách này.

Luôn lưu tài liệu và dữ liệu của bạn trước khi khởi động lại, tắt hoặc đăng xuất. Ảnh chụp màn hình Ubuntu được cung cấp ở đây tương tự như tắt cả các cửa sổ xác nhận phân phối.



## Shutting Down and Restarting in Ubuntu

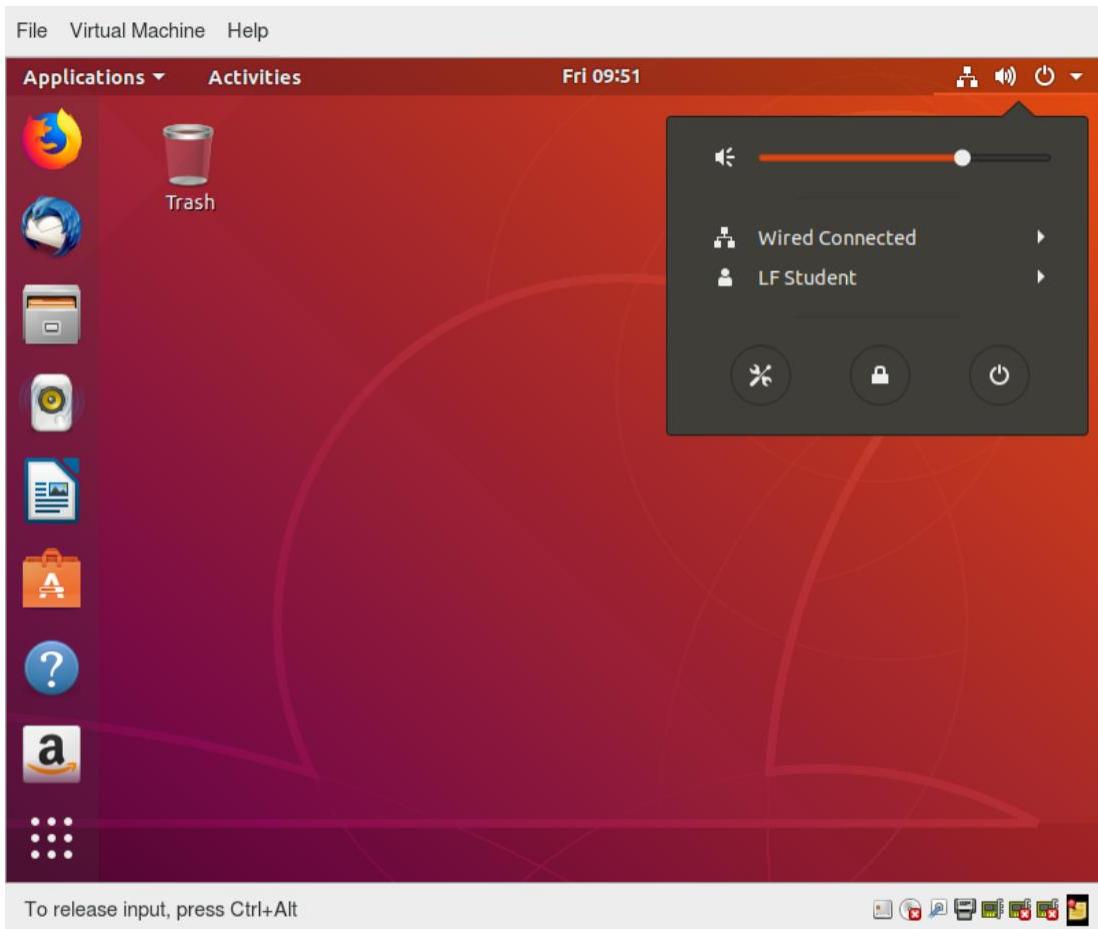
### Tạm nghỉ (Suspending)

Hầu hết các máy tính hiện đại đều hỗ trợ Chế độ treo hoặc Chế độ ngủ khi bạn ngừng sử dụng máy tính trong một thời gian ngắn. Chế độ Tạm dừng lưu trạng thái hệ thống hiện tại và cho phép bạn tiếp tục phiên của mình nhanh hơn trong khi vẫn duy trì, nhưng sử dụng rất ít năng lượng. Nó hoạt động bằng cách giữ cho các ứng dụng, hệ thống máy tính để bàn của bạn, v.v., trong RAM hệ thống, nhưng tắt tất cả các phần cứng khác. Chế độ Tạm dừng bỏ qua thời gian khởi động toàn hệ thống và tiếp tục sử dụng công suất tối thiểu.

Để tạm dừng hệ thống trong các bản phân phối Ubuntu cũ hơn, quy trình này giống như tắt máy hoặc khóa màn hình. Sau khi bạn nhấp vào biểu tượng ở phía trên bên phải, bạn chỉ cần chọn Tạm dừng, như thể hiện trong ảnh chụp nhanh bên dưới

Tuy nhiên, nó khá đơn giản và phổ biến trong các bản phân phối dựa trên Gnome gần đây. Nếu bạn nhấp vào biểu tượng Nguồn và giữ trong một thời gian ngắn và phát hành, bạn sẽ nhận được biểu tượng dấu chấm kép, sau đó bạn nhấp vào để tạm dừng hệ thống.

Lưu ý: Để đánh thức hệ thống của bạn và tiếp tục phiên của bạn, hãy di chuyển chuột hoặc nhấn bất kỳ nút nào trên bàn phím. Hệ thống sẽ thức dậy với màn hình bị khóa, giống như bạn đã tự khóa nó; gõ mật khẩu của bạn để tiếp tục.



## Suspending Ubuntu System

### 3.5 Một số thao tác cơ bản với giao diện đồ họa trên Linux

Ngay cả người dùng có kinh nghiệm cũng có thể quên lệnh chính xác khởi chạy một ứng dụng hoặc chính xác những tùy chọn và đối số mà nó yêu cầu. May mắn thay, Linux cho phép bạn nhanh chóng mở các ứng dụng bằng giao diện đồ họa.

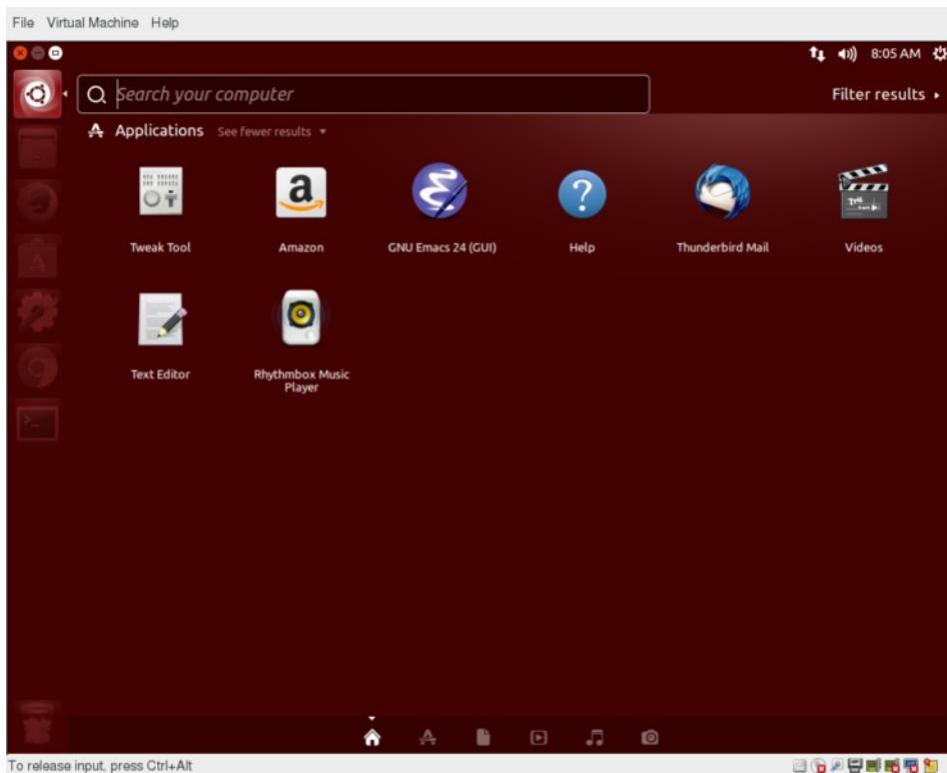
Các ứng dụng được tìm thấy ở những nơi khác nhau trong Linux (và trong Gnome):

Tùy menu **Ứng dụng** ở góc trên bên trái

Tùy menu **Hoạt động** ở góc trên bên trái

Trong một số phiên bản Ubuntu, từ nút Dash ở góc trên bên trái

Đối với KDE và một số môi trường khác, các ứng dụng có thể được mở từ nút ở góc dưới bên trái.



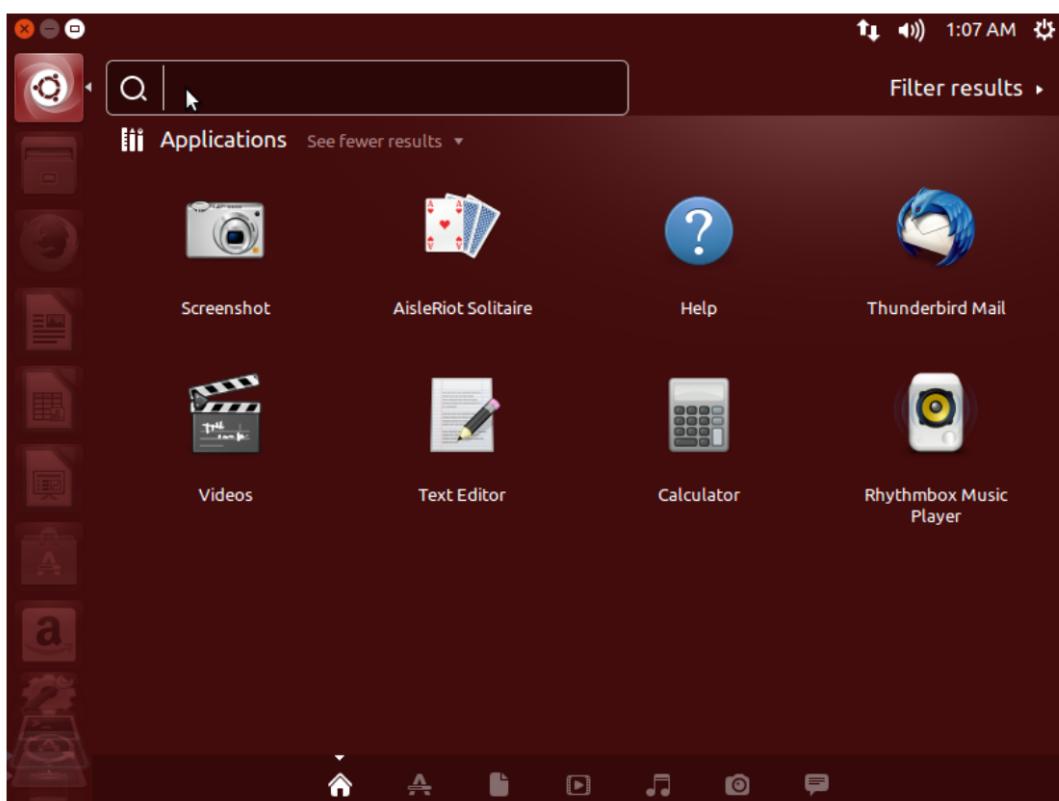
#### Opening Applications on Ubuntu

## Tìm kiếm các ứng dụng

Không giống như các hệ điều hành khác, bản cài đặt ban đầu của Linux thường đi kèm với một loạt các ứng dụng và kho lưu trữ phần mềm có chứa hàng ngàn chương trình cho phép bạn thực hiện nhiều tác vụ khác nhau với máy tính của mình. Đối với hầu hết các tác vụ chính, một ứng dụng mặc định thường được cài đặt. Tuy nhiên, bạn luôn có thể cài đặt nhiều ứng dụng hơn và thử các tùy chọn khác nhau.

Ví dụ: Firefox phổ biến là trình duyệt mặc định trong nhiều bản phân phối Linux, trong khi Epiphany, Konqueror và Chromium (cơ sở nguồn mở cho Google Chrome) thường có sẵn để cài đặt từ kho phần mềm. Các trình duyệt web độc quyền, như Opera và Chrome, cũng có sẵn.

Việc định vị các ứng dụng từ menu Gnome và KDE rất dễ dàng, vì chúng được sắp xếp gọn gàng trong các menu con chức năng.

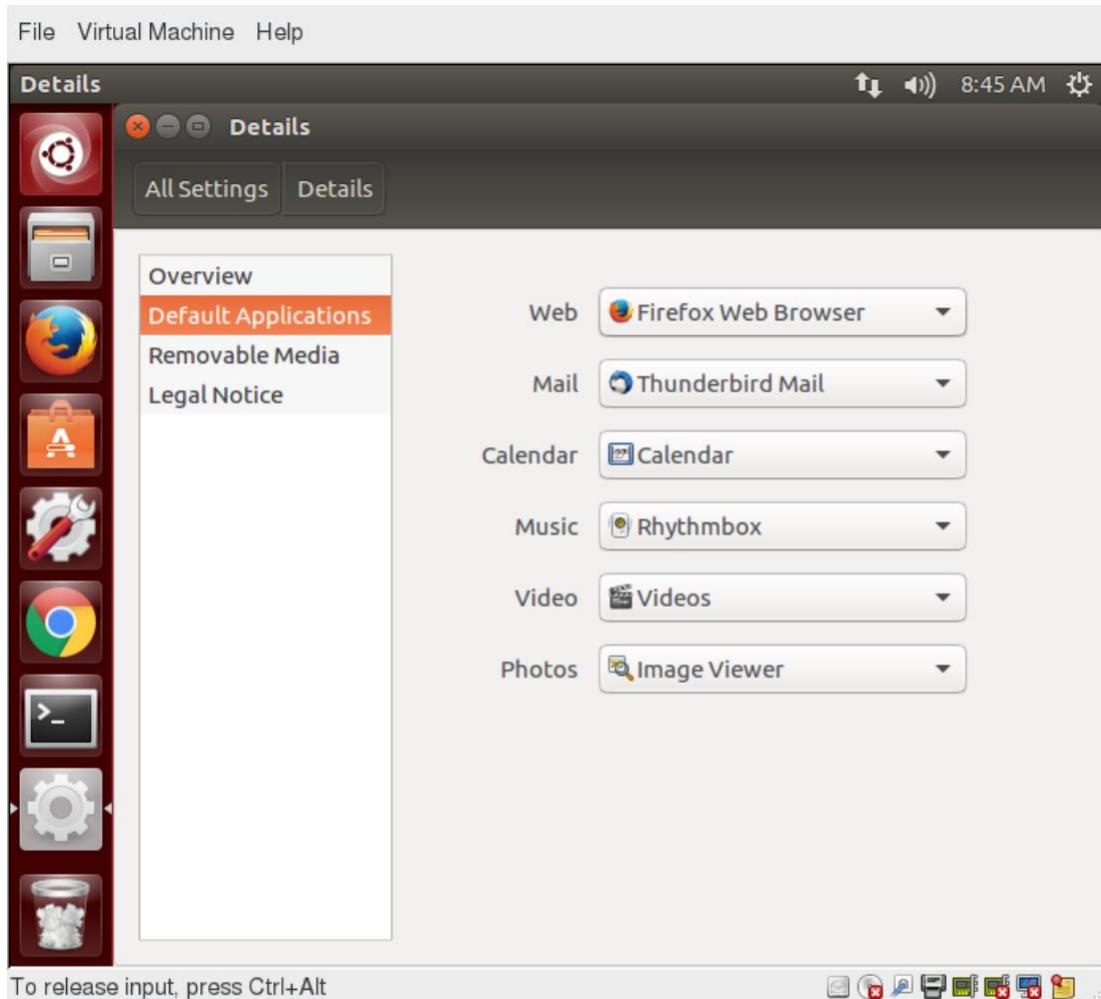


## Locating Applications

### Ứng dụng ngầm định

Nhiều ứng dụng có sẵn để thực hiện các nhiệm vụ khác nhau và để mở một tệp thuộc loại đã cho. Ví dụ: bạn có thể nhấp vào địa chỉ web trong khi đọc email và khởi chạy trình duyệt như Firefox hoặc Chrome.

Để đặt các ứng dụng mặc định, hãy vào menu Cài đặt hệ thống (trên tất cả các bản phân phối Linux gần đây) và sau đó nhấp vào Chi tiết > Hệ thống > Ứng dụng mặc định. Danh sách chính xác sẽ khác với những gì được hiển thị ở đây trong ảnh chụp màn hình Ubuntu theo những gì thực sự được cài đặt và có sẵn trên hệ thống của bạn.



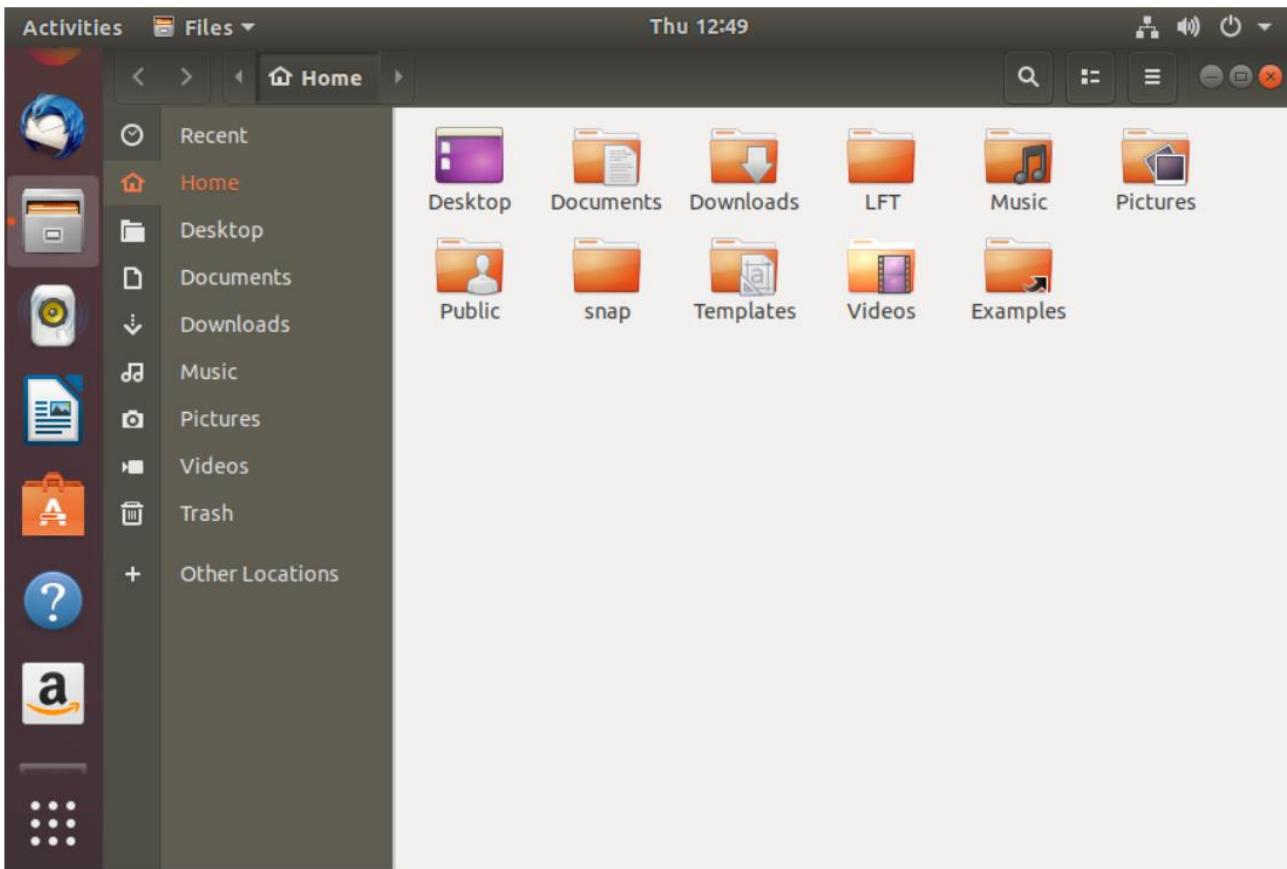
## Default Applications

3.6

### Quản lý tập tin

Mỗi bản phân phối thực hiện tiện ích Trình quản lý tệp, được sử dụng để điều hướng hệ thống tệp. Nó có thể xác định vị trí các tệp và khi tệp được nhấp vào, tệp sẽ chạy nếu đó là chương trình hoặc ứng dụng được liên kết sẽ được khởi chạy bằng cách sử dụng tệp dưới dạng dữ liệu. Hành vi này hoàn toàn quen thuộc với bất kỳ ai đã sử dụng các hệ điều hành khác.

Để khởi động Trình quản lý tệp, bạn sẽ phải định vị biểu tượng của nó, một tủ tệp, có thể dễ dàng tìm thấy trong mục Yêu thích hoặc Ứng dụng.



## File Manager

Trình quản lý tệp (Tệp trong trường hợp Ubuntu) sẽ mở một cửa sổ với thư mục Home của bạn được hiển thị. Bảng điều khiển bên trái của cửa sổ Trình quản lý tệp chứa danh sách các thư mục thường được sử dụng, như Máy tính, Trang chủ, Máy tính để bàn, Tài liệu, Tải xuống và Thùng rác.

Bạn có thể nhấp vào biểu tượng Kính lúp ở góc trên bên phải của cửa sổ Trình quản lý tệp để tìm kiếm tệp hoặc thư mục (thư mục).

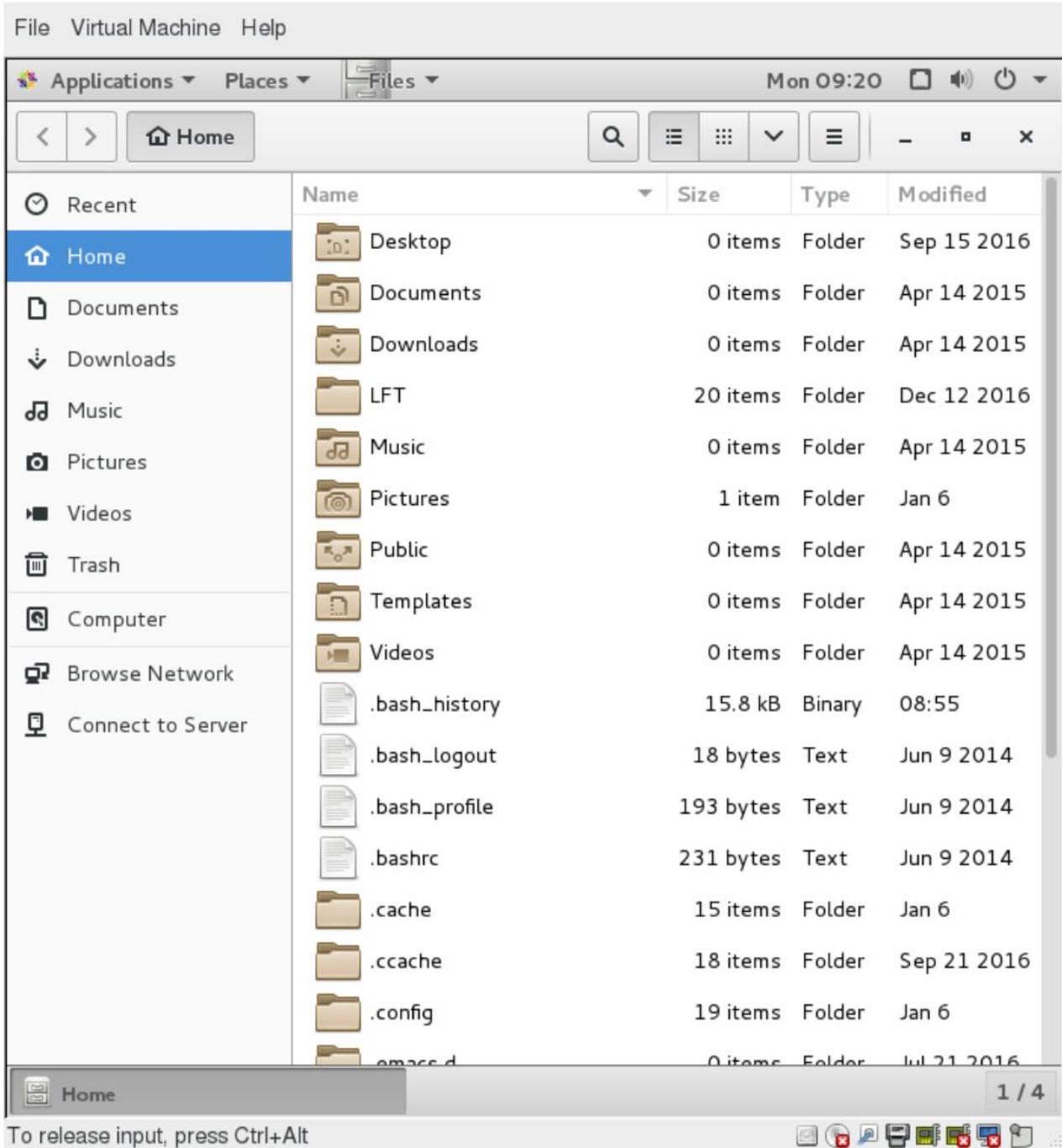
### 3.7 Thư mục nhà (Home Directories)

Trình quản lý tệp cho phép bạn truy cập các vị trí khác nhau trên máy tính và mạng, bao gồm thư mục Chính, Máy tính để bàn, Máy tính, Mạng và các thiết bị đính kèm khác. Tùy chọn Duyệt qua Mạng và Kết nối với Máy chủ truy cập các thiết bị được chia sẻ và kết nối mạng, chẳng hạn như máy chủ tệp và máy in có trên mạng cục bộ.

Mỗi người dùng có tài khoản trên hệ thống sẽ có một thư mục chính, thường được tạo dưới / home và thường được đặt tên theo người dùng, chẳng hạn như / home / student.

Theo mặc định, các tệp mà người dùng lưu sẽ được đặt trong cây thư mục bắt đầu từ đó. Tạo tài khoản, cho dù trong quá trình cài đặt hệ thống hoặc sau đó, khi người dùng mới được thêm vào, cũng tạo ra các thư mục mặc định được tạo trong thư mục chính của người dùng, chẳng hạn như Tài liệu, Máy tính để bàn và Tải xuống.

Trong ảnh chụp màn hình hiển thị cho CentOS 7, chúng tôi đã chọn định dạng danh sách và cũng đang hiển thị các tệp ẩn (những tệp bắt đầu bằng dấu chấm). Xem nếu bạn có thể làm tương tự trên phân phối của bạn.



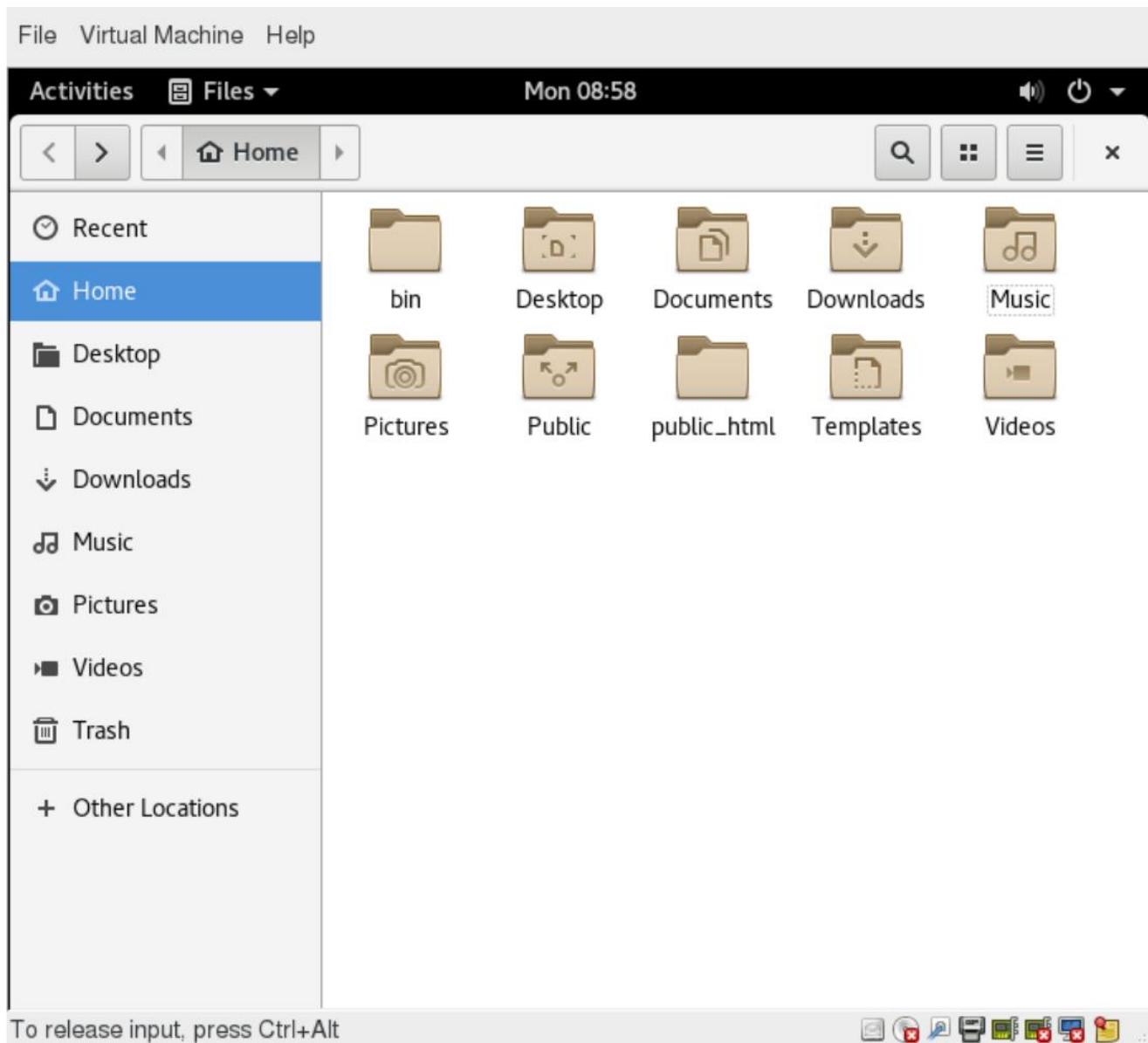
## Home Directories

### 3.8 Xem nội dung tập tin

Trình quản lý tập tin cho phép bạn xem các tệp và thư mục theo nhiều cách.

Bạn có thể chuyển đổi giữa các định dạng Biểu tượng và Danh sách, bằng cách nhấp vào các biểu tượng quen thuộc ở thanh trên cùng hoặc bạn có thể nhấn CTRL-1 hoặc CTRL-2 tương ứng.

Ngoài ra, bạn cũng có thể sắp xếp các tệp và thư mục theo tên, kích thước, loại hoặc ngày sửa đổi để sắp xếp thêm. Để làm như vậy, nhấp vào Xem và chọn Sắp xếp các mục.



## Viewing Files in openSUSE

Một tùy chọn hữu ích khác là hiển thị các tệp ẩn (đôi khi được gọi là tệp hệ thống không chính xác), thường là các tệp cấu hình được ẩn theo mặc định và tên bắt đầu bằng dấu chấm. Để hiển thị các tệp ẩn, chọn Hiển thị tệp ẩn từ menu hoặc nhấn CTRL-H.

Trình duyệt tệp cung cấp nhiều cách để tùy chỉnh chế độ xem cửa sổ của bạn để tạo điều kiện dễ dàng kéo và thả các thao tác tệp. Bạn cũng có thể thay đổi kích thước của các biểu tượng bằng cách chọn Phóng to và Thu nhỏ trong menu Xem.

### 3.9 Tìm kiếm tập tin

Trình quản lý tệp bao gồm một công cụ tìm kiếm tuyệt vời bên trong cửa sổ trình duyệt tệp.

Bấm Tìm kiếm trên thanh công cụ (để hiển thị hộp văn bản).

Nhập từ khóa vào hộp văn bản. Điều này khiến hệ thống thực hiện tìm kiếm để quy từ thư mục hiện tại cho bất kỳ tệp hoặc thư mục nào chứa một phần của từ khóa này.

Để mở Trình quản lý tệp từ dòng lệnh, trên hầu hết các hệ thống chỉ cần nhập nautilus.

Lưu ý: Cả hai phương pháp trên sẽ mở giao diện đồ họa cho chương trình.

Phím tắt để vào hộp văn bản tìm kiếm là CTRL-F. Bạn có thể thoát khỏi chế độ xem hộp văn bản tìm kiếm bằng cách nhấp vào nút Tìm kiếm hoặc CTRL-F một lần nữa.

Một cách nhanh chóng khác để truy cập vào một thư mục cụ thể là nhấn CTRL-L, nó sẽ cung cấp cho bạn hộp văn bản Vị trí để nhập đường dẫn đến thư mục.

Bạn có thể tinh chỉnh tìm kiếm của mình ngoài từ khóa ban đầu bằng cách cung cấp menu thả xuống để tiếp tục lọc tìm kiếm.

Dựa trên Vị trí hoặc Loại tệp, chọn tiêu chí bổ sung từ danh sách thả xuống.

Để tạo lại tìm kiếm, nhấp vào nút Tải lại

Để thêm nhiều tiêu chí tìm kiếm, nhấp vào nút + và chọn Tiêu chí tìm kiếm bổ sung.

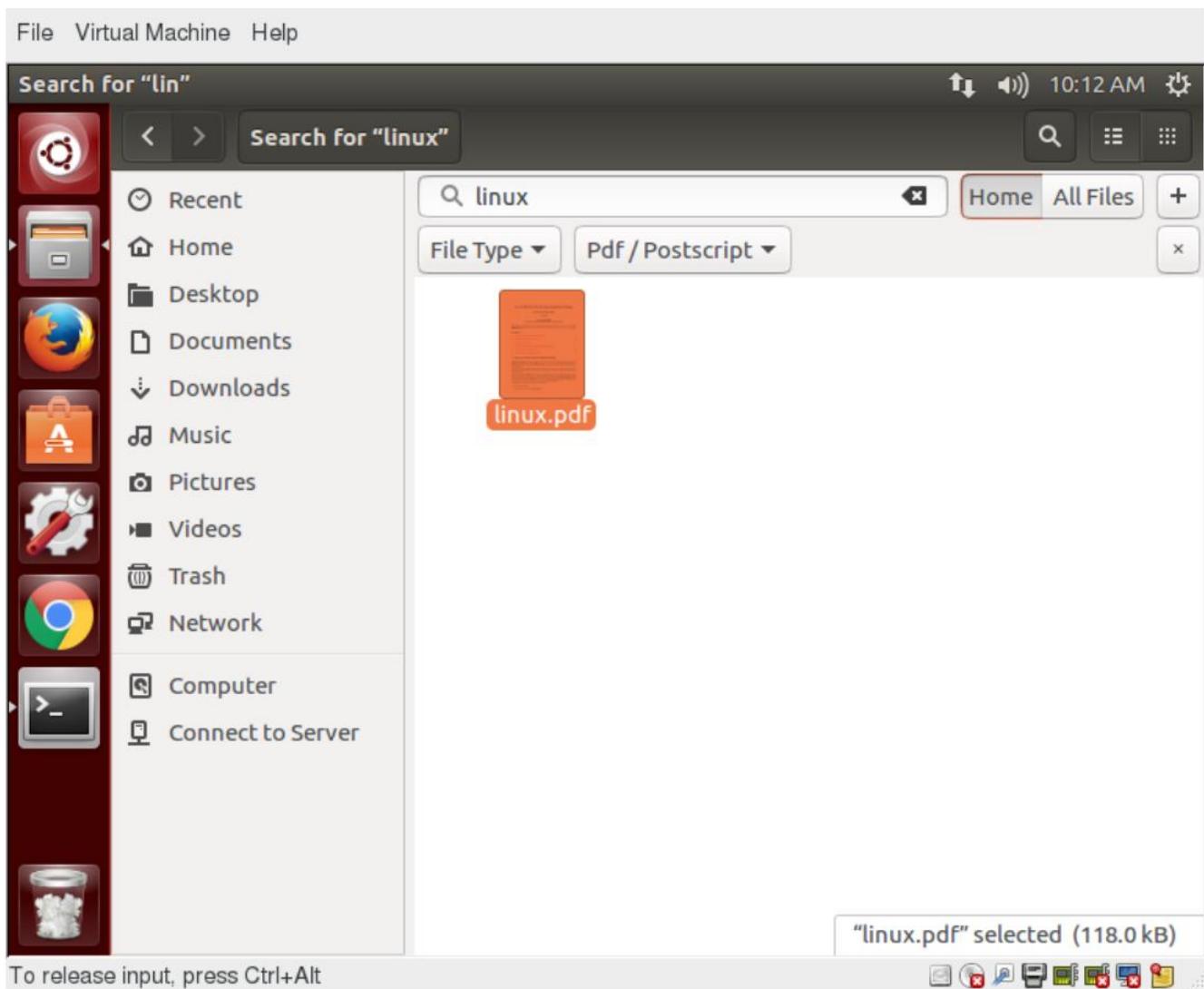
Ví dụ: nếu bạn muốn tìm một tệp PDF có chứa từ Linux trong thư mục Home của bạn, hãy điều hướng đến thư mục chính của bạn và tìm kiếm từ từ Linux Linux. Bạn sẽ thấy rằng tiêu chí tìm kiếm mặc định giới hạn tìm kiếm trong thư mục chính của bạn.

Để hoàn thành công việc, nhấp vào nút + để thêm một tiêu chí tìm kiếm khác, chọn Loại tệp cho loại tiêu chí và chọn PDF trong danh sách thả xuống Loại tệp.

### 3.10 Chính sửa tập tin

Chỉnh sửa bất kỳ tệp văn bản nào thông qua giao diện đồ họa rất dễ dàng trong môi trường máy tính để bàn Gnome. Chỉ cần nhấp đúp vào tệp trên màn hình nền hoặc trong cửa sổ trình duyệt tệp Nautilus để mở tệp bằng trình soạn thảo văn bản mặc định.

Trình soạn thảo văn bản mặc định trong Gnome là gedit. Nó đơn giản nhưng mạnh mẽ, lý tưởng để chỉnh sửa tài liệu, ghi chú nhanh và lập trình. Mặc dù gedit được thiết kế như một trình soạn thảo văn bản cho mục đích chung, nó cung cấp các tính năng bổ sung để



## Searching for Files

kiểm tra chính tả, tô sáng, liệt kê tệp và thống kê.

### 3.11 Xóa tập tin

Xóa một tệp trong Nautilus sẽ tự động di chuyển các tệp đã xóa vào thư mục .local / share / Trash / files / (một thùng rác sắp xếp) trong thư mục Home của người dùng. Có một số cách để xóa các tập tin và thư mục bằng Nautilus.

Chọn tất cả các tập tin và thư mục mà bạn muốn xóa.

Nhấn Xóa (trong Unity / KDE) hoặc CTRL-Xóa (trong Gnome) trên bàn phím của bạn. Hoặc, nhấp chuột phải vào tệp.

Chọn Di chuyển đến Thùng rác. Hoặc, làm nổi bật các tập tin.

Nhấp vào Chính sửa và Di chuyển đến Thùng rác thông qua giao diện đồ họa.

Để xóa vĩnh viễn một tập tin:

Trên bảng điều khiển bên trái trong cửa sổ trình duyệt tệp Nautilus, nhấp chuột phải vào thư mục Thùng rác.

Chọn Thùng rác.

Hoặc, chọn tệp hoặc thư mục bạn muốn xóa vĩnh viễn và nhấn Shift-Delete.

Để phòng ngừa, bạn không bao giờ nên xóa thư mục Home của mình, vì làm như vậy rất có thể sẽ xóa tất cả các tệp cấu hình Gnome của bạn và có thể ngăn bạn đăng nhập. Nhiều cấu hình hệ thống và chương trình cá nhân được lưu trữ trong thư mục Home của bạn.

### **3.12 Thiết lập các tham số hệ thống chạy Linux bằng giao diện đồ họa**

Bảng điều khiển Cài đặt hệ thống cho phép bạn kiểm soát hầu hết các tùy chọn cấu hình cơ bản và cài đặt máy tính để bàn, chẳng hạn như chỉ định độ phân giải màn hình, quản lý kết nối mạng hoặc thay đổi ngày và giờ của hệ thống.

Đối với Trình quản lý màn hình Gnome, một lần nhấp vào góc trên bên phải và sau đó chọn hình ảnh công cụ (tuốc nơ vít bắt chéo bằng cờ lê). Tùy thuộc vào bản phân phối của bạn, bạn cũng có thể tìm các cách khác để vào cấu hình cài đặt.

Các ảnh chụp màn hình ở đây cho thấy cách khởi chạy các cài đặt và màn hình ban đầu trông như thế nào trên cả hệ thống Fedora 28 và Red Hat 7. Giao diện chính xác trên hệ thống của bạn sẽ giống với một trong hai ví dụ này.

#### **System Settings Menus**

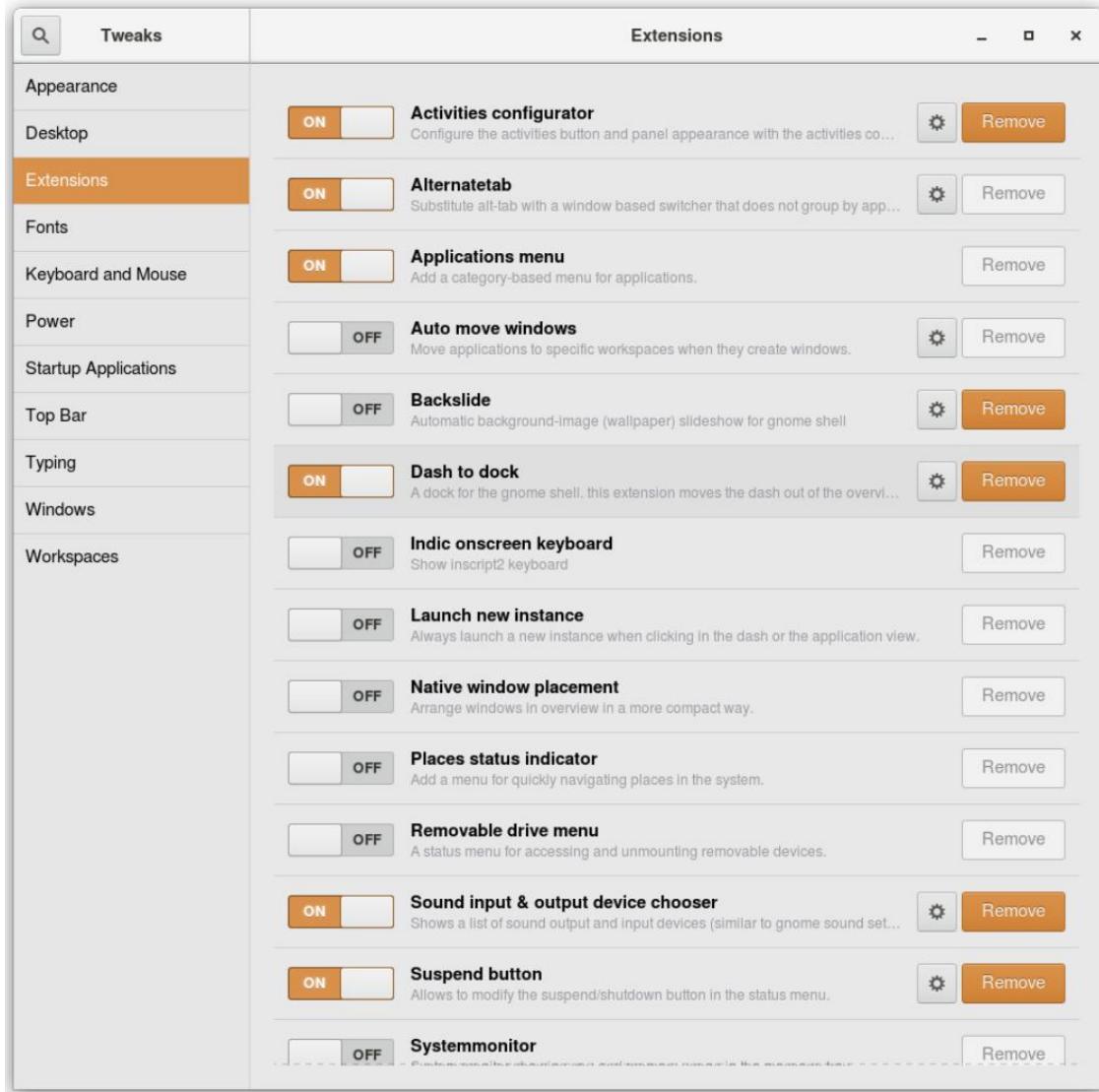
##### **Gnome-tweak-tool**

Để tìm hiểu sâu hơn về cấu hình, người ta có thể nhấp vào Thiết bị trên menu trước để định cấu hình các mục như màn hình, bàn phím, máy in, v.v.

Rất nhiều cài đặt cấu hình được cá nhân hóa không xuất hiện trên các menu cài đặt mà chúng ta đã thảo luận. Thay vào đó, bạn phải khởi chạy một công cụ có tên gnome-chỉnh-tool hoặc gnome-chỉnh trên các bản phân phối Linux gần đây. Chúng tôi chưa thực sự thảo luận về việc làm việc tại dòng lệnh, nhưng bạn luôn có thể khởi chạy một chương trình như thế này bằng cách thực hiện Alt-F2 và gõ lệnh. Có những bản phân phối có liên kết đến các menu chỉnh trong cài đặt, nhưng chủ yếu là vì một lý do bí ẩn nào đó, chúng che khuất sự tồn tại của công cụ không thể thiếu này.

Những điều quan trọng bạn có thể làm với công cụ này bao gồm chọn chủ đề, định cấu hình các tiện ích mở rộng mà bạn có thể nhận được từ bản phân phối hoặc tải xuống từ Internet, kiểm soát phông chữ và đặt chương trình nào bắt đầu khi bạn đăng nhập.

Ảnh chụp màn hình ở đây là từ hệ thống Red Hat 7 với khá nhiều phần mở rộng được cài đặt, nhưng không phải tất cả đều được sử dụng.



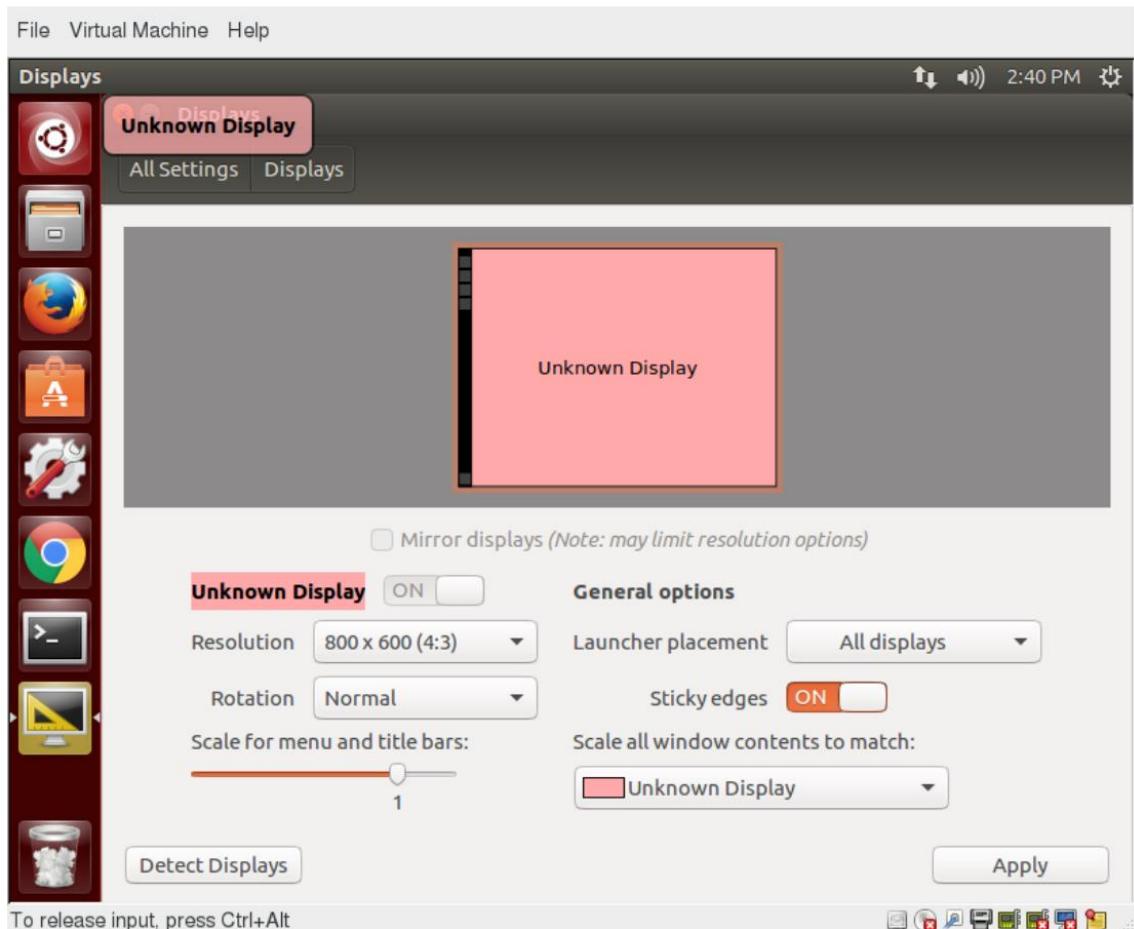
## Extensions Installed on Red Hat 7

## Display Settings

Bảng Hiển thị trong Cài đặt hệ thống chứa các cài đặt phổ biến nhất để thay đổi giao diện màn hình. Các cài đặt này hoạt động độc lập với trình điều khiển hiển thị cụ thể mà bạn đang chạy.

Tìm menu Cài đặt hiển thị có thẻ mất một số công việc thám tử. Trên các hệ thống gần đây, bạn có thể cần nhấp vào Cài đặt > Thiết bị > Hiển thị vì menu con sẽ không xuất hiện trong menu cài đặt chính. Ngoài ra, diện mạo thực tế có thể khá khác so với ảnh chụp màn hình được trình bày dưới đây; các phiên bản gần đây của cả Ubuntu và Fedora đều khá đơn giản nếu không có đồ họa.

Nếu hệ thống của bạn sử dụng trình điều khiển thẻ video độc quyền (thường là từ nVidia hoặc AMD), bạn có thể sẽ có một chương trình cấu hình cho trình điều khiển đó không có trong Cài đặt hệ thống. Chương trình này có thể cung cấp nhiều tùy chọn cấu hình hơn, nhưng cũng có thể phức tạp hơn và có thể yêu cầu quyền truy cập sysadmin (root). Nếu có thể, bạn nên định cấu hình cài đặt trong bảng Hiển thị thay vì chương trình cấu hình độc quyền.

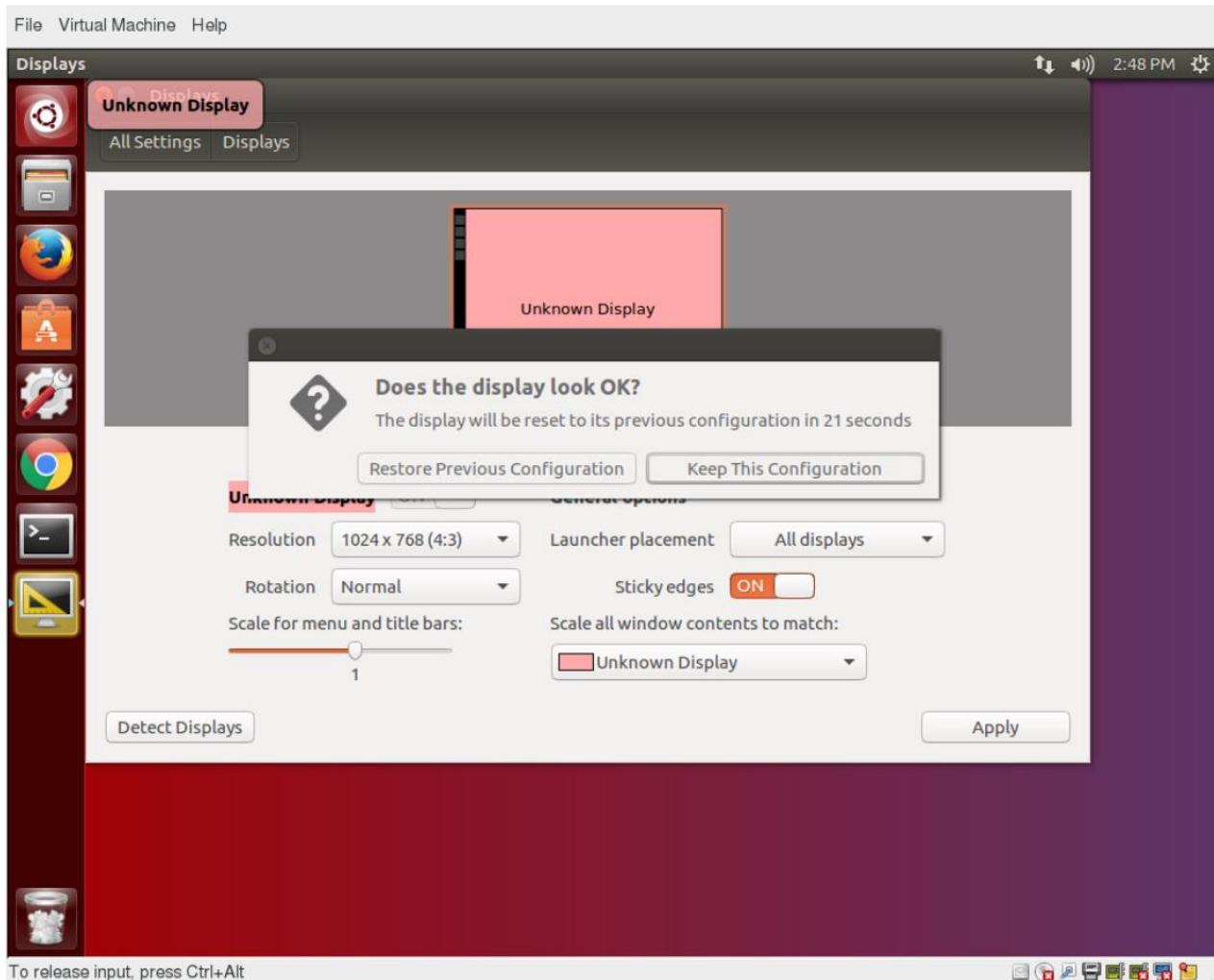


## Display Settings

Máy chủ X, thực sự cung cấp GUI, sử dụng tệp /etc/X11/xorg.conf làm tệp cấu hình nếu nó tồn tại. Trong các bản phân phối Linux hiện đại, tệp này thường chỉ xuất hiện trong các trường hợp bất thường, chẳng hạn như khi một số trình điều khiển đồ họa ít phổ biến hơn đang được sử dụng. Thay đổi trực tiếp tệp cấu hình này thường dành cho người dùng cao cấp hơn.

## Setting Resolution and Configuring Multiple Screens

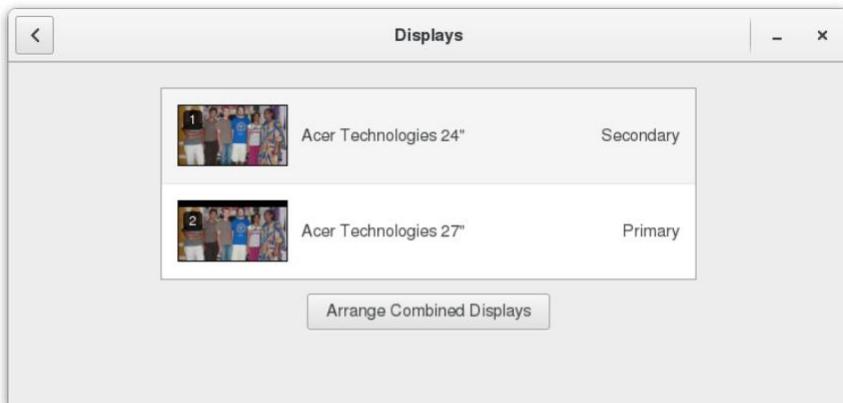
Mặc dù hệ thống của bạn thường sẽ tự động tìm ra độ phân giải tốt nhất cho màn hình của bạn, nhưng trong một số trường hợp, nó có thể bị lỗi này hoặc bạn có thể muốn thay đổi độ phân giải để đáp ứng nhu cầu của mình.



## Setting Resolution

Bạn có thể thực hiện việc này bằng bảng Hiển thị. Việc chuyển sang độ phân giải mới sẽ có hiệu lực khi bạn bấm Áp dụng, và sau đó xác nhận rằng độ phân giải đang hoạt động. Trong trường hợp độ phân giải được chọn không hoạt động hoặc bạn không hài lòng với ngoại hình, hệ thống sẽ chuyển về độ phân giải ban đầu sau một khoảng thời gian ngắn. Một lần nữa, sự xuất hiện chính xác của màn hình cấu hình sẽ thay đổi rất nhiều giữa các bản phân phối và phiên bản, nhưng thường khá trực quan và dễ dàng, một khi bạn tìm thấy các menu cấu hình.

Trong hầu hết các trường hợp, cấu hình cho nhiều màn hình được thiết lập tự động dưới dạng một màn hình lớn bao trùm tất cả các màn hình, sử dụng dự đoán hợp lý cho bộ cục màn hình. Nếu bộ cục màn hình không như mong muốn, một hộp kiểm có thể bật chế độ được nhân đôi, trong đó màn hình tương tự được nhìn thấy trên tất cả các màn hình. Trong ảnh chụp màn hình, nhấp vào một trong hai hình ảnh cho phép bạn định cấu hình độ phân giải của từng hình ảnh và liệu chúng có tạo một màn hình lớn hay phản chiếu cùng một video, v.v.



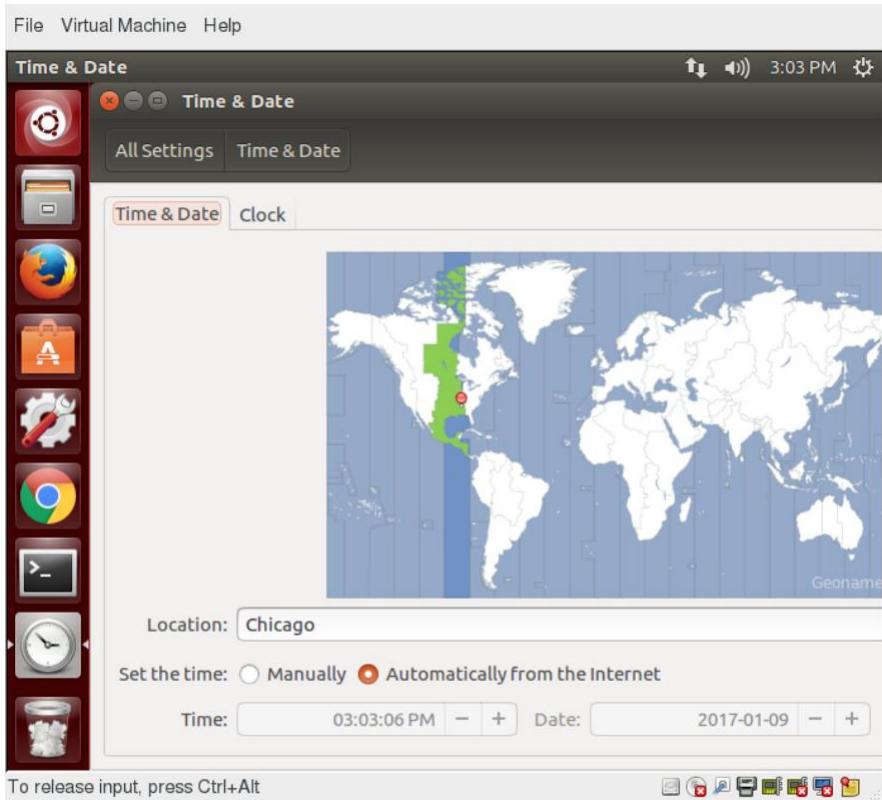
## Configuring Multiple Screens

### Date and Time Settings

Theo mặc định, Linux luôn sử dụng Giờ phôi hợp quốc tế (UTC) để chấm công nội bộ của riêng mình. Các giá trị thời gian được hiển thị hoặc lưu trữ dựa trên cài đặt múi giờ của hệ thống để có thời gian thích hợp. UTC tương tự, nhưng chính xác hơn, Giờ trung bình Greenwich (GMT).

Nếu bạn nhấp vào thời gian hiển thị trên bảng trên cùng, bạn có thể điều chỉnh định dạng mà ngày và giờ được hiển thị; trên một số bản phân phối, bạn cũng có thể thay đổi các giá trị.

Cửa sổ Cài đặt Ngày và Giờ chi tiết hơn có thể được truy cập từ cửa sổ Cài đặt hệ thống> Chi tiết.



## Date and Time Settings in Ubuntu

Ảnh chụp màn hình Ubuntu cũ hơn được hiển thị ở trên đẹp hơn về mặt đồ họa so với ảnh chụp màn hình Gnome tiêu chuẩn bên dưới (từ openSUSE), nhưng nội dung thông tin giống hệt nhau. Lưu ý rằng bạn phải "mở khóa" để thay đổi bất cứ điều gì, có nghĩa là nhập mật khẩu gốc.

Tuy nhiên, trong Ubuntu 18.04 LTS, màn hình đồ họa đẹp mắt đã được thay thế bằng cùng một màn hình đơn giản được sử dụng trong tất cả các máy tính để bàn Gnome gần đây.

Cài đặt "tự động" để cập đến việc sử dụng Giao thức thời gian mạng (NTP) mà chúng ta sẽ thảo luận tiếp theo.

### Network Time Protocol

Giao thức thời gian mạng (NTP) là giao thức phổ biến và đáng tin cậy nhất để đặt giờ địa phương thông qua các máy chủ Internet. Hầu hết các bản phân phối Linux bao gồm một thiết lập NTP đang hoạt động, để cập đến các máy chủ thời gian cụ thể được phân phối. Điều này có nghĩa là không cần thiết lập, ngoài "bật" hoặc "tắt", để đồng bộ hóa thời gian mạng. Nếu muốn, có thể cấu hình chi tiết hơn bằng cách chỉnh sửa tệp cấu hình NTP tiêu chuẩn (/etc/ntp.conf) cho các tiện ích NTP của Linux.

### 3.13 Cấu hình mạng

Tất cả các bản phân phối Linux có tệp cấu hình mạng, nhưng định dạng và vị trí tệp có thể khác nhau từ bản phân phối này đến bản phân phối khác. Chính sửa bằng tay các tệp này có thể xử lý các thiết lập khá phức tạp, nhưng không năng động hoặc dễ học và dễ sử dụng. Quản lý mạng được phát triển để làm cho mọi thứ dễ dàng hơn và thống nhất hơn trên các bản phân phối. Nó có thể liệt kê tất cả các mạng khả dụng (cả có dây và không dây), cho phép lựa chọn mạng băng rộng có dây, không dây hoặc di động, xử lý mật khẩu và thiết lập Mạng riêng ảo (VPN). Ngoại trừ các tình huống bất thường, tốt nhất là để tiên ích Trình quản lý mạng thiết lập các kết nối của bạn và theo dõi các cài đặt của bạn.

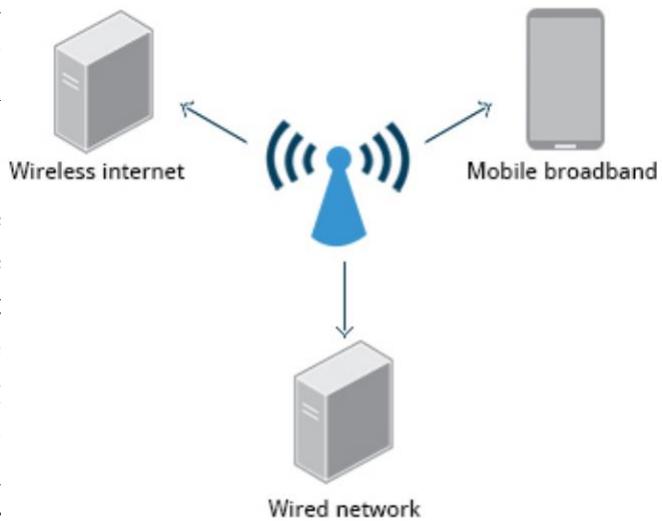
Trong phần này, bạn sẽ tìm hiểu cách quản lý các kết nối mạng, bao gồm kết nối có dây và không dây, kết nối băng thông rộng và VPN di động.

#### Wired and Wireless Connections

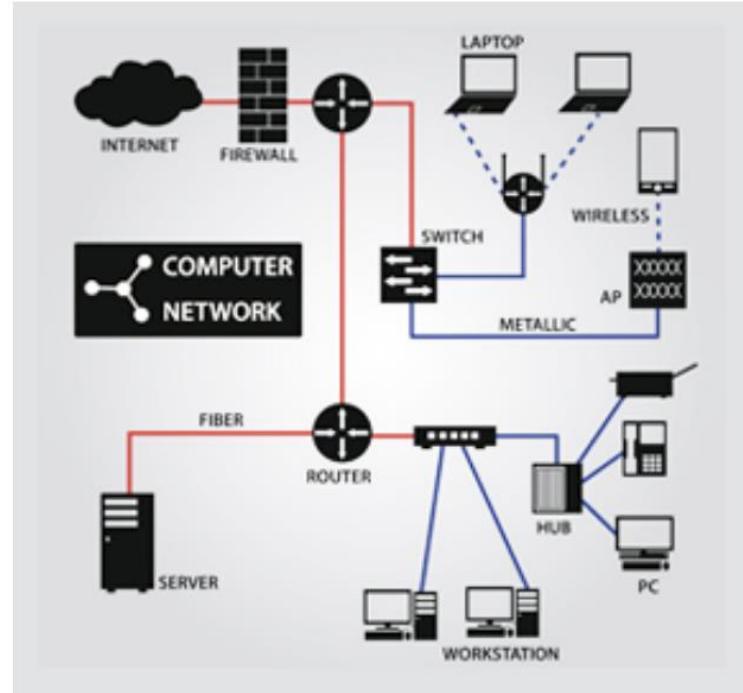
Kết nối có dây thường không yêu cầu cấu hình phức tạp hoặc thủ công. Giao diện phản ứng và sự hiện diện tín hiệu được tự động phát hiện và sau đó Trình quản lý mạng đặt cài đặt mạng thực tế thông qua Giao thức cấu hình máy chủ động (DHCP).

Đối với các cấu hình tĩnh không sử dụng DHCP, thiết lập thủ công cũng có thể được thực hiện dễ dàng thông qua Trình quản lý mạng. Bạn cũng có thể thay đổi địa chỉ Điều khiển truy cập phương tiện Ethernet (MAC) nếu phần cứng của bạn hỗ trợ. Địa chỉ MAC là số thập lục phân duy nhất của card mạng của bạn.

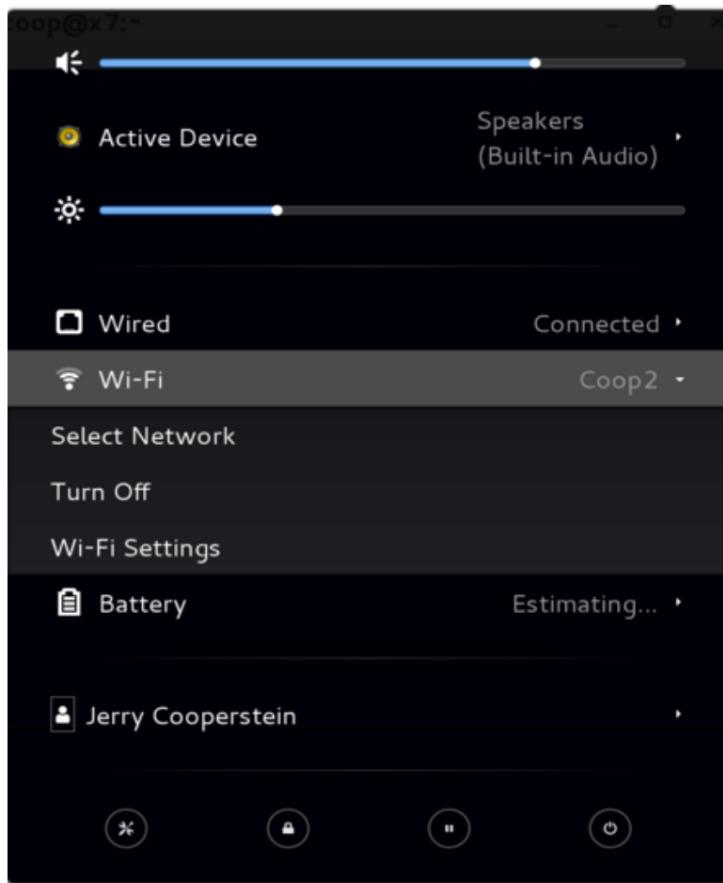
Mạng không dây không được kết nối với máy theo mặc định. Bạn có thể xem danh sách các mạng không dây khả dụng và xem mạng nào bạn được kết nối bằng cách sử dụng Trình quản lý mạng. Sau đó, bạn có thể thêm, chỉnh sửa hoặc xóa các mạng không dây đã biết và cũng chỉ định mạng nào bạn muốn kết nối theo mặc định khi có mặt.



#### Network Configuration



## Configuring Wireless Connections



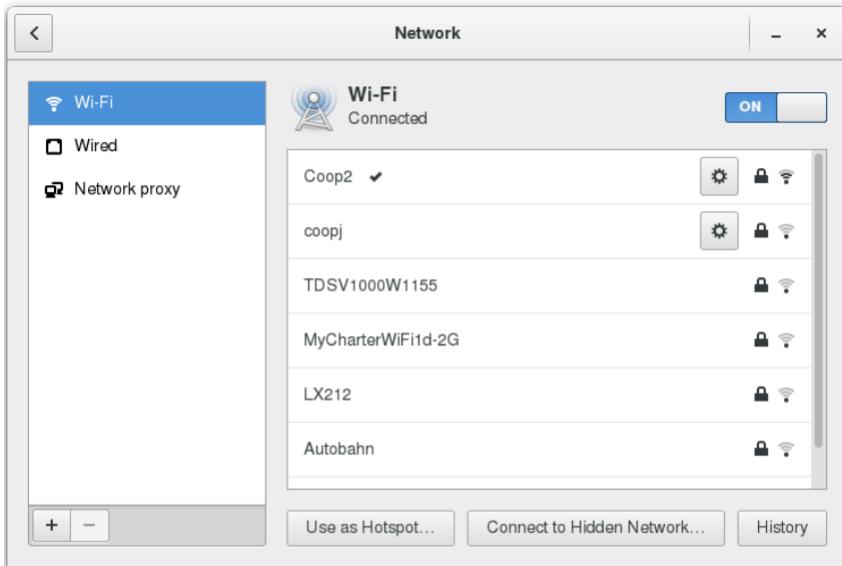
Để định cấu hình mạng không dây trong mọi phân phối dựa trên Gnome gần đây: Nhấp vào góc trên bên phải của bảng trên cùng, sẽ hiển thị cửa sổ cài đặt được trình bày bên dưới. Tiếp theo, chọn mục menu Wi-Fi.

Chọn mạng không dây bạn muốn kết nối. Nếu đó là một mạng an toàn, lần đầu tiên nó sẽ yêu cầu bạn nhập mật khẩu phù hợp. Theo mặc định, mật khẩu sẽ được lưu cho các kết nối tiếp theo.



## Selecting a Network

Nếu bạn nhấp vào Cài đặt Wi-Fi, bạn sẽ hiển thị ảnh chụp màn hình thứ ba. Nếu bạn nhấp vào biểu tượng Bánh răng cho bất kỳ kết nối nào, bạn có thể định cấu hình chi tiết hơn.



## Configuring the Network of Your Choice

Các bản phân phối Linux cũ hơn và khác có thể trông khá khác biệt về chi tiết, nhưng các bước và lựa chọn về cơ bản là giống hệt nhau, vì tất cả chúng đều đang chạy Trình quản lý mạng với quần áo có lẽ hơi khác nhau.

### Mobile Broadband and VPN Connections

Bạn có thể thiết lập kết nối băng thông rộng di động với Trình quản lý mạng, trình này sẽ khởi chạy trình hướng dẫn để thiết lập chi tiết kết nối cho mỗi kết nối.

Sau khi cấu hình xong, mạng được cấu hình tự động mỗi khi mạng băng thông rộng được đính kèm.

Quản lý mạng cũng có thể quản lý các kết nối VPN của bạn.

Nó hỗ trợ nhiều công nghệ VPN, chẳng hạn như IPSec gốc, Cisco OpenConnect (qua máy khách Cisco hoặc máy khách nguồn mở), Microsoft PPTP và OpenVPN.

Bạn có thể nhận được hỗ trợ cho VPN dưới dạng một gói riêng biệt từ nhà phân phối của bạn. Bạn cần cài đặt gói này nếu VPN ưa thích của bạn không được hỗ trợ.



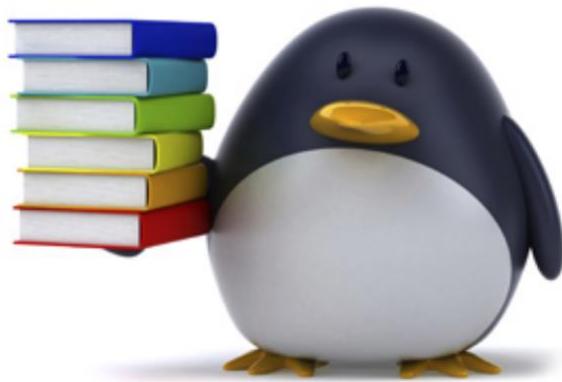
### 3.14 Cài đặt và nâng cấp các gói phần mềm

Mỗi gói trong bản phân phối Linux cung cấp một phần của hệ thống, chẳng hạn như nhân Linux, trình biên dịch C, mã phần mềm được chia sẻ để tương tác với các thiết bị USB hoặc trình duyệt web Firefox.

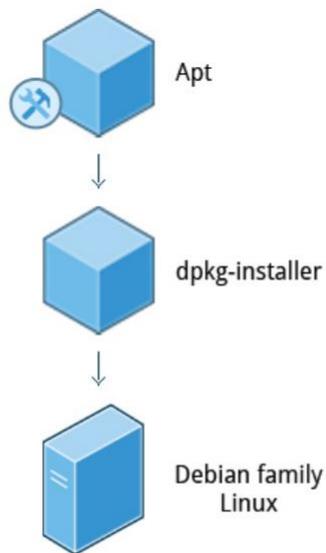
Các gói thường phụ thuộc vào nhau. Ví dụ: vì Firefox có thể giao tiếp bằng SSL / TLS, nên nó sẽ phụ thuộc vào gói cung cấp khả năng mã hóa và giải mã giao tiếp SSL và TLS và sẽ không cài đặt trừ khi gói đó cũng được cài đặt cùng lúc.

Một tiện ích xử lý các chi tiết cấp thấp của việc giải nén một gói và đặt các mảnh ở đúng nơi. Hầu hết thời gian, bạn sẽ làm việc với một tiện ích cấp cao hơn, biết cách tải xuống các gói từ Internet và có thể quản lý các nhóm và phụ thuộc cho bạn.

Trong phần này, bạn sẽ tìm hiểu cách cài đặt và cập nhật phần mềm trong Linux bằng hệ thống đóng gói Debian (cũng được sử dụng bởi các hệ thống như Ubuntu) và hệ thống đóng gói RPM (được sử dụng bởi cả hai hệ thống gia đình Red Hat và SUSE).



## Debian Packaging



## Package Management in the Debian Family System

nó trên nó (ví dụ: apt-get, aptitude, synaptic, Ubuntu v.v.). Mặc dù các kho apt thường tương thích với nhau, nhưng phần mềm chứa chúng

Hãy cùng xem xét Quản lý gói trong hệ thống gia đình Debian.

dpkg là trình quản lý gói cơ bản cho các hệ thống này. Nó có thể cài đặt, gỡ bỏ và xây dựng các gói. Không giống như các hệ thống quản lý gói cấp cao hơn, nó không tự động tải xuống và cài đặt các gói và đáp ứng các phụ thuộc của chúng.

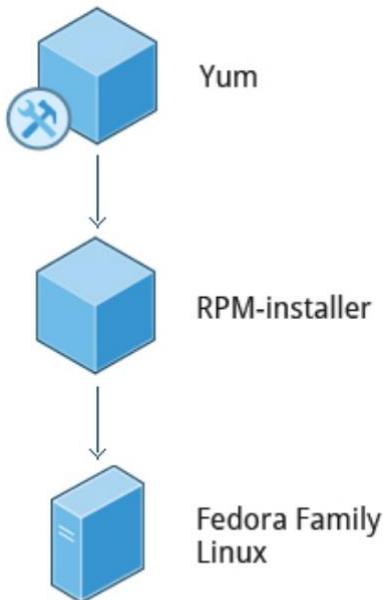
Đối với các hệ thống dựa trên Debian, hệ thống quản lý gói cấp cao hơn là hệ thống tiện ích apt (Công cụ gói nâng cao). Nói chung, trong khi mỗi bản phân phối trong gia đình Debian sử dụng apt, nó sẽ tạo giao diện người dùng của riêng

thường không có. Do đó, hầu hết các kho apt đều nhắm đến một phân phối cụ thể (như Ubuntu) và thường các nhà phân phối phần mềm giao hàng với nhiều kho để hỗ trợ nhiều phân phối. Trình diễn sử dụng Trung tâm phần mềm Ubuntu được hiển thị sau trong phần này.

## Red Hat Package Manager (RPM)

Red Hat Gói Manager (RPM) là hệ thống quản lý gói phổ biến trên các bản phân phối Linux. Nó được phát triển bởi Red Hat và được chấp nhận bởi một số bản phân phối khác, bao gồm openSUSE, Mandriva, CentOS, Oracle Linux và các bản phân phối khác.

Trình quản lý gói cáp cao khác nhau giữa các bản phân phối: Các bản phân phối gia đình Red Hat trước đây đã sử dụng định dạng kho lưu trữ được sử dụng bởi yum (Yellowdog Updater, Modified), mặc dù Fedora gần đây sử dụng thay thế, dnf, vẫn sử dụng cùng định dạng. Các bản phân phối gia đình SUSE cũng sử dụng RPM, nhưng sử dụng giao diện zypper. Dự án Gnome cũng sử dụng PackageKit làm giao diện hợp nhất.



## Red Hat Package Manager

### **3.15 Kết nối với thiết bị và dịch vụ bên ngoài**

#### **Thiết bị lưu trữ ngoài**

Khi gắn các thiết bị như máy in, máy quét, đĩa mềm, USB, CD/DVD, hầu như bạn không phải cài đặt thêm Driver, hệ thống sẽ tự động nhận. Hệ điều hành Ubuntu hỗ trợ tốt cơ chế Plug & Play (cắm và chạy).

Như đã nói ở Chương 3, khi gắn mỗi thiết bị vào máy tính của bạn nó sẽ tương ứng với một tập tin.

- Đĩa mềm - **/dev/floppy**
- Thiết bị lưu trữ (dùng chuẩn USB) - **/dev/sda, /dev/sdb,...**
- Đĩa CD/DVD - **/dev/hdc**

Để sử dụng được các thiết bị này, hệ thống **tự động** thao tác **gắn** (mount) thiết bị đó lên một thư mục (mount point). Như vậy, khi bạn thao tác (thêm/xóa/sửa) trên các thiết bị lưu trữ thực chất bạn chỉ làm việc trên các thư mục Chẳng hạn:

- **/dev/floppy** - **/media/floppy**
- **/dev/sda1** - **/media/usb**
- **/dev/hdc** - **/media/cdrom**

Do đó, **bạn phải gỡ** (umount) thiết bị ra, bằng cách click chuột phải lên thiết bị trên màn hình Desktop và chọn **Eject**.

#### **Cấu hình máy in**

Vào **System/Administration/Printing** để cấu hình máy in.

Trường hợp máy in cắm trực tiếp vào máy tính thì chọn **Local Printer** rồi bấm **Forward**. Chọn hãng sản xuất, loại máy in. Thí dụ máy in LaserJet 1320 của nhà sản xuất HP.

Bấm **Forward** và **Apply** để kết thúc.

Trường hợp cài máy in qua mạng, máy in được chia sẻ bởi một máy cài hệ điều hành Windows (active người dùng guest), ta chọn **Windows Printer (SMB)**.

#### **Kết nối Workgroup/Domain từ mạng Windows**

Để xem khai báo **Workgroup** hoặc **domain** hãy chọn:

**System/Administration/Shared Folders**

Chọn thẻ **General Properties** sau đó khai báo tên **Workgroup** hoặc **domain** (thí dụ OSS). Nhớ đánh dấu vào ô **This computer is a WINS server**

Bấm **Close** và khởi động lại máy

### Tìm kiếm các máy tính trong mạng LAN

Để xem các máy tính có trong mạng, chọn **Places/Network Servers**.

Sau đó mở **Windows Network** rồi vào một **Workgroup** (OSS) bạn sẽ thấy danh sách các tài nguyên trên mạng.

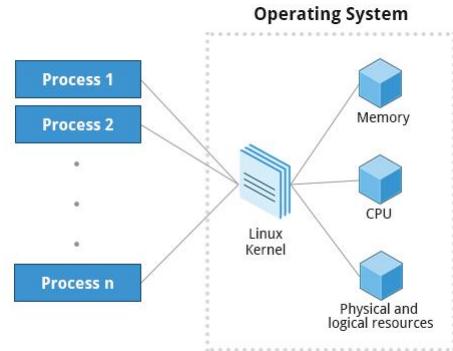
### 3.16 Câu hỏi ôn tập Chương 3

## Chương 4. Quản lý tiến trình trong Linux

### 4.1 Tiến trình là gì ?

Một tiến trình chỉ đơn giản là một thể hiện của một hoặc nhiều tác vụ (luồng) liên quan thực thi trên máy tính của bạn. Nó không giống như một chương trình hoặc một lệnh. Một lệnh duy nhất thực sự có thể bắt đầu một số tiến trình cùng một lúc. Một số tiến trình độc lập với nhau và những tiến trình khác có liên quan. Lỗi của một quá trình có thể hoặc không thể ảnh hưởng đến các tiến trình khác đang chạy trên hệ thống.

Các tiến trình sử dụng nhiều tài nguyên hệ thống, chẳng hạn như bộ nhớ, chu trình CPU (đơn vị xử lý trung tâm) và các thiết bị ngoại vi, như máy in và màn hình. Hệ điều hành (đặc biệt là kernel) chịu trách nhiệm phân bổ một phần thích hợp các tài nguyên này cho từng tiến trình và đảm bảo việc sử dụng hệ thống được tối ưu hóa tổng thể.



### 4.2 Phân loại tiến trình

Một cửa sổ đầu cuối (một loại vỏ lệnh) là một quá trình chạy miễn là cần thiết. Nó cho phép người dùng thực thi các chương trình và truy cập tài nguyên trong môi trường tương tác. Bạn cũng có thể chạy các chương trình trong nền, điều đó có nghĩa là chúng bị tách ra khỏi vỏ.

Các tiến trình có thể có nhiều loại khác nhau tùy theo nhiệm vụ đang được thực hiện. Dưới đây là một số loại tiến trình khác nhau, cùng với các mô tả và ví dụ của chúng:

Loại tiến trình mô tả bằng ví dụ

**Tiến trình tương tác:** cần được bắt đầu bởi người dùng, tại một dòng lệnh hoặc thông qua giao diện đồ họa như biểu tượng hoặc lựa chọn menu (bash, firefox...)

**Các tiến trình theo lô:** các tiến trình tự động được lên lịch từ đó và sau đó ngắt kết nối khỏi thiết bị đầu cuối. Các tác vụ này được xếp hàng và hoạt động trên cơ sở FIFO (nhập trước, xuất trước). (update)

**Các tiến trình daemons** (server xử lý chạy liên tục). Nhiều người được khởi chạy trong quá trình khởi động hệ thống và sau đó chờ người dùng hoặc yêu cầu hệ thống chỉ ra rằng dịch vụ của họ là bắt buộc. (httpd, xinetd, sshd)

**Các tiến trình chủ đề nhẹ** Đây là các tác vụ chạy dưới tiến trình chính, chia sẻ bộ nhớ và các tài nguyên khác, nhưng được hệ thống lên lịch và chạy trên cơ sở cá nhân. Một luồng riêng lẻ có thể kết thúc mà không kết thúc toàn bộ quá trình và một tiến trình có thể tạo

các luồng mới bất cứ lúc nào. Nhiều chương trình không tầm thường là đa luồng. (firefox, gnome-terminal-server)

**Tiến trình hạt nhân** Nhiệm vụ nhân Linux mà người dùng không bắt đầu cũng không chấm dứt và có ít quyền kiểm soát. Chúng có thể thực hiện các hành động như di chuyển một luồng từ CPU này sang CPU khác hoặc đảm bảo các hoạt động đầu vào / đầu ra vào đĩa được hoàn thành. (kthreadd, move, ksoftirqd)

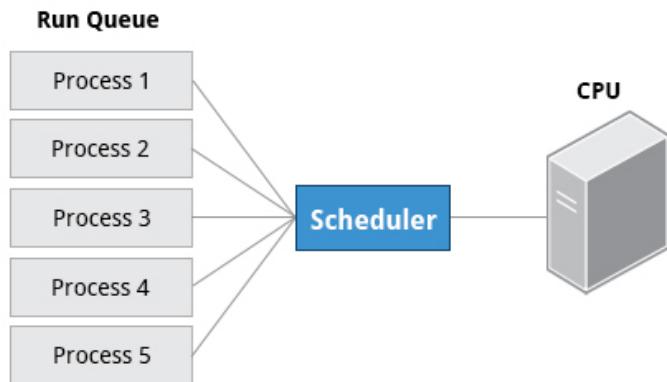
### 4.3 Định thời gian và trạng thái của tiến trình

Một hàm nhân quan trọng được gọi là bộ lập lịch liên tục thay đổi các quá trình bật và tắt CPU, chia sẻ thời gian theo mức độ ưu tiên tương đối, cần bao nhiêu thời gian và bao nhiêu đã được cấp cho một tác vụ.

Khi một tiến trình ở trạng thái được gọi là đang chạy, điều đó có nghĩa là nó hiện đang thực thi các hướng dẫn trên CPU hoặc đang chờ để được cấp một phần thời gian (một lát cắt thời gian) để nó có thể thực thi. Tất cả các tiến trình trong trạng thái này nằm trên cái được gọi là hàng đợi chạy và trên máy tính có nhiều CPU hoặc lõi, có một hàng đợi chạy trên mỗi máy.

Tuy nhiên, đôi khi các quá trình đi vào trạng thái được gọi là trạng thái ngủ, thường là khi chúng đang chờ điều gì đó xảy ra trước khi chúng có thể tiếp tục, có lẽ để người dùng gõ một cái gì đó. Trong điều kiện này, một quá trình đang ngủ trên hàng đợi.

Có một số trạng thái quá trình ít thường xuyên hơn, đặc biệt là khi một quá trình đang kết thúc. Đôi khi, một quá trình con hoàn thành, nhưng quá trình cha mẹ của nó không hỏi về trạng thái của nó. Thật thú vị, một quá trình như vậy được cho là ở trạng thái zombie; nó không thực sự sống nhưng vẫn xuất hiện trong danh sách các tiến trình của hệ thống



### 4.4 Định danh (ID) của tiến trình và chủ thể trạng thái (thread)

Tại bất kỳ thời điểm nào, luôn có nhiều quá trình được thực thi. Hệ điều hành theo dõi chúng bằng cách gán cho mỗi số ID tiến trình (PID) duy nhất. Bộ vi xử lý được sử dụng để theo dõi trạng thái quá trình, sử dụng CPU, sử dụng bộ nhớ, chính xác là nơi tài nguyên được đặt trong bộ nhớ và các đặc điểm khác.

Các bộ vi xử lý mới thường được chỉ định theo thứ tự tăng dần khi các tiến trình được sinh ra. Do đó, PID 1 biểu thị quá trình init (quá trình khởi tạo) và các quá trình thành công dần dần được gán số cao hơn.

Bảng giải thích các loại PID và mô tả của chúng:

### Mô tả loại ID

ID tiến trình (PID) Số ID tiến trình duy nhất

tiến trình ID cha mẹ (PPID) (Parent) đã bắt đầu quá trình này. Nếu cha mẹ chết, PPID sẽ đè cập đến cha mẹ nuôi; trên các hạt nhân gần đây, đây là kthreadd có PPID = 2.

ID chủ đè (TID) Số ID chủ đè. Điều này giống như PID cho các tiến trình đơn luồng. Đối với một tiến trình đa luồng, mỗi luồng chia sẻ cùng một PID, nhưng có một TID duy nhất.

## 4.5 Kết thúc tiến trình

Tại một số điểm, một trong những ứng dụng của bạn có thể ngừng hoạt động bình thường.

Làm thế nào để bạn loại bỏ nó?

Để chấm dứt một quá trình, bạn có thể gõ kill -SIGKILL <pid> hoặc kill -9 <pid>.

Lưu ý, tuy nhiên, bạn chỉ có thể giết các tiến trình của riêng bạn; những người thuộc về người dùng khác là vượt quá giới hạn, trừ khi bạn đã root.

```
File Edit View Search Terminal Help
c7:/tmp>ps
 PID TTY      TIME CMD
 2964 pts/2    00:00:00 bash
 31268 pts/2   00:00:00 cat
 31273 pts/2   00:00:00 ps
c7:/tmp>kill -9 31268
[1]+  Killed                 cat
c7:/tmp>ps
 PID TTY      TIME CMD
 2964 pts/2    00:00:00 bash
 31280 pts/2   00:00:00 ps
c7:/tmp>
```

## 4.6 Định danh của người dùng và nhóm

Nhiều người dùng có thể truy cập đồng thời một hệ thống và mỗi người dùng có thể chạy nhiều tiến trình. Hệ điều hành xác định người dùng bắt đầu quá trình bằng ID người dùng thực (RUID) được gán cho người dùng.

Người dùng xác định quyền truy cập cho người dùng được xác định bằng UID hiệu quả (EUID). EUID có thể hoặc không thể giống như RUID.

Người dùng có thể được phân loại thành các nhóm khác nhau. Mỗi nhóm được xác định bởi ID nhóm thực (RGID). Quyền truy cập của nhóm được xác định bởi ID nhóm hiệu quả (EGID). Mỗi người dùng có thể là thành viên của một hoặc nhiều nhóm.

Hầu hết thời gian chúng tôi bỏ qua các chi tiết này và chỉ nói về ID người dùng (UID).

### USER IDS



RUID

Identifies the user who started the process

### USER GROUP IDS



RGID

Identifies the group that started the process



EUID

Determines the access rights of the user

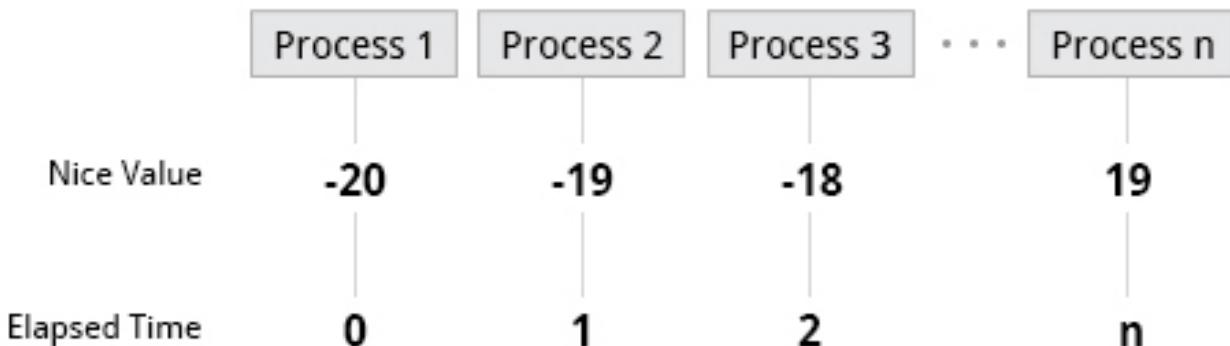


EGID

Determines the access rights of the group

#### 4.7 Mức độ ưu tiên của tiến trình

Tại bất kỳ thời điểm nào, nhiều tiến trình đang chạy (nghĩa là trong hàng đợi chạy) trên hệ thống. Tuy nhiên, CPU thực sự chỉ có thể chứa một nhiệm vụ tại một thời điểm, giống như một chiếc xe có thể chỉ có một người lái xe tại một thời điểm. Một số tiến trình quan trọng hơn các tiến trình khác, vì vậy Linux cho phép bạn thiết lập và thao tác ưu tiên tiến trình. Các tiến trình ưu tiên cao hơn được cấp nhiều thời gian hơn trên CPU.



Ưu tiên cho một tiến trình có thể được đặt bằng cách chỉ định một giá trị đẹp hoặc độ đẹp cho tiến trình. Giá trị tốt đẹp càng thấp, mức độ ưu tiên càng cao. Giá trị thấp được gán cho các tiến trình quan trọng, trong khi giá trị cao được gán cho các tiến trình có thể chờ lâu hơn. Một tiến trình có giá trị đẹp chỉ đơn giản là cho phép các tiến trình khác được thực hiện trước. Trong Linux, giá trị đẹp là -20 đại diện cho mức ưu tiên cao nhất và 19 đại diện cho mức thấp nhất. Điều này nghe có vẻ ngược, nhưng quy ước này, tiến trình càng đẹp, mức độ ưu tiên càng thấp, quay trở lại những ngày đầu tiên của UNIX.

Bạn cũng có thể chỉ định cái gọi là ưu tiên thời gian thực cho các tác vụ nhạy cảm với thời gian, chẳng hạn như điều khiển máy thông qua máy tính hoặc thu thập dữ liệu đến. Đây chỉ là một ưu tiên rất cao và không bị nhầm lẫn với cái được gọi là thời gian thực cũng khác biệt về mặt khái niệm và có nhiều việc phải làm để đảm bảo một công việc được hoàn thành trong một cửa sổ thời gian được xác định rõ.

#### 4.8 Mức trung bình tải

Tải trung bình là trung bình của số tải trong một khoảng thời gian nhất định. Nó đưa vào các tiến trình tài khoản đó là:

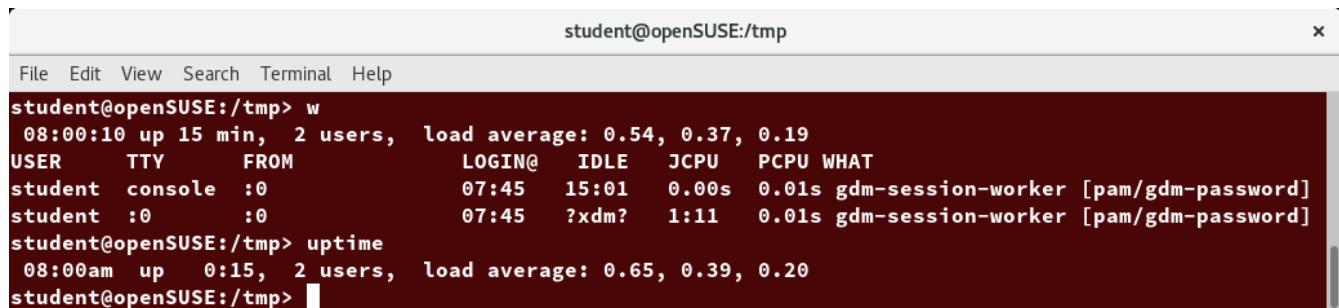
Tích cực chạy trên CPU

Được coi là có thể chạy được, nhưng chờ CPU có sẵn

Ngủ: tức là chờ đợi một loại tài nguyên nào đó (thông thường, I / O) sẽ có sẵn.

Lưu ý: Linux khác với các hệ điều hành giống như UNIX khác ở chỗ nó bao gồm các tiến trình ngủ. Hơn nữa, nó chỉ bao gồm cái gọi là giấc ngủ không bị gián đoạn, những thứ không thể đánh thức dễ dàng.

Có thể xem trung bình tải bằng cách chạy w, top hoặc uptime. Chúng tôi sẽ giải thích những con số trên trang tiếp theo.



A screenshot of a terminal window titled "student@openSUSE:/tmp". The window shows the output of several commands:

```
student@openSUSE:/tmp> w
 08:00:10 up 15 min, 2 users,  load average: 0.54, 0.37, 0.19
USER   TTY      FROM          LOGIN@    IDLE     JCPU   PCPU WHAT
student  console :0          07:45   15:01   0.00s  0.01s gdm-session-worker [pam/gdm-password]
student  :0            07:45   ?xdm?    1:11   0.01s gdm-session-worker [pam/gdm-password]
student@openSUSE:/tmp> uptime
 08:00am up  0:15, 2 users,  load average: 0.65, 0.39, 0.20
student@openSUSE:/tmp>
```

### Giải nghĩa mức trung bình tải

Trung bình tải được hiển thị bằng ba bộ số khác nhau, như trong ví dụ sau:

Phần cuối cùng của thông tin là tải trung bình của hệ thống. Giả sử hệ thống của chúng tôi là một hệ thống CPU đơn, ba số trung bình tải được hiểu như sau:

0,45: Trong phút cuối, hệ thống đã được sử dụng trung bình 45%.

0,17: Trong 5 phút sử dụng cuối cùng là 17%.

0,12: Trong 15 phút sử dụng cuối cùng là 12%.

Nếu chúng ta thấy giá trị 1,00 ở vị trí thứ hai, điều đó có nghĩa là hệ thống CPU đơn được sử dụng 100%, trung bình, trong 5 phút qua; Điều này là tốt nếu chúng ta muốn sử dụng đầy đủ một hệ thống. Giá trị trên 1,00 cho một hệ thống CPU đơn ngụ ý rằng hệ thống đã được sử dụng quá mức: có nhiều tiến trình cần CPU hơn CPU có sẵn.

Nếu chúng ta có nhiều CPU, giả sử hệ thống bốn CPU, chúng ta sẽ chia số lượng trung bình tải cho số lượng CPU. Trong trường hợp này, ví dụ, nhìn thấy trung bình tải 1 phút là 4,00 ngụ ý rằng toàn bộ hệ thống được sử dụng 100% ( $4,00 / 4$ ) trong phút cuối.

Tăng ngắn hạn thường không phải là một vấn đề. Một đỉnh cao mà bạn thấy có khả năng là một sự bùng nổ của hoạt động, không phải là một cấp độ mới. Ví dụ, khi khởi động, nhiều quá trình bắt đầu và sau đó hoạt động lắng xuống. Nếu một đỉnh cao được nhìn thấy trong trung bình tải 5 và 15 phút, điều đó có thể gây lo ngại.

```
File Edit View Search Terminal Help
c7:/home/coop>w
 08:09:25 up 54 min, 5 users, load average: 0.45, 0.17, 0.12
USER     TTY      FROM          LOGIN@    IDLE      JCPU      PCPU WHAT
coop     :0       :0            07:15     ?xdm?      5:57      0.06s gdm-se
s
coop     pts/0    :0            07:15     37.00s   0.08s   1.82s /usr/l
i
coop     pts/1    :0            07:15     10:48    0.03s   0.03s bash
coop     pts/2    :0            07:15     11:09    0.04s   0.04s bash
coop     pts/3    :0            07:15     5.00s   0.05s   0.00s w
c7:/home/coop>
```

#### 4.9 Các tiến trình chạy nền và trên mặt

Linux hỗ trợ xử lý công việc nền và tiền cảnh. Một công việc trong bối cảnh này chỉ là một lệnh được khởi chạy từ cửa sổ terminal. Các công việc tiền cảnh chạy trực tiếp từ shell và khi một công việc nền trước đang chạy, các công việc khác cần chờ truy cập shell (ít nhất là trong cửa sổ terminal đó nếu sử dụng GUI) cho đến khi hoàn thành. Điều này là tốt khi công việc hoàn thành nhanh chóng. Nhưng điều này có thể có tác động bất lợi nếu công việc hiện tại sẽ mất nhiều thời gian (thậm chí vài giờ) để hoàn thành.

Trong những trường hợp như vậy, bạn có thể chạy công việc trong nền và giải phóng shell cho các tác vụ khác. Công việc nền sẽ được thực hiện ở mức ưu tiên thấp hơn, do đó, sẽ cho phép thực hiện trọn vẹn các tác vụ tương tác và bạn có thể nhập các lệnh khác trong cửa sổ đầu cuối trong khi công việc nền đang chạy. Theo mặc định, tất cả các công việc được thực hiện ở nền trước. Bạn có thể đặt một công việc trong nền bằng cách thêm & vào lệnh, ví dụ: update&.

```
File Edit View Search Terminal Help
c7:/usr/src/linux>make O=../linux-7 > /dev/null &
[1] 25682
c7:/usr/src/linux>sleep 100
^Z
[2]+  Stopped                  sleep 100
c7:/usr/src/linux>bg
[2]+ sleep 100 &
c7:/usr/src/linux>fg
sleep 100
^C
c7:/usr/src/linux>Succeed: decoded and checked 2145844 instructions

[1]+  Done                    make O=../linux-7 > /dev/null
c7:/usr/src/linux>
```

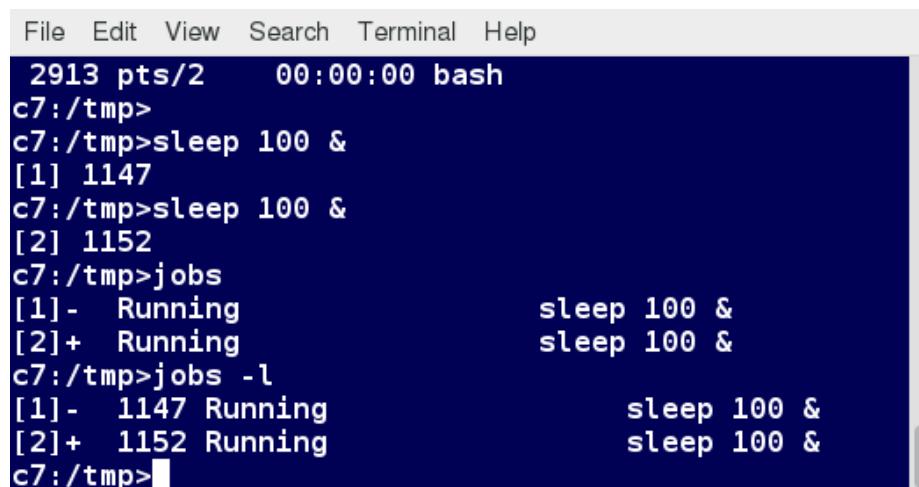
Bạn có thể sử dụng CTRL-Z để tạm dừng một công việc nền trước hoặc CTRL-C để chấm dứt một công việc nền trước và luôn có thể sử dụng các lệnh bg và fg để chạy một quá trình trong nền và tiền cảnh, tương ứng.

#### 4.10 Quản lý các công việc (jobs)

Tiện ích **jobs** hiển thị tất cả các công việc đang chạy trong nền. Màn hình hiển thị ID công việc, trạng thái và tên lệnh, như hiển thị ở đây.

**jobs -l** cung cấp thông tin giống như các công việc, bao gồm cả PID của các công việc nền.

Các công việc nền được kết nối với cửa sổ đầu cuối, vì vậy, nếu bạn đăng xuất, tiện ích công việc sẽ không hiển thị các công việc bắt đầu từ cửa sổ đó.



```
File Edit View Search Terminal Help
2913 pts/2    00:00:00 bash
c7:/tmp>
c7:/tmp>sleep 100 &
[1] 1147
c7:/tmp>sleep 100 &
[2] 1152
c7:/tmp>jobs
[1]-  Running                  sleep 100 &
[2]+  Running                  sleep 100 &
c7:/tmp>jobs -l
[1]-  1147 Running             sleep 100 &
[2]+  1152 Running             sleep 100 &
c7:/tmp>
```

#### Lệnh ps (System V Style)

**ps** cung cấp thông tin về các tiến trình hiện đang chạy được khóa bởi PID. Nếu bạn muốn cập nhật lặp đi lặp lại trạng thái này, bạn có thể sử dụng các biến thể được cài đặt hàng đầu hoặc thường được cài đặt khác, chẳng hạn như htop hoặc trên đỉnh, từ dòng lệnh hoặc gọi ứng dụng giám sát hệ thống đồ họa phân phối của bạn.

ps có nhiều tùy chọn để chỉ định chính xác những nhiệm vụ cần kiểm tra, thông tin nào sẽ hiển thị về chúng và chính xác nên sử dụng định dạng đầu ra nào.

```
student@ubuntu:~$ ps -ef
UID      PID  PPID  C STIME TTY          TIME CMD
root      1      0  2 16:07 ?        00:00:02 /sbin/init auto noprompt
root      2      0  0 16:07 ?        00:00:00 [kthreadd]
root      3      2  0 16:07 ?        00:00:00 [ksoftirqd/0]
root      4      2  0 16:07 ?        00:00:00 [kworker/0:0]
root      5      2  0 16:07 ?        00:00:00 [kworker/0:0H]
root      6      2  0 16:07 ?        00:00:00 [kworker/u256:0]
root      7      2  0 16:07 ?        00:00:00 [rcu_sched]
root      8      2  0 16:07 ?        00:00:00 [rcu_bh]
root      9      2  0 16:07 ?        00:00:00 [migration/0]
root     10      2  0 16:07 ?        00:00:00 [watchdog/0]
root     11      2  0 16:07 ?        00:00:00 [watchdog/1]
root     12      2  0 16:07 ?        00:00:00 [migration/1]
root     13      2  0 16:07 ?        00:00:00 [ksoftirqd/1]
.....
student   3669  3084  0 16:08 ?        00:00:00 /usr/lib/gnome-terminal/gnome-terminal-server
student   3675  3669  0 16:08 pts/18    00:00:00 bash
student   3701  3324  0 16:08 ?        00:00:00 zeitgeist-databub
student   3712  3708  0 16:08 ?        00:00:00 /usr/bin/zeitgeist-daemon
student   3719  3084  0 16:08 ?        00:00:00 /usr/lib/x86_64-linux-gnu/zeitgeist-fts
student   3785  3324  0 16:09 ?        00:00:00 update-notifier
root     3792      1  22 16:09 ?        00:00:02 /usr/bin/python3 /usr/sbin/aptd
student  4074  3675  0 16:09 pts/18    00:00:00 ps -ef
student@ubuntu:~$
```

Không có tùy chọn, ps sẽ hiển thị tất cả các tiến trình đang chạy trong trình bao hiện tại. Bạn có thể sử dụng tùy chọn -u để hiển thị thông tin của các tiến trình cho tên người dùng được chỉ định. Lệnh ps -ef hiển thị tất cả các tiến trình trong hệ thống một cách chi tiết. Lệnh ps -eLf tiến thêm một bước và hiển thị một dòng thông tin cho mỗi luồng (hãy nhớ rằng, một quá trình có thể chứa nhiều luồng).

## Lệnh ps (BSD Style)

```

File Edit View Search Terminal Help
c7:/home/coop>ps aux | head -10
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      1  0.0  0.0 125680  5544 ?      Ss  07:34  0:01 /usr/lib/systemd/sys
root      2  0.0  0.0     0   0 ?      S  07:34  0:00 [kthreadd]
root      3  0.0  0.0     0   0 ?      S  07:34  0:00 [ksoftirqd/0]
root      5  0.0  0.0     0   0 ?      S< 07:34  0:00 [kworker/0:0H]
root      7  0.0  0.0     0   0 ?      S  07:34  0:02 [rcu_preempt]
root      8  0.0  0.0     0   0 ?      S  07:34  0:00 [rcu_sched]
root      9  0.0  0.0     0   0 ?      S  07:34  0:00 [rcu_bh]
root     10  0.0  0.0     0   0 ?      S  07:34  0:00 [rcuop/0]
root     11  0.0  0.0     0   0 ?      S  07:34  0:00 [rcuos/0]
c7:/home/coop>ps axo stat,priority,pid,pcpu,comm | head -10
STAT PRI  PID %CPU COMMAND
Ss    20    1  0.0 systemd
S    20    2  0.0 kthreadd
S    20    3  0.0 ksoftirqd/0
S<    0    5  0.0 kworker/0:0H
S    20    7  0.0 rcu_preempt
S    20    8  0.0 rcu_sched
S    20    9  0.0 rcu_bh
S    20   10  0.0 rcuop/0
S    20   11  0.0 rcuos/0
c7:/home/coop>

```

**ps** có một kiểu đặc tả tùy chọn khác, xuất phát từ loại UNIX của BSD, trong đó các tùy chọn được chỉ định mà không có dấu gạch ngang trước. Ví dụ, lệnh **ps aux** hiển thị tất cả các tiến trình của tất cả người dùng. Lệnh **ps axo** cho phép bạn chỉ định thuộc tính nào bạn muốn xem. Ảnh chụp màn hình cho thấy một đầu ra mẫu của **ps** với vòng loại **ax** và **xo**.

## 4.11 Cây tiến trình

**pstree** hiển thị các tiến trình đang chạy trên hệ thống dưới dạng sơ đồ cây cho thấy mối quan hệ giữa một tiến trình và tiến trình mẹ của nó và bất kỳ tiến trình nào khác mà nó tạo ra. Các mục lặp lại của một quá trình không được hiển thị và các luồng được hiển thị trong dấu ngoặc nhọn.

```
student@ubuntu:~/Pictures$ pstree
systemd--ModemManager---{gdbus}
                         |   {gmain}
                         |
                         NetworkManager---dhclient
                                 |   dnsmasq
                                 |   {gdbus}
                                 |   {gmain}
                         accounts-daemon---{gdbus}
                                 |   {gmain}
                         acpid
                         getty
                         apache2---2*[apache2---26*[{apache2}]]
                         avahi-daemon---avahi-daemon
                         cgmanager
                         collectl
                         colord---{gdbus}
                                 |   {gmain}
                         cron
                         cups-browsed---{gdbus}
                                 |   {gmain}
```

## 4.12 Lệnh top

Mặc dù chế độ xem tinh về những gì hệ thống đang làm là hữu ích, việc theo dõi hiệu suất hệ thống trực tiếp theo thời gian cũng có giá trị. Một lựa chọn sẽ là chạy ps đều đặn, cứ sau hai phút. Một cách khác tốt hơn là sử dụng hàng đầu để nhận các cập nhật thời gian thực liên tục (cứ sau hai giây theo mặc định), cho đến khi bạn thoát bằng cách nhập q. top rõ ràng làm nổi bật quá trình nào đang tiêu thụ nhiều chu kỳ và bộ nhớ CPU nhất (sử dụng các lệnh thích hợp từ bên trong).

### Dòng đầu tiên của kết xuất lệnh top

Dòng đầu tiên của đầu ra hàng đầu hiển thị một bản tóm tắt nhanh chóng về những gì đang xảy ra trong hệ thống, bao gồm:

File Edit View Search Terminal Help											
top - 10:52:22 up 3:17, 5 users, load average: 1.11, 0.43, 0.20											
Tasks: 333 total, 3 running, 330 sleeping, 0 stopped, 0 zombie											
%Cpu(s): 24.6 us, 5.5 sy, 0.0 ni, 65.8 id, 4.0 wa, 0.0 hi, 0.0 si, 0.0 st											
KiB Mem : 16283344 total, 9176888 free, 1713680 used, 5392776 buff/cache											
KiB Swap: 8290300 total, 8290300 free, 0 used. 13387900 avail Mem											
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
10831	root	20	0	113636	2512	2256	R	73.1	0.0	0:02.20	awk
10830	root	20	0	141180	23968	3232	S	14.6	0.1	0:00.44	objdump
2442	coop	20	0	2080636	280092	86628	S	12.6	1.7	2:29.41	gnome-shell
10832	root	20	0	4196	676	596	R	5.0	0.0	0:00.15	test_get_len
1449	root	20	0	467744	44588	29032	S	2.0	0.3	1:15.14	Xorg
3819	coop	20	0	539608	160724	63500	S	2.0	1.0	3:06.89	skype-bin
7492	coop	20	0	1184072	228944	84920	S	1.3	1.4	0:17.70	thunderbird
4428	coop	20	0	644404	55776	49988	S	1.0	0.3	0:00.23	gnome-screensho
4058	coop	20	0	1492732	327060	108116	S	0.7	2.0	2:04.10	chrome
28	root	rt	0	0	0	0	S	0.3	0.0	0:00.03	migration/2
68	root	20	0	0	0	0	S	0.3	0.0	0:00.49	rcuop/6
2372	coop	20	0	1492444	63244	28244	S	0.3	0.4	0:02.38	gnome-settings-
2758	coop	20	0	659456	65372	51556	S	0.3	0.4	0:03.99	gnome-terminal-
1	root	20	0	125680	5544	3676	S	0.0	0.0	0:01.55	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
7	root	20	0	0	0	0	S	0.0	0.0	0:02.61	rcu_prempt
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_sched
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
10	root	20	0	0	0	0	S	0.0	0.0	0:00.43	rcuop/0
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuos/0

Hệ thống đã hoạt động được bao lâu rồi

Có bao nhiêu người dùng đang đăng nhập

Trung bình tải là gì.

Trung bình tải xác định mức độ bận rộn của hệ thống. Trung bình tải 1,00 trên mỗi CPU cho biết hệ thống được đăng ký đầy đủ, nhưng không bị quá tải. Nếu trung bình tải vượt quá giá trị này, nó chỉ ra rằng các tiến trình đang cạnh tranh về thời gian của CPU. Nếu trung bình tải rất cao, nó có thể chỉ ra rằng hệ thống đang gặp sự cố, chẳng hạn như quá trình chạy trốn (một quá trình ở trạng thái không phản hồi).

File Edit View Search Terminal Help											
<b>top - 10:52:22 up 3:17, 5 users, load average: 1.11, 0.43, 0.20</b>											
Tasks: 333 total, 3 running, 330 sleeping, 0 stopped, 0 zombie											
%Cpu(s): 24.6 us, 5.5 sy, 0.0 ni, 65.8 id, 4.0 wa, 0.0 hi, 0.0 si, 0.0 st											
KiB Mem : 16283344 total, 9176888 free, 1713680 used, 5392776 buff/cache											
KiB Swap: 8290300 total, 8290300 free, 0 used. 13387900 avail Mem											
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
10831	root	20	0	113636	2512	2256	R	73.1	0.0	0:02.20	awk
10830	root	20	0	141180	23968	3232	S	14.6	0.1	0:00.44	objdump
2442	coop	20	0	2080636	280092	86628	S	12.6	1.7	2:29.41	gnome-shell
<b>10832</b>	<b>root</b>	<b>20</b>	<b>0</b>	<b>4196</b>	<b>676</b>	<b>596</b>	<b>R</b>	<b>5.0</b>	<b>0.0</b>	<b>0:00.15</b>	<b>test_get_len</b>
1449	root	20	0	467744	44588	29032	S	2.0	0.3	1:15.14	Xorg
3819	coop	20	0	539608	160724	63500	S	2.0	1.0	3:06.89	skype-bin
7492	coop	20	0	1184072	228944	84920	S	1.3	1.4	0:17.70	thunderbird
4428	coop	20	0	644404	55776	49988	S	1.0	0.3	0:00.23	gnome-screensho
4058	coop	20	0	1492732	327060	108116	S	0.7	2.0	2:04.10	chrome
28	root	rt	0	0	0	0	S	0.3	0.0	0:00.03	migration/2
68	root	20	0	0	0	0	S	0.3	0.0	0:00.49	rcuop/6
2372	coop	20	0	1492444	63244	28244	S	0.3	0.4	0:02.38	gnome-settings-
2758	coop	20	0	659456	65372	51556	S	0.3	0.4	0:03.99	gnome-terminal-
1	root	20	0	125680	5544	3676	S	0.0	0.0	0:01.55	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
7	root	20	0	0	0	0	S	0.0	0.0	0:02.61	rcu_preempt
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_sched
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
10	root	20	0	0	0	0	S	0.0	0.0	0:00.43	rcuop/0
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuos/0

## Dòng thứ hai trong kết xuất lệnh top

Dòng thứ hai của đầu ra hàng đầu hiển thị tổng số tiến trình, số lượng quá trình chạy, ngủ, dừng và zombie. So sánh số lượng các tiến trình đang chạy với trung bình tải giúp xác định xem hệ thống đã đạt đến công suất hay có lẽ một người dùng cụ thể đang chạy quá nhiều tiến trình. Các tiến trình dừng nên được kiểm tra để xem mọi thứ có chạy đúng không.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
10831	root	20	0	113636	2512	2256	R	73.1	0.0	0:02.20	awk
10830	root	20	0	141180	23968	3232	S	14.6	0.1	0:00.44	objdump
2442	coop	20	0	2080636	280092	86628	S	12.6	1.7	2:29.41	gnome-shell
10832	root	20	0	4196	676	596	R	5.0	0.0	0:00.15	test_get_len
1449	root	20	0	467744	44588	29032	S	2.0	0.3	1:15.14	Xorg
3819	coop	20	0	539608	160724	63500	S	2.0	1.0	3:06.89	skype-bin
7492	coop	20	0	1184072	228944	84920	S	1.3	1.4	0:17.70	thunderbird
4428	coop	20	0	644404	55776	49988	S	1.0	0.3	0:00.23	gnome-screensho
4058	coop	20	0	1492732	327060	108116	S	0.7	2.0	2:04.10	chrome
28	root	rt	0	0	0	0	S	0.3	0.0	0:00.03	migration/2
68	root	20	0	0	0	0	S	0.3	0.0	0:00.49	rcuop/6
2372	coop	20	0	1492444	63244	28244	S	0.3	0.4	0:02.38	gnome-settings-
2758	coop	20	0	659456	65372	51556	S	0.3	0.4	0:03.99	gnome-terminal-
1	root	20	0	125680	5544	3676	S	0.0	0.0	0:01.55	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
7	root	20	0	0	0	0	S	0.0	0.0	0:02.61	rcu_preempt
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_sched
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
10	root	20	0	0	0	0	S	0.0	0.0	0:00.43	rcuop/0
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuos/0

### Dòng thứ ba trong kết xuất lệnh top

Dòng thứ ba của đầu ra trên cùng cho biết thời gian CPU được phân chia giữa người dùng (chúng tôi) và hạt nhân (sy) bằng cách hiển thị phần trăm thời gian CPU được sử dụng cho mỗi lần.

Tỷ lệ phần trăm công việc của người dùng đang chạy ở mức ưu tiên thấp hơn (tính độc đáo - ni) sau đó được liệt kê. Chế độ không tải (id) nên thấp nếu trung bình tải cao và ngược lại. Tỷ lệ phần trăm công việc đang chờ (wa) cho I / O được liệt kê. Ngắt bao gồm tỷ lệ phần cứng (hi) so với ngắt phần mềm (si). Steal time (st) thường được sử dụng với các máy ảo, có một số thời gian CPU nhàn rỗi của nó dành cho các mục đích sử dụng khác.

```

File Edit View Search Terminal Help
top - 10:52:22 up 3:17, 5 users, load average: 1.11, 0.43, 0.20
Tasks: 333 total, 3 running, 330 sleeping, 0 stopped, 0 zombie
%Cpu(s): 24.6 us, 5.5 sy, 0.0 ni, 65.8 id, 4.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 16283344 total, 9176888 free, 1713680 used, 5392776 buff/cache
KiB Swap: 8290300 total, 8290300 free, 0 used. 13387900 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
10831 root      20   0 113636  2512  2256 R 73.1  0.0  0:02.20 awk
10830 root      20   0 141180 23968  3232 S 14.6  0.1  0:00.44 objdump
2442 coop      20   0 2080636 280092 86628 S 12.6  1.7  2:29.41 gnome-shell
10832 root      20   0  4196   676   596 R  5.0  0.0  0:00.15 test_get_len
1449 root      20   0 467744 44588 29032 S  2.0  0.3  1:15.14 Xorg
3819 coop      20   0 539608 160724 63500 S  2.0  1.0  3:06.89 skype-bin
7492 coop      20   0 1184072 228944 84920 S  1.3  1.4  0:17.70 thunderbird
4428 coop      20   0 644404 55776 49988 S  1.0  0.3  0:00.23 gnome-screensho
4058 coop      20   0 1492732 327060 108116 S  0.7  2.0  2:04.10 chrome
 28 root       rt  0   0   0   0 S  0.3  0.0  0:00.03 migration/2
 68 root      20   0   0   0   0 S  0.3  0.0  0:00.49 rcuop/6
2372 coop      20   0 1492444 63244 28244 S  0.3  0.4  0:02.38 gnome-settings-
2758 coop      20   0 659456 65372 51556 S  0.3  0.4  0:03.99 gnome-terminal-
 1 root      20   0 125680 5544 3676 S  0.0  0.0  0:01.55 systemd
 2 root      20   0   0   0   0 S  0.0  0.0  0:00.01 kthreadd
 3 root      20   0   0   0   0 S  0.0  0.0  0:00.00 ksoftirqd/0
 5 root      0 -20   0   0   0 S  0.0  0.0  0:00.00 kworker/0:0H
 7 root      20   0   0   0   0 S  0.0  0.0  0:02.61 rCU_prempt
 8 root      20   0   0   0   0 S  0.0  0.0  0:00.00 rCU_sched
 9 root      20   0   0   0   0 S  0.0  0.0  0:00.00 rCU_bh
10 root      20   0   0   0   0 S  0.0  0.0  0:00.43 rcuop/0
11 root      20   0   0   0   0 S  0.0  0.0  0:00.00 rcuos/0

```

## Dòng thứ tư và năm trong kết xuất lệnh top

Dòng thứ tư và thứ năm của đầu ra trên cùng biểu thị mức sử dụng bộ nhớ, được chia thành hai loại:

Bộ nhớ vật lý (RAM) - hiển thị trên dòng 4.

Hoán đổi không gian - hiển thị trên dòng 5.

Cả hai loại đều hiển thị tổng bộ nhớ, bộ nhớ đã sử dụng và không gian trống.

Bạn cần theo dõi việc sử dụng bộ nhớ rất cẩn thận để đảm bảo hiệu năng hệ thống tốt. Khi bộ nhớ vật lý cạn kiệt, hệ thống bắt đầu sử dụng không gian trống (không gian lưu trữ tạm thời trên ổ cứng) làm nhóm bộ nhớ mở rộng và do việc truy cập đĩa chậm hơn nhiều so với truy cập bộ nhớ, điều này sẽ ảnh hưởng tiêu cực đến hiệu suất hệ thống.

Nếu hệ thống bắt đầu sử dụng trao đổi thường xuyên, bạn có thể thêm nhiều không gian trống. Tuy nhiên, thêm bộ nhớ vật lý cũng nên được xem xét.

File Edit View Search Terminal Help											
top - 10:52:22 up 3:17, 5 users, load average: 1.11, 0.43, 0.20											
Tasks: 333 total, 3 running, 330 sleeping, 0 stopped, 0 zombie											
%Cpu(s): 24.6 us, 5.5 sy, 0.0 ni, 65.8 id, 4.0 wa, 0.0 hi, 0.0 si, 0.0 st											
KiB Mem : 16283344 total, 9176888 free, 1713680 used, 5392776 buff/cache											
KiB Swap: 8290300 total, 8290300 free, 0 used. 13387900 avail Mem											
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
10831	root	20	0	113636	2512	2256	R	73.1	0.0	0:02.20	awk
10830	root	20	0	141180	23968	3232	S	14.6	0.1	0:00.44	objdump
2442	coop	20	0	2080636	280092	86628	S	12.6	1.7	2:29.41	gnome-shell
10832	root	20	0	4196	676	596	R	5.0	0.0	0:00.15	test_get_len
1449	root	20	0	467744	44588	29032	S	2.0	0.3	1:15.14	Xorg
3819	coop	20	0	539608	160724	63500	S	2.0	1.0	3:06.89	skype-bin
7492	coop	20	0	1184072	228944	84920	S	1.3	1.4	0:17.70	thunderbird
4428	coop	20	0	644404	55776	49988	S	1.0	0.3	0:00.23	gnome-screensho
4058	coop	20	0	1492732	327060	108116	S	0.7	2.0	2:04.10	chrome
28	root	rt	0	0	0	0	S	0.3	0.0	0:00.03	migration/2
68	root	20	0	0	0	0	S	0.3	0.0	0:00.49	rcuop/6
2372	coop	20	0	1492444	63244	28244	S	0.3	0.4	0:02.38	gnome-settings-
2758	coop	20	0	659456	65372	51556	S	0.3	0.4	0:03.99	gnome-terminal-
1	root	20	0	125680	5544	3676	S	0.0	0.0	0:01.55	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
7	root	20	0	0	0	0	S	0.0	0.0	0:02.61	rcu_premempt
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_sched
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
10	root	20	0	0	0	0	S	0.0	0.0	0:00.43	rcuop/0
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuos/0

## Danh sách các tiến trình trong kết xuất lệnh top

Mỗi dòng trong danh sách tiến trình của đầu ra hàng đầu hiển thị thông tin về một tiến trình. Theo mặc định, các tiến trình được sắp xếp theo mức sử dụng CPU cao nhất. Thông tin sau đây về mỗi tiến trình được hiển thị:

Số nhận dạng quá trình (PID)

Chủ sở hữu tiến trình (USER)

Ưu tiên (PR) và giá trị tốt đẹp (NI)

Áo (VIRT), vật lý (RES) và bộ nhớ dùng chung (SHR)

Tình trạng (S)

Tỷ lệ phần trăm của CPU (% CPU) và bộ nhớ (% MEM) được sử dụng

Thời gian thực hiện (TIME +)

Lệnh (LỆNH).

File Edit View Search Terminal Help												
top - 10:52:22 up 3:17, 5 users, load average: 1.11, 0.43, 0.20												
Tasks: 333 total, 3 running, 330 sleeping, 0 stopped, 0 zombie												
%Cpu(s): 24.6 us, 5.5 sy, 0.0 ni, 65.8 id, 4.0 wa, 0.0 hi, 0.0 si, 0.0 st												
KiB Mem : 16283344 total, 9176888 free, 1713680 used, 5392776 buff/cache												
KiB Swap: 8290300 total, 8290300 free, 0 used. 13387900 avail Mem												
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND	
10831	root	20	0	113636	2512	2256	R	73.1	0.0	0:02.20	awk	
10830	root	20	0	141180	23968	3232	S	14.6	0.1	0:00.44	objdump	
2442	coop	20	0	2080636	280092	86628	S	12.6	1.7	2:29.41	gnome-shell	
10832	root	20	0	4196	676	596	R	5.0	0.0	0:00.15	test_get_len	
1449	root	20	0	467744	44588	29032	S	2.0	0.3	1:15.14	Xorg	
3819	coop	20	0	539608	160724	63500	S	2.0	1.0	3:06.89	skype-bin	
7492	coop	20	0	1184072	228944	84920	S	1.3	1.4	0:17.70	thunderbird	
4428	coop	20	0	644404	55776	49988	S	1.0	0.3	0:00.23	gnome-screensho	
4058	coop	20	0	1492732	327060	108116	S	0.7	2.0	2:04.10	chrome	
28	root	rt	0	0	0	0	S	0.3	0.0	0:00.03	migration/2	
68	root	20	0	0	0	0	S	0.3	0.0	0:00.49	rcuop/6	
2372	coop	20	0	1492444	63244	28244	S	0.3	0.4	0:02.38	gnome-settings-	
2758	coop	20	0	659456	65372	51556	S	0.3	0.4	0:03.99	gnome-terminal-	
1	root	20	0	125680	5544	3676	S	0.0	0.0	0:01.55	systemd	
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd	
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0	
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H	
7	root	20	0	0	0	0	S	0.0	0.0	0:02.61	rcu_preempt	
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_sched	
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh	
10	root	20	0	0	0	0	S	0.0	0.0	0:00.43	rcuop/0	
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuos/0	

## Các phím tương tác với lệnh top

Bên cạnh thông tin báo cáo, top có thể được sử dụng tương tác để theo dõi và kiểm soát các tiến trình. Trong khi top đang chạy trong cửa sổ terminal, bạn có thể nhập các lệnh đơn để thay đổi hành vi của nó. Ví dụ: bạn có thể xem các tiến trình được xếp hạng hàng đầu dựa trên việc sử dụng CPU hoặc bộ nhớ. Nếu cần, bạn có thể thay đổi mức độ ưu tiên của các tiến trình đang chạy hoặc bạn có thể dừng / hủy một tiến trình.

Bảng liệt kê những gì xảy ra khi nhấn các phím khác nhau khi chạy trên cùng:

- t Hiển thị hoặc ẩn thông tin tóm tắt (hàng 2 và 3)
- m Hiển thị hoặc ẩn thông tin bộ nhớ (hàng 4 và 5)
- A Sắp xếp danh sách tiến trình theo người tiêu dùng tài nguyên hàng đầu
- r Renice (thay đổi mức độ ưu tiên của) một tiến trình cụ thể
- k Giết một tiến trình cụ thể
- f Vào màn hình cấu hình trên cùng
- o Tương tác chọn một thứ tự sắp xếp mới trong danh sách tiến trình

## 4.13 Lập kế hoạch cho các tiến trình trong tương lai

Giả sử bạn cần thực hiện một nhiệm vụ vào một ngày cụ thể đôi khi trong tương lai. Tuy nhiên, bạn biết bạn sẽ rời xa máy vào ngày hôm đó. Làm thế nào bạn sẽ thực hiện nhiệm vụ? Bạn có thể sử dụng chương trình tiện ích để thực thi bất kỳ lệnh không tương tác nào tại một thời điểm được chỉ định, như được minh họa trong sơ đồ:

```
[test3@CentOS ~]$ at now + 2 days
at> cat file1.txt
at> <EOT>
job 3 at 2014-07-12 11:58
[test3@CentOS ~]$ █
```

The terminal session shows the use of the 'at' command to schedule a task for two days from now. The command 'at now + 2 days' specifies the time when the task needs to be performed. The command 'cat file1.txt' specifies the task to be performed. A carriage return (<EOT>) is used to end the command. The output shows the job number (job 3) and the scheduled time (2014-07-12 11:58). A note indicates to press CTRL-D here.

Specifies when the task needs to be performed after two days from now

This command specifies the task to be performed.

Press CTRL-D here.

## Lệnh cron

**cron** là một chương trình tiện ích lập kế hoạch dựa trên thời gian. Nó có thể khởi chạy các công việc nền thường lệ vào thời gian và / hoặc ngày cụ thể trên cơ sở đang diễn ra. cron được điều khiển bởi một tệp cấu hình có tên / etc / crontab (bảng cron), chứa các lệnh shell khác nhau cần được chạy vào thời gian được lên lịch chính xác. Có cả tệp crontab trên toàn hệ thống và các tệp dựa trên người dùng cá nhân. Mỗi dòng của tệp crontab đại diện cho một công việc và bao gồm một biểu thức được gọi là CRON, theo sau là một lệnh shell để thực thi.

Lệnh crontab -e sẽ mở trình soạn thảo crontab để chỉnh sửa các công việc hiện có hoặc để tạo các công việc mới. Mỗi dòng của tệp crontab sẽ chứa 6 trường:

Giá trị mô tả trường

MIN phút 0 đến 59

Giờ GIỜ 0 đến 23

Ngày DOM tháng 1-31

Trường tháng 1-12

Ngày DOW trong tuần 0-6 (0 = Chủ nhật)

Lệnh CMD Bất kỳ lệnh nào sẽ được thực thi

Ví dụ:

Mục nhập \* \* \* \* \* /usr/local/bin/execute/this/script.sh sẽ sắp xếp công việc để thực thi 'script.sh' mỗi phút mỗi giờ mỗi ngày trong tháng và mỗi tháng và mỗi ngày trong tuần.

Mục nhập 30 08 10 06 \* / home / sysadmin / full-backup sẽ lên lịch sao lưu toàn bộ vào 8h30 sáng, ngày 10 tháng 6, bất kể ngày nào trong tuần.

## Lệnh sleep

Đôi khi, một lệnh hoặc công việc phải bị trì hoãn hoặc bị đình chỉ. Giả sử, ví dụ, một ứng dụng đã đọc và xử lý nội dung của tệp dữ liệu và sau đó cần lưu báo cáo trên hệ thống sao lưu. Nếu hệ thống sao lưu hiện đang bận hoặc không có sẵn, ứng dụng có thể được thực hiện sleep để ngủ (chờ) cho đến khi có thể hoàn thành công việc. Một sự chậm trễ như vậy có thể là gắn thiết bị sao lưu và chuẩn bị cho nó viết sleep tạm dừng thực thi trong ít nhất khoảng thời gian được chỉ định, có thể được cung cấp dưới dạng số giây (mặc định), phút, giờ hoặc ngày. Sau khi thời gian đó trôi qua (hoặc tín hiệu ngắt đã được nhận), việc thực thi sẽ tiếp tục.

Cú pháp là:

sleep SỐ [SUFFIX] ...

trong đó SUFFIX có thể là:

s trong vài giây (mặc định)

m trong vài phút

h hàng giờ

d trong nhiều ngày.

sleep và cron khá khác nhau; sleep trì hoãn thực hiện trong một khoảng thời gian cụ thể, trong khi cron bắt đầu thực hiện sau đó.

```
student@openSUSE:/tmp
File Edit View Search Terminal Help
student@openSUSE:/tmp> cat testsleep.sh
#!/bin/bash
if [ "$1" == "" ] ; then TIME=10 ; else TIME=$1 ; fi
echo -e "Going to sleep for $TIME seconds\n"
sleep $TIME
echo -e "I awoke after $TIME seconds\n"
student@openSUSE:/tmp> ./testsleep.sh
Going to sleep for 10 seconds

I awoke after 10 seconds

student@openSUSE:/tmp> ./testsleep.sh 3
Going to sleep for 3 seconds

I awoke after 3 seconds

student@openSUSE:/tmp>
```

#### 4.14 Câu hỏi ôn tập Chương 4

##### 1. *Lấy thông tin uptime và load averages*

Xem hệ thống của bạn đã hoạt động được bao lâu.

Hiển thị trung bình tải của nó.

##### 2. *Các công việc chạy nền và mặt tiền*

Khởi chạy một chương trình đồ họa từ cửa sổ đầu cuối, để người ta không còn có thể gõ vào cửa sổ. gedit là một lựa chọn dễ dàng, nhưng bạn có thể thay thế bất kỳ chương trình nào khác thực hiện điều này.

Tệp giải pháp chứa tiến trình từng bước để đặt công việc ở chế độ nền, đưa chúng trở lại nền trước, v.v. Vui lòng lặp lại các bước, thay thế chương trình bạn đang sử dụng nếu không phải là gedit.

## Sử dụng lệnh at cho xử lý tiến trình lô trong tương lai

Lịch trình một nhiệm vụ rất đơn giản để chạy vào một thời điểm trong tương lai kể từ bây giờ. Điều này có thể đơn giản như chạy ls hoặc ngày và lưu đầu ra. Bạn có thể sử dụng thời gian ngắn nhất là một phút trong tương lai.

Lưu ý rằng lệnh sẽ chạy trong thư mục mà bạn lén lịch với.

Làm cái này:

Từ một kịch bản bash ngắn.

Tương tác.

+ Sắp lịch tiến trình lặp với lệnh cron

Thiết lập một công việc định kỳ để thực hiện một số nhiệm vụ đơn giản mỗi ngày vào lúc 10 giờ sáng.

1. Mô tả tiến trình là gì và phân biệt giữa các loại tiến trình
2. Liệt kê các thuộc tính của tiến trình
3. Mô tả quản lý các tiến trình bằng cách lệnh **ps**, **top**, **htop**
4. Mô tả về tải trung bình tải và các dữ liệu liên quan tới tiến trình
5. Mô tả thao tác quản lý tiến trình bằng cách đưa vào chạy nền (background), chuyển lên mặt tiến (foreground)
6. Mô tả các lệnh at, cron và sleep để lén lịch chạy tiến trình trong tương lai hoặc tạm dừng

# Chương 5. Quản lý hệ thống tập tin và soạn thảo tập tin trên Linux

## 5.1 Giới thiệu hệ thống tập tin Linux

Trong Linux (và tất cả các hệ điều hành giống như UNIX), người ta thường nói rằng Mọi thứ đều là một tập tin, hoặc ít nhất nó được xử lý như vậy. Điều này có nghĩa là cho dù bạn đang xử lý các tệp dữ liệu và tài liệu thông thường hoặc với các thiết bị như card âm thanh và máy in, bạn tương tác với chúng thông qua cùng loại hoạt động Nhập / Xuất (I / O). Điều này giúp đơn giản hóa mọi thứ: bạn mở một tập tin trên máy tính và các hoạt động bình thường như đọc tệp và viết lên nó (đó là một lý do tại sao các trình soạn thảo văn bản, mà bạn sẽ tìm hiểu trong phần sắp tới, rất quan trọng).

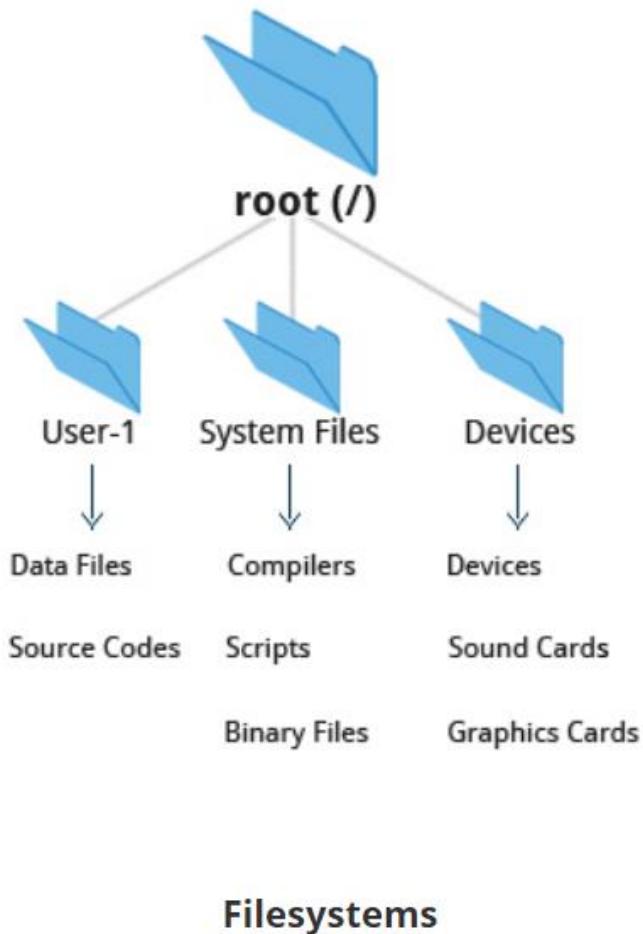
Trên nhiều hệ thống (bao gồm cả Linux), hệ thống tập tin được cấu trúc như một cái cây. Cây thường được miêu tả là đảo ngược và bắt đầu từ thư mục gốc thường được gọi là thư mục gốc, đánh dấu sự khởi đầu của hệ thống tập tin phân cấp và đôi khi còn được gọi là thân cây, hoặc đơn giản được ký hiệu là /. Thư mục gốc không giống với người dùng root. Hệ thống tệp phân cấp cũng chứa các thành phần khác trong đường dẫn (tên thư mục), được phân tách bằng dấu gạch chéo (/), như trong /usr/bin/emacs, trong đó phần tử cuối cùng là tên tệp thực tế.

Trong phần này, bạn sẽ tìm hiểu về một số khái niệm cơ bản, bao gồm phân cấp hệ thống tệp, cũng như về phân vùng đĩa.

### Các đặc trưng của hệ thống tập tin

Linux hỗ trợ một số loại hệ thống tệp gốc, được tạo bởi các nhà phát triển Linux, chẳng hạn như: ext3, ext4, squashfs, btrfs...

Nó cũng cung cấp việc triển khai các hệ thống tập tin được sử dụng trên các hệ điều hành ngoài hành tinh khác, chẳng hạn như các hệ thống từ Windows (ntfs, vfat), SGI (xfs), IBM (jfs), MacOS (hfs, hfs +).



### Filesystems

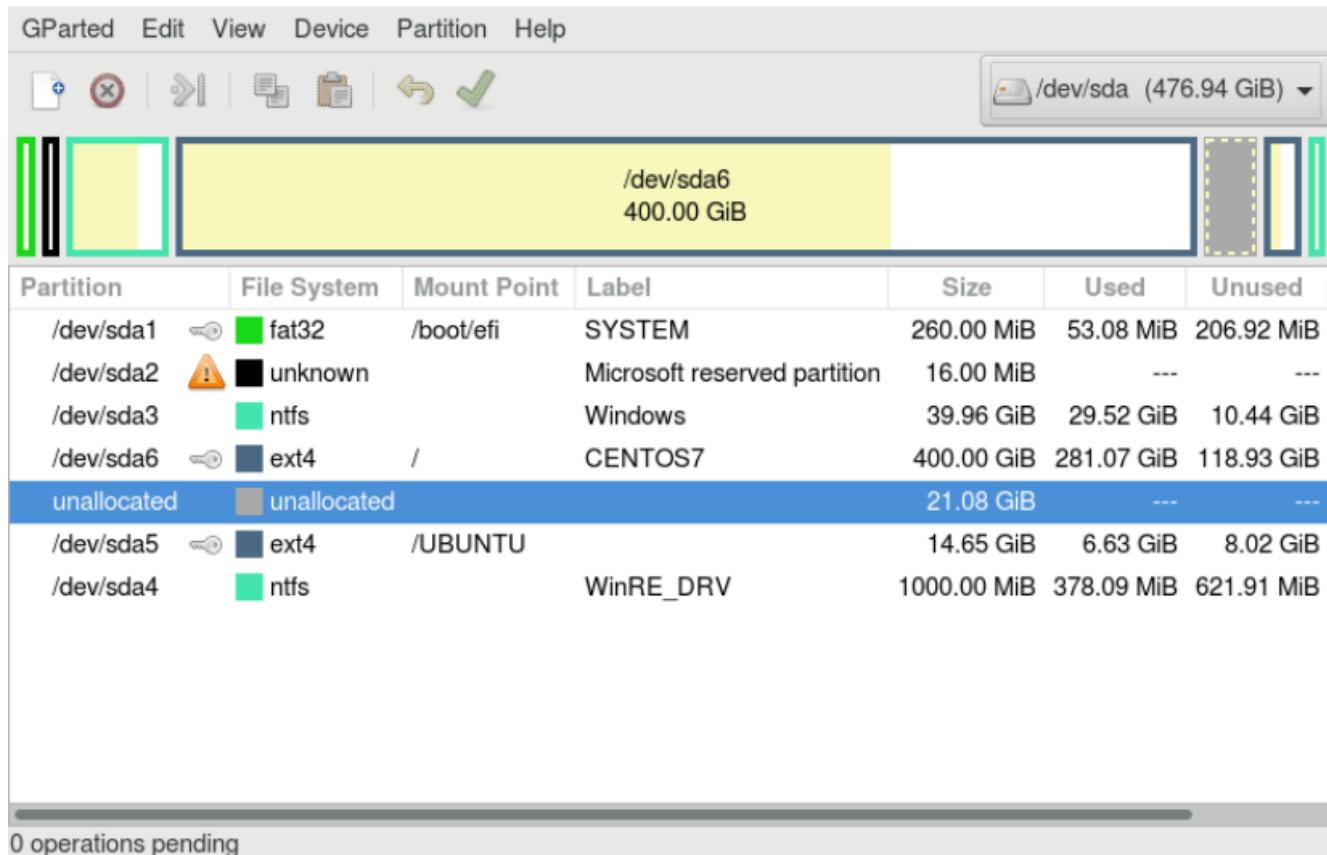
Nhiều hệ thống tập tin cũ hơn, chẳng hạn như FAT, cũng được hỗ trợ.

Thông thường, có nhiều loại hệ thống tệp được sử dụng trên máy, dựa trên các cân nhắc như kích thước tệp, tần suất chúng được sửa đổi, loại phần cứng nào chúng sử dụng và loại tốc độ truy cập nào là cần thiết, v.v. Các loại hệ thống tập tin tiên tiến nhất được sử dụng phổ biến là các loại nhật ký: ext4, xfs, btrfs và jfs. Chúng có nhiều tính năng hiện đại và hiệu suất cao, và rất khó để vô tình bị hỏng.

## 5.2 Phân vùng Linux

Như chúng ta đã thảo luận trước đó, mỗi hệ thống tệp trên hệ thống Linux chiếm một phân vùng đĩa cứng. Các phân vùng giúp tổ chức nội dung của các đĩa theo loại và việc sử dụng dữ liệu chung. Ví dụ, các chương trình quan trọng cần thiết để chạy hệ thống thường được giữ trên một phân vùng riêng (được gọi là root hoặc '/') so với chương trình chứa các tệp thuộc sở hữu của người dùng thông thường của hệ thống đó ('/home'). Ngoài ra, các tệp tạm thời được tạo và hủy trong quá trình hoạt động bình thường của Linux có thể nằm trên các phân vùng chuyên dụng. Một lợi thế của loại cách ly này theo loại và tính biến đổi là khi tất cả không gian có sẵn trên một phân vùng cụ thể bị cạn kiệt, hệ thống vẫn có thể hoạt động bình thường.

Các hình ảnh cho thấy việc sử dụng một tiện ích có tên **gparted**, hiển thị bố cục phân vùng trên một hệ thống có ba hệ điều hành trên đó: Windows 10, RHEL 7 và Ubuntu 18.04.



### Linux Partitions: gparted

#### 5.3 Các điểm kết nối của hệ thống tập tin

Trước khi bạn có thể bắt đầu sử dụng một hệ thống tập tin, bạn cần gắn nó vào cây hệ thống tập tin tại một điểm gắn kết. Đây chỉ đơn giản là một thư mục (có thể có hoặc không trống) trong đó hệ thống tập tin sẽ được đính kèm (gắn kết). Đôi khi, bạn có thể cần tạo thư mục nếu nó chưa tồn tại.

Cảnh báo: Nếu bạn gắn hệ thống tệp vào thư mục không trống, nội dung cũ của thư mục đó sẽ được che đậy và không thể truy cập cho đến khi hệ thống tệp bị ngắt. Vì vậy, các điểm gắn kết thường là các thư mục trống.

## Kết nối và gỡ bỏ kết nối

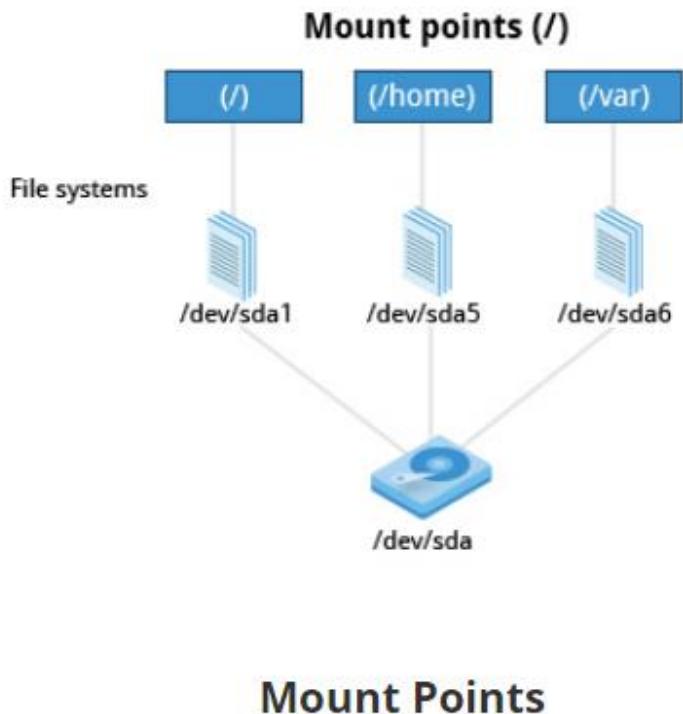
Lệnh mount được sử dụng để đính kèm một hệ thống tệp (có thể là cục bộ vào máy tính hoặc, như chúng ta sẽ thảo luận, trên mạng) ở đâu đó trong cây hệ thống tệp. Các đối số cơ bản là nút thiết bị và điểm gắn kết. Ví dụ,

```
$ sudo mount /dev/sda5 /home
```

sẽ đính kèm hệ thống tập tin có trong phân vùng đĩa được liên kết với nút thiết bị /dev/sda5, vào cây hệ thống tập tin tại điểm gắn kết /home. Có nhiều cách khác để chỉ định phân vùng ngoài nút thiết bị, chẳng hạn như sử dụng nhãn đĩa hoặc UUID.

Để ngắt kết nối phân vùng, lệnh sẽ là:

```
$ sudo umount /home
```



## Mount Points

```

student@debian:~$ mount | head -10
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,relatime,size=10240k,nr_inodes=470703,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,relatime,size=757028k,mode=755)
/dev/sdal on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
student@debian:~$ df -hT
Filesystem      Type     Size   Used  Avail Use% Mounted on
/dev/sdal        ext4    19G   5.3G  13G  31% /
udev            devtmpfs 10M     0    10M  0% /dev
tmpfs           tmpfs   740M   9.0M  731M  2% /run
tmpfs           tmpfs   1.9G   88K   1.9G  1% /dev/shm
tmpfs           tmpfs   5.0M   4.0K   5.0M  1% /run/lock
tmpfs           tmpfs   1.9G     0   1.9G  0% /sys/fs/cgroup
tmpfs           tmpfs   370M   12K   370M  1% /run/user/1000
/dev/sr0          iso9660 246M  246M     0 100% /media/cdrom0
student@debian:~$ 

```

## Mounting and Unmounting

Lưu ý lệnh là umount, không unmount! Chỉ người dùng root (đăng nhập với quyền root hoặc sử dụng sudo) mới có đặc quyền để chạy các lệnh này, trừ khi hệ thống đã được cấu hình khác.

Nếu bạn muốn nó tự động khả dụng mỗi khi hệ thống khởi động, bạn cần chỉnh sửa / etc / fstab cho phù hợp (tên viết tắt của bảng hệ thống tập tin). Nhìn vào tập tin này sẽ cho bạn thấy cấu hình của tất cả các hệ thống tập tin được cấu hình sẵn. man fstab sẽ hiển thị cách sử dụng tệp này và cách định cấu hình tệp.

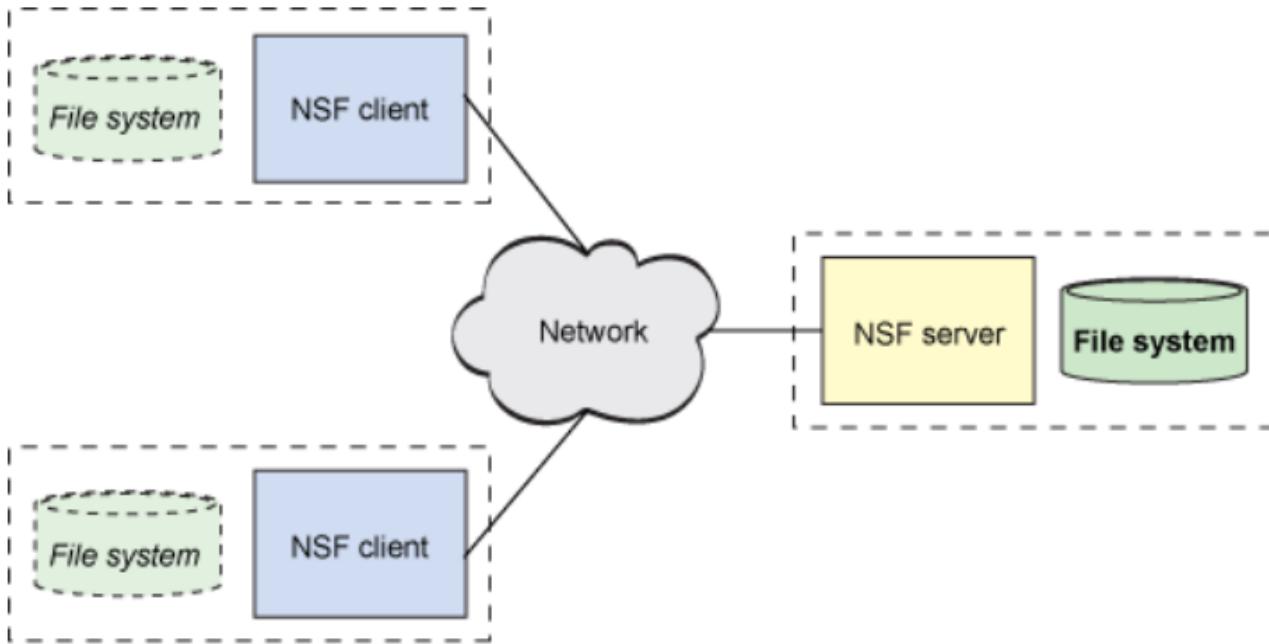
Gõ mount mà không có bất kỳ đối số nào sẽ hiển thị tất cả các hệ thống tập tin được gắn hiện tại.

Lệnh df -Th (không có đĩa) sẽ hiển thị thông tin về các hệ thống tệp được gắn kết, bao gồm loại hệ thống tệp và thông kê sử dụng về không gian hiện được sử dụng và có sẵn.

## 5.4 NFS - Hệ thống tập tin mạng

Thường là cần thiết để chia sẻ dữ liệu trên các hệ thống vật lý có thể ở cùng một vị trí hoặc bất kỳ nơi nào có thể truy cập Internet. Một hệ thống tập tin mạng (đôi khi được gọi là phân tán) có thể có tất cả dữ liệu của nó trên một máy hoặc có trải rộng trên nhiều nút mạng. Một loạt các hệ thống tập tin khác nhau có thể được sử dụng cục bộ trên các máy

riêng lẻ; một hệ thống tập tin mạng có thể được coi là một nhóm các hệ thống tập tin cấp thấp hơn với các loại khác nhau.



### The Client-Server Architecture of NFS

Nhiều quản trị viên hệ thống gắn các thư mục nhà của người dùng từ xa trên một máy chủ để cấp cho họ quyền truy cập vào cùng một tệp và tệp cấu hình trên nhiều hệ thống máy khách. Điều này cho phép người dùng đăng nhập vào các máy tính khác nhau, nhưng vẫn có quyền truy cập vào cùng các tệp và tài nguyên.

Hệ thống tập tin phổ biến nhất như vậy được đặt tên đơn giản là NFS (Hệ thống tập tin mạng). Nó có một lịch sử rất dài và được phát triển đầu tiên bởi Sun Microsystems. Một triển khai phổ biến khác là CIFS (còn được gọi là SAMBA), có gốc từ Microsoft.

#### NFS trên máy chủ

Bây giờ chúng ta sẽ xem chi tiết về cách sử dụng NFS trên máy chủ.

Trên máy chủ, NFS sử dụng trình deamons (tiến trình dịch vụ và kết nối mạng tích hợp trong Linux) và các máy chủ hệ thống khác được khởi động tại dòng lệnh bằng cách nhập:

```
$ sudo systemctl start nfs
```

Tệp văn bản /etc/export chứa các thư mục và quyền mà chủ nhà sẵn sàng chia sẻ với các hệ thống khác qua NFS. Một mục rất đơn giản trong tệp này có thể trông như sau:

```
/project * .example.com (rw)
```

```
File Edit View Search Terminal Help
c7:/tmp>cat /etc/exports
/usr/local      192.168.1.0/24(rw,no_root_squash,sync)
/tmp           192.168.1.0/24(rw,no_root_squash)
/ISO_IMAGES    192.168.1.0/24(rw,no_root_squash)
/DEAD          192.168.1.0/24(rw,no_root_squash)
/VIRTUAL       192.168.1.0/24(rw,no_root_squash)

/var/ftp/pub   *(ro,insecure,all_squash)
c7:/tmp>sudo exportfs
/usr/local      192.168.1.0/24
/tmp           192.168.1.0/24
/ISO_IMAGES    192.168.1.0/24
/DEAD          192.168.1.0/24
/VIRTUAL       192.168.1.0/24
/usr/local/FTP <world>
c7:/tmp>
```

## NFS on the Server

Mục này cho phép thư mục /projects được gắn kết bằng NFS với quyền đọc và ghi (rw) và được chia sẻ với các máy chủ khác trong miền example.com. Như chúng ta sẽ trình bày chi tiết trong chương tiếp theo, mọi tệp trong Linux đều có ba quyền có thể: đọc (r), write (w) và thực thi (x).

Sau khi sửa đổi tệp /etc/export, bạn có thể sử dụng lệnh exportfs -av để thông báo cho Linux về các thư mục mà bạn cho phép được gắn từ xa bằng NFS. Bạn cũng có thể khởi động lại NFS bằng sudo systemctl khởi động lại nfs, nhưng điều này nặng hơn, vì nó tạm dừng NFS trong một thời gian ngắn trước khi khởi động lại. Để đảm bảo dịch vụ NFS khởi động bất cứ khi nào hệ thống được khởi động, hãy phát sudo systemctl kích hoạt nfs.

### NFS trên máy khách

Trên máy khách, nếu muốn hệ thống tệp từ xa được gắn tự động khi khởi động hệ thống, tệp /etc/fstab được sửa đổi để thực hiện việc này.

Ví dụ: một mục trong tệp /etc/fstab của máy khách có thể trông giống như sau:

servername: /projects /mnt/nfs/projects nfs mặc định 0 0

Bạn cũng có thể gắn hệ thống tập tin từ xa mà không cần khởi động lại hoặc dưới dạng gắn kết một lần bằng cách trực tiếp sử dụng lệnh mount:

```
$ sudo mount tên máy chủ: / dự án / mnt / nfs / dự án
```

Hãy nhớ rằng, nếu / etc / fstab không được sửa đổi, mount từ xa này sẽ không xuất hiện vào lần tiếp theo hệ thống được khởi động lại. Hơn nữa, bạn có thể muốn sử dụng tùy chọn nofail trong fstab trong trường hợp máy chủ NFS không hoạt động khi khởi động.

```
root@ubuntu:/etc# cat fstab
# /etc/fstab: static file system information.

# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>

# / was on /dev/sdal during installation
UUID=a1982023-a965-43e9-98c6-d2bc4415aa6b / ext4 errors=remount-ro 0 1

# swap was on /dev/sda5 during installation
UUID=79106f18-9933-4561-b79c-624941696c0b none swap sw 0 0

/dev/fd0 /media/floppy0 auto rw,user,noauto,exec,utf8 0 0

192.168.1.200:/ISO_IMAGES /ISO_IMAGES nfs defaults 0 0
root@ubuntu:/etc# df -hT
Filesystem Type Size Used Avail Use% Mounted on
udev devtmpfs 1.9G 0 1.9G 0% /dev
tmpfs tmpfs 381M 6.3M 375M 2% /run
/dev/sdal ext4 16G 6.6G 8.3G 45% /
tmpfs tmpfs 1.9G 212K 1.9G 1% /dev/shm
tmpfs tmpfs 5.0M 4.0K 5.0M 1% /run/lock
tmpfs tmpfs 1.9G 0 1.9G 0% /sys/fs/cgroup
cgmfs tmpfs 100K 0 100K 0% /run/cgmanager/fs
tmpfs tmpfs 381M 60K 381M 1% /run/user/1000
192.168.1.200:/ISO_IMAGES nfs4 109G 41G 63G 40% /ISO_IMAGES
root@ubuntu:/etc#
```

## NFS on the Client

5.5  
Thư  
mục  
nhà  
của  
người  
dùng

Trong phần này, bạn sẽ học cách xác định và phân biệt giữa các thư mục quan trọng nhất được tìm thấy trong Linux. Chúng tôi bắt đầu với không gian thư mục nhà của người dùng thông thường.

Mỗi người dùng có một thư mục nhà, thường được đặt dưới / nhà. Thư mục / root ("slash-root") trên các hệ thống Linux hiện đại không khác gì thư mục chính của người dùng root (hoặc siêu người dùng hoặc tài khoản quản trị viên hệ thống).

Trên các hệ thống nhiều người dùng, cơ sở hạ tầng thư mục / home thường được gắn dưới dạng một hệ thống tệp riêng biệt trên phân vùng riêng hoặc thậm chí được xuất (chia sẻ) từ xa trên mạng thông qua NFS.

Đôi khi, bạn có thể nhóm người dùng dựa trên bộ phận hoặc chức năng của họ. Sau đó, bạn có thể tạo các thư mục con trong thư mục / home cho mỗi nhóm này. Ví dụ, một trường học có thể tổ chức / nhà với những thứ như sau:

**/home/faculty/**

**/home/staff/**

**/home/students/**

## 5.6 Thư mục /bin và /sbin

Thư mục / bin chứa các nhị phân thực thi, các lệnh thiết yếu được sử dụng để khởi động hệ thống hoặc trong chế độ người dùng đơn và các lệnh thiết yếu được yêu cầu bởi tất cả người dùng hệ thống, chẳng hạn như cat, cp, ls, mv, ps và rm.

Tương tự, thư mục / sbin được dành cho các nhị phân thiết yếu liên quan đến quản trị hệ thống, chẳng hạn như fsck và tắt máy. Để xem danh sách các chương trình này, gõ:

**\$ ls /bin /sbin**

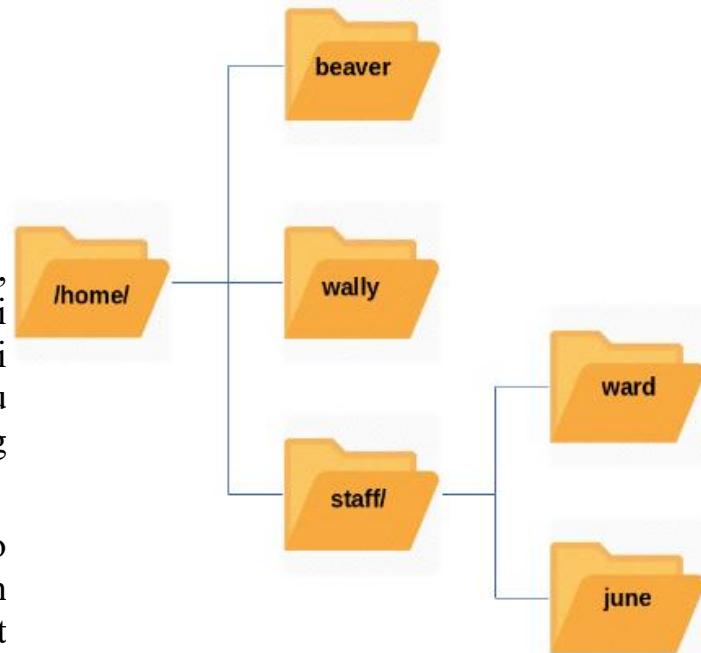
Các lệnh không cần thiết (về mặt lý thuyết) để hệ thống khởi động hoặc hoạt động ở chế độ người dùng đơn được đặt trong thư mục / usr / bin và / usr / sbin. Trong lịch sử, điều này đã được thực hiện để / usr có thể được gắn kết như một hệ thống tệp riêng biệt có thể được gắn vào giai đoạn khởi động hệ thống sau này hoặc thậm chí qua mạng. Tuy nhiên, ngày nay hầu hết đều thấy sự khác biệt này là lỗi thời. Trên thực tế, nhiều bản phân phối đã được phát hiện là không thể khởi động với sự phân tách này, vì phương thức này đã không được sử dụng hoặc thử nghiệm trong một thời gian dài.

Do đó, trên một số bản phân phối / usr / bin và / bin mới nhất của Linux thực sự chỉ được liên kết một cách tương trưng với nhau, như / usr / sbin và / sbin.

## 5.7 Hệ thống tập tin /proc

Một số hệ thống tệp nhất định, như hệ thống được gắn tại / proc, được gọi là hệ thống tệp giả vì chúng không có sự hiện diện vĩnh viễn ở bất kỳ đâu trên đĩa.

Hệ thống tập tin / proc chứa các tệp ảo (các tệp chỉ tồn tại trong bộ nhớ) cho phép xem dữ liệu kernel thay đổi liên tục. Hệ thống tập tin này chứa các tập tin và thư mục bắt chước cấu trúc kernel và thông tin cấu hình. Nó không chứa các tệp thực, nhưng thông tin hệ



**Home Directories**

thống thời gian chạy, ví dụ: bộ nhớ hệ thống, thiết bị được gắn, cấu hình phần cứng, v.v ... Một số tệp quan trọng trong /proc là:

/proc/cpuinfo  
/proc/interrupts  
/proc/meminfo  
/proc/mounts  
/proc/partitions  
/proc/version

/proc cũng có các thư mục con, bao gồm:

/proc/<ID-Process - #>  
/proc/sys

Ví dụ đầu tiên cho thấy có một thư mục cho mọi quá trình đang chạy trên hệ thống, chứa thông tin quan trọng về nó. Ví dụ thứ hai cho thấy một thư mục ảo chứa nhiều thông tin về toàn bộ hệ thống, đặc biệt là phần cứng và cấu hình của nó. Hệ thống tập tin /proc rất hữu ích vì thông tin mà nó báo cáo chỉ được thu thập khi cần thiết và không bao giờ cần lưu trữ trên đĩa.

## 5.8 Thư mục /dev

Thư mục / dev chứa các nút thiết bị, một loại tệp giả được sử dụng bởi hầu hết các thiết bị phần cứng và phần mềm, ngoại trừ các thiết bị mạng. Thư mục này là:

Làm trống trên phân vùng đĩa khi nó không được gắn kết

Chứa các mục được tạo bởi hệ thống udev, tạo và quản lý các nút thiết bị trên Linux, tạo chúng một cách linh hoạt khi tìm thấy thiết bị. Thư mục / dev chứa các mục như:

/dev/sda1 (phân vùng đầu tiên trên đĩa cứng đầu tiên)  
/dev/lp1 (máy in thứ hai)  
/dev/random (một nguồn số ngẫu nhiên)

## 5.9 Thư mục /var

Thư mục /var chứa các tệp được dự kiến sẽ thay đổi kích thước và nội dung khi hệ thống đang chạy (var là viết tắt của biến), chẳng hạn như các mục trong các thư mục sau:

Tệp nhật ký hệ thống: /var/log

Gói và tệp cơ sở dữ liệu: /var/lib

Hàng đợi in: /var/spool

Tệp tạm thời: /var/tmp.

Thư mục /var có thể được đặt trên hệ thống tệp riêng của mình để có thể điều chỉnh tăng trưởng của tệp và kích thước tệp không ảnh hưởng nghiêm trọng đến hệ thống. Các thư mục dịch vụ mạng như /var/ftp (dịch vụ FTP) và /var/www (dịch vụ web HTTP) cũng được tìm thấy trong /var.

## 5.10 Thư mục /etc

Thư mục /etc là trang chủ cho các tệp cấu hình hệ thống. Nó không chứa các chương trình nhị phân, mặc dù có một số tập lệnh thực thi. Ví dụ: /etc/resolv.conf cho hệ thống biết nơi sẽ truy cập mạng để lấy tên máy chủ để ánh xạ địa chỉ IP (DNS). Các tệp như passwd, bóng và nhóm để quản lý tài khoản người dùng được tìm thấy trong thư mục /etc. Mặc dù một số bản phân phối trong lịch sử có cơ sở hạ tầng rộng lớn theo /etc (ví dụ, Red Hat và SUSE đã sử dụng /etc/sysconfig), với sự ra đời của systemd có sự đồng đều hơn nhiều trong số các bản phân phối ngày nay.

Lưu ý rằng /etc dành cho các tệp cấu hình trên toàn hệ thống và chỉ siêu người dùng mới có thể sửa đổi các tệp ở đó. Các tập tin cấu hình dành riêng cho người dùng luôn được tìm thấy trong thư mục chính của họ.

## 5.11 Thư mục /boot

Thư mục /boot chứa một vài tệp cần thiết để khởi động hệ thống. Đối với mỗi kernel thay thế được cài đặt trên hệ thống, có bốn tệp:

*vmlinuz*

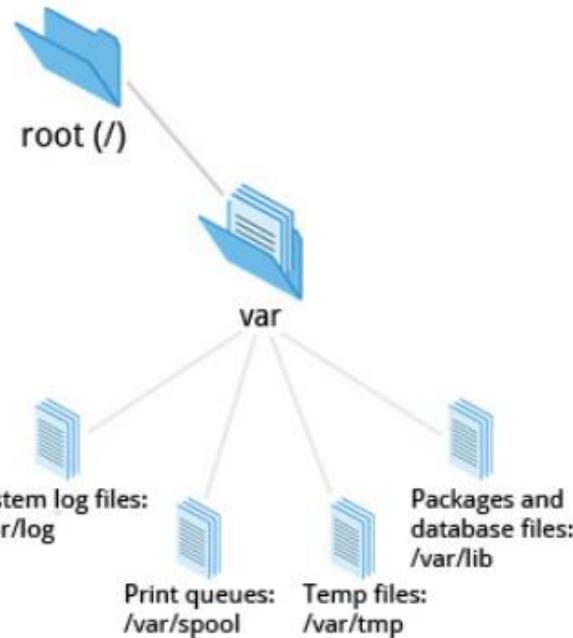
Hạt nhân Linux được nén, cần thiết để khởi động.

*initramfs*

Hệ thống tập tin ram ban đầu, được yêu cầu để khởi động, đôi khi được gọi là initrd, không phải initramfs.

*config*

Tệp cấu hình kernel, chỉ được sử dụng để gỡ lỗi và ghi số.



## The /var Directory

## *System.map*

Bảng ký hiệu hạt nhân, chỉ được sử dụng để gỡ lỗi.

Mỗi tệp này có một phiên bản kernel được gắn vào tên của nó.

Các tệp Grand Unified Bootloader (GRUB) như /boot/grub/grub.conf hoặc /boot/grub2/grub2.cfg cũng được tìm thấy trong thư mục / boot.

## **5.12 Thư mục /lib và /lib64**

/lib chứa các thư viện (mã chung được chia sẻ bởi các ứng dụng và cần thiết để chúng chạy) cho các chương trình thiết yếu trong /bin và /sbin. Những tên tệp thư viện hoặc bắt đầu bằng ld hoặc lib. Ví dụ: /lib/libncurses.so.5.9.

Hầu hết trong số này là những gì được gọi là thư viện được tải động (còn được gọi là thư viện chia sẻ hoặc Đối tượng chia sẻ (SO)). Trên một số bản phân phối Linux tồn tại thư mục /lib64 chứa các thư viện 64 bit, trong khi /lib chứa các phiên bản 32 bit.

Các mô-đun hạt nhân (mã hạt nhân, thường là trình điều khiển thiết bị, có thể được tải và tải mà không cần khởi động lại hệ thống) được đặt trong /lib/module/<kernel-version-number>.

## **5.13 Các thư mục ổ đĩa tháo lắp rời: /media, /run, /mnt**

Người ta thường sử dụng phương tiện di động, chẳng hạn như ổ USB, CD và DVD. Để làm cho tài liệu có thể truy cập thông qua hệ thống tập tin thông thường, nó phải được gắn tại một vị trí thuận tiện. Hầu hết các hệ thống Linux đều được định cấu hình để mọi phương tiện lưu động có thể tự động được gắn khi hệ thống nhận thấy có thứ gì đó đã được cắm.

Trong lịch sử, điều này được thực hiện trong thư mục /media, các bản phân phối Linux hiện đại đặt các điểm gắn kết này trong thư mục /run. Ví dụ: ổ đĩa USB có nhãn "myusbdrive" cho tên người dùng "student" sẽ được gắn tại /run/media/student/myusbdrive.

Thư mục /mnt đã được sử dụng từ những ngày đầu của UNIX để cài đặt tạm thời các hệ thống tập tin. Đây có thể là những thứ trên phương tiện lưu động, nhưng thường có thể là các hệ thống tệp mạng với NFS, thường không được gắn kết. Hoặc đây có thể là các phân vùng tạm thời, hay còn gọi là các hệ thống tệp loopback, là các tệp giả vờ là các phân vùng.

## **5.14 Các thư mục khác dưới thư mục gốc /**

Có một số thư mục bổ sung được tìm thấy trong thư mục gốc:

**/opt** Gói phần mềm ứng dụng tùy chọn

**/sys** Hệ thống tập tin giả ảo cung cấp thông tin về hệ thống và phần cứng, có thể được sử dụng để thay đổi các tham số hệ thống và cho mục đích gỡ lỗi

**/srv** Dữ liệu dành riêng cho trang web được cung cấp bởi hệ thống, ít khi sử dụng

**/tmp** Tệp tạm thời; trên một số bản phân phối bị xóa trong quá trình khởi động lại và / hoặc thực sự có thể là một ramdisk trong bộ nhớ

**/usr** Ứng dụng, tiện ích và dữ liệu đa người dùng

Thư mục con dưới /usr chứa các chương trình và tập lệnh không cần thiết về mặt lý thuyết (theo nghĩa là không cần thiết để khởi động hệ thống ban đầu) và có ít nhất các thư mục con sau:

**/usr/include** Các tệp headers được sử dụng để biên dịch các ứng dụng

**/usr/lib** Thư viện cho các chương trình trong /usr/bin và /usr/sbin

**/usr/lib64** Thư viện 64 bit cho các chương trình 64 bit trong /usr/bin và /usr/sbin

**/usr/sbin** Các nhị phân hệ thống không cần thiết, chẳng hạn như trình nền hệ thống

**/usr/share** Dữ liệu được chia sẻ được sử dụng bởi các ứng dụng, thường không phụ thuộc vào kiến trúc

**/usr/src** Mã nguồn, thường dành cho nhân Linux

**/usr/** Dữ liệu cục bộ và chương trình dành riêng cho máy cục bộ. Các thư mục con bao gồm **bin**, **sbin**, **lib**, **share**, **include**, v.v.

**/usr/bin** Đây là thư mục chính của các lệnh thực thi trên hệ thống

## 5.15 So sánh các tập tin với lệnh diff

Bây giờ bạn đã biết về hệ thống tập tin và cấu trúc của nó, hãy để học cách quản lý tập tin và thư mục.

diff được sử dụng để so sánh các tập tin và thư mục. Chương trình tiện ích thường được sử dụng này có nhiều tùy chọn hữu ích (xem: man diff) bao gồm:

**-c** Cung cấp danh sách các khác biệt bao gồm ba dòng ngữ cảnh trước và sau các dòng khác nhau về nội dung

**-r** Được sử dụng để so sánh đệ quy các thư mục con, cũng như thư mục hiện tại

**-i** Bỏ qua trường hợp chữ cái

**-w** Bỏ qua sự khác biệt trong không gian và tab (khoảng trắng)

**-q** Im lặng: chỉ báo cáo nếu các tệp khác nhau mà không liệt kê các khác biệt

Để so sánh hai tệp, tại dấu nhắc lệnh, nhập diff [tùy chọn] <filename1> <filename2>. diff có nghĩa là được sử dụng cho các tập tin văn bản; đối với các tệp nhị phân, người ta có thể sử dụng cmp.

Trong phần này, bạn sẽ tìm hiểu các phương pháp bổ sung để so sánh các tệp và cách áp dụng các bản vá cho các tệp.

### Sử dụng lệnh diff3 và patch

Bạn có thể so sánh ba tệp cùng một lúc bằng diff3, sử dụng một tệp làm cơ sở tham chiếu cho hai tệp kia. Ví dụ: giả sử cả bạn và đồng nghiệp đều thực hiện các sửa đổi cho cùng một tệp làm việc cùng một lúc một cách độc lập. diff3 có thể hiển thị sự khác biệt dựa trên tệp chung mà cả hai bạn bắt đầu. Cú pháp của diff3 như sau:

**\$ diff3 MY-FILE COMMON-FILE YOUR-FILE**

Nhiều sửa đổi cho mã nguồn và các tệp cấu hình được phân phối sử dụng các bản vá, được áp dụng, không ngạc nhiên, với chương trình vá. Một tệp vá chứa các deltas (thay đổi) cần thiết để cập nhật phiên bản cũ hơn của tệp lên tệp mới. Các tệp vá thực sự được tạo ra bằng cách chạy diff với các tùy chọn chính xác, như trong:

**\$ diff -Nur newfile newfile> patchfile**

Chỉ phân phối bản vá là ngắn gọn và hiệu quả hơn so với phân phối toàn bộ tệp. Ví dụ: nếu chỉ cần một dòng cần thay đổi trong tệp chứa 1000 dòng, tệp vá sẽ chỉ dài một vài dòng.

Để áp dụng một bản vá, bạn chỉ cần thực hiện một trong hai phương pháp dưới đây:

**\$ patch -p1 <patchfile**

**\$ patch patchfile gốc**

Việc sử dụng đầu tiên là phổ biến hơn, vì nó thường được sử dụng để áp dụng các thay đổi cho toàn bộ cây thư mục, thay vì chỉ một tệp, như trong ví dụ thứ hai. Để hiểu việc sử dụng tùy chọn -p1 và nhiều tùy chọn khác, hãy xem trang hướng dẫn để biết bản vá.

### 5.16 Sử dụng các tiện ích tập tin

Trong Linux, phần mở rộng của tệp thường không phân loại nó theo cách có thể trong các hệ điều hành khác. Người ta không thể cho rằng một tệp có tên file.txt là một tệp văn bản chứ không phải là một chương trình thực thi. Trong Linux, tên tệp thường có ý nghĩa hơn đối với người dùng hệ thống so với chính hệ thống. Trong thực tế, hầu hết các ứng dụng trực tiếp kiểm tra nội dung của tệp để xem loại đối tượng đó thay vì dựa vào tiện ích mở rộng. Điều này rất khác với cách Windows xử lý tên tệp, ví dụ tên tệp kết thúc bằng .exe, đại diện cho tệp nhị phân thực thi.

```

File Edit View Search Terminal Help
c7:/tmp/s_08>ls -F
EXAMPLES/          lab1_putget.c      lab1_read_write.c  Makefile
lab1_copytfrom.c   lab1_putget.ko    lab_char.h       Module.symvers
lab1_copytfrom.ko  lab1_read_write* lab_miscdev.h
c7:/tmp/s_08>file *
EXAMPLES:          directory
lab1_copytfrom.c:  C source, ASCII text
lab1_copytfrom.ko: ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV), BuildID
[sha1]=a976b92720e02eec958944ef818c32c1b393b4bb, not stripped
lab1_putget.c:     assembler source, ASCII text
lab1_putget.ko:    ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV), BuildID
[sha1]=f94634f6bc901031238758d139b6df0fea29ac97, not stripped
lab1_read_write:   ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically
linked (uses shared libs), for GNU/Linux 2.6.32, BuildID[sha1]=d08c490db17cd6
63132866bf72046758a571ffa5, not stripped
lab1_read_write.c: C source, ASCII text
lab_char.h:        assembler source, ASCII text
lab_miscdev.h:     C source, ASCII text
Makefile:          ASCII text
Module.symvers:   empty
c7:/tmp/s_08>

```

### Using the file Utility

Bản chất thực sự của một tập tin có thể được xác định bằng cách sử dụng tiện ích tập tin. Đối với các tên tập được đưa ra làm đối số, nó sẽ kiểm tra nội dung và các đặc điểm nhất định để xác định xem các tập có phải là văn bản thuần túy, thư viện dùng chung, chương trình thực thi, tập lệnh hoặc nội dung nào khác không.

### 5.17 Sao lưu dữ liệu trong Linux

Có nhiều cách bạn có thể sao lưu dữ liệu hoặc thậm chí toàn bộ hệ thống của bạn. Các cách cơ bản để làm như vậy bao gồm việc sử dụng sao chép đơn giản với cp và sử dụng rsync mạnh mẽ hơn.



Cả hai có thể được sử dụng để đồng bộ hóa toàn bộ cây thư mục. Tuy nhiên, rsync hiệu quả hơn, bởi vì nó kiểm tra xem tệp đã được sao chép đã tồn tại chưa. Nếu tệp tồn tại và không có thay đổi về kích thước hoặc thời gian sửa đổi, rsync sẽ tránh một bản sao không cần thiết và tiết kiệm thời gian. Hơn nữa, vì rsync chỉ sao chép các phần của tệp đã thực sự thay đổi, nên nó có thể rất nhanh.

Lệnh **cp** chỉ có thể sao chép tệp đến và từ đích trên máy cục bộ (trừ khi bạn sao chép

đến hoặc từ hệ thống tệp được gắn bằng NFS), nhưng rsync cũng có thể được sử dụng để sao chép tệp từ máy này sang máy khác. Các vị trí được chỉ định trong mục tiêu: biểu mẫu đường dẫn, trong đó mục tiêu có thể ở dạng ai đó @ host. Phần ai đó @ là tùy chọn và được sử dụng nếu người dùng từ xa khác với người dùng cục bộ.

Lệnh **rsync** rất hiệu quả khi sao chép đệ quy một cây thư mục này sang cây khác, bởi vì chỉ có sự khác biệt được truyền qua mạng. Người ta thường đồng bộ hóa cây thư mục đích với nguồn gốc, sử dụng tùy chọn -r để đệ quy xuống cây thư mục sao chép tất cả các tệp và thư mục bên dưới tệp được liệt kê dưới dạng nguồn.

## Sử dụng rsync

rsync là một tiện ích rất mạnh mẽ. Ví dụ, một cách rất hữu ích để sao lưu thư mục dự án có thể là sử dụng lệnh sau:

```
$ rsync -r project-X archive-machine: archives / project-X
```

Lưu ý rằng rsync có thể rất phá hoại! Lạm dụng ngẫu nhiên có thể gây hại rất nhiều cho dữ liệu và chương trình, bằng cách vô tình sao chép các thay đổi vào nơi họ không muốn. Hãy cẩn thận để xác định các tùy chọn và đường dẫn chính xác. Chúng tôi khuyên bạn trước tiên nên kiểm tra lệnh rsync của mình bằng tùy chọn -dry-run để đảm bảo rằng nó cung cấp kết quả mà bạn muốn.

Để sử dụng rsync tại dấu nhắc lệnh, hãy nhập rsync sourcefile Destinationfile, trong đó một trong hai tệp có thể nằm trên máy cục bộ hoặc trên máy được nối mạng; Nội dung của sourcefile sẽ được sao chép vào Destinationfile.

Một sự kết hợp tốt của các tùy chọn được hiển thị trong:

```
$ rsync --progress -avrxH --delete sourcedir Destdir
```

## 5.18 Nén dữ liệu tập tin

Dữ liệu tệp thường được nén để tiết kiệm dung lượng ổ đĩa và giảm thời gian truyền tệp qua mạng.

Linux sử dụng một số phương pháp để thực hiện việc nén này, bao gồm:

gzip Tiện ích nén Linux được sử dụng thường xuyên nhất

bzip2 Tạo các tệp nhỏ hơn đáng kể so với các tệp được tạo bởi gzip

xz Tiện ích nén hiệu quả nhất về không gian được sử dụng trong Linux

zip Thường được yêu cầu kiểm tra và giải nén tài liệu lưu trữ từ các hệ điều hành khác

Những kỹ thuật này khác nhau về hiệu quả của việc nén (tiết kiệm được bao nhiêu dung lượng) và thời gian để nén; nói chung, các kỹ thuật hiệu quả hơn mất nhiều thời gian hơn. Thời gian giải nén không thay đổi nhiều bằng các phương pháp khác nhau.

Ngoài ra, tiện ích tar thường được sử dụng để nhóm các tệp trong một kho lưu trữ và sau đó nén toàn bộ kho lưu trữ cùng một lúc.

gzip là tiện ích nén Linux được sử dụng thường xuyên nhất. Nó nén rất tốt và rất nhanh.

Bảng sau đây cung cấp một số ví dụ sử dụng:

gzip \* Nén tất cả các tệp trong thư mục hiện tại; mỗi tệp được nén và đổi tên bằng phần mở rộng .gz

gzip -r projectX Nén tất cả các tệp trong thư mục projectX, cùng với tất cả các tệp trong tất cả các thư mục trong projectX

gunzip foo Khử nén foo được tìm thấy trong tệp foo.gz. Dưới mui xe, lệnh gunzip thực sự giống như gzip

bzip2 có một cú pháp tương tự như gzip nhưng nó sử dụng thuật toán nén khác nhau và tạo ra các tệp nhỏ hơn đáng kể, với giá phải mất nhiều thời gian hơn để thực hiện công việc của nó. Vì vậy, nó có nhiều khả năng được sử dụng để nén các tệp lớn hơn.

Ví dụ về sử dụng phổ biến cũng tương tự như gzip:

bzip2 \* Nén tất cả các tệp trong thư mục hiện tại và thay thế từng tệp bằng một tệp được đổi tên bằng phần mở rộng .bz2

bunzip2 \*.bz2 Giải nén tất cả các tệp có phần mở rộng là .bz2 trong thư mục hiện tại. Dưới mui xe, bunzip2 giống như gọi bzip2 -d

xz là tiện ích nén hiệu quả nhất về không gian được sử dụng trong Linux và hiện được sử dụng để lưu trữ tài liệu lưu trữ của nhân Linux. Một lần nữa, nó giao dịch tốc độ nén chậm hơn cho tỷ lệ nén thậm chí cao hơn.

Một số ví dụ sử dụng:

\$ xz \* Nén tất cả các tệp trong thư mục hiện tại và thay thế từng tệp bằng một tệp có phần mở rộng .xz

xz foo Nén tệp foo vào foo.xz bằng mức nén mặc định (-6) và xóa foo nếu quá trình nén thành công

xz -dk bar.xz Giải nén bar.xz thành thanh và không xóa bar.xz ngay cả khi giải nén thành công

xz -dcf a.txt b.txt.xz> abcd.txt Giải nén hỗn hợp các tệp nén và không nén thành đầu ra tiêu chuẩn, sử dụng một lệnh duy nhất

\$ xz -d \*.xz Giải nén các tệp được nén bằng xz

Các tệp nén được lưu trữ với phần mở rộng .xz.

Chương trình zip không thường được sử dụng để nén các tệp trong Linux, nhưng thường được yêu cầu để kiểm tra và giải nén tài liệu lưu trữ từ các hệ điều hành khác. Nó chỉ được sử dụng trong Linux khi bạn nhận được tệp nén từ người dùng Windows. Đây là một chương trình kế thừa.

saو lưu zip \* Nén tất cả các tệp trong thư mục hiện tại và đặt chúng vào tệp backup.zip  
zip -r backup.zip ~ Lưu trữ thư mục đăng nhập của bạn (~) và tất cả các tệp và thư mục bên dưới nó trong tệp backup.zip

unzip backup.zip Trích xuất tất cả các tệp trong tệp backup.zip và đặt chúng vào thư mục hiện tại

Trong lịch sử, tar là viết tắt của "băng lưu trữ" và được sử dụng để lưu trữ các tệp vào băng từ. Nó cho phép bạn tạo hoặc trích xuất các tệp từ một tệp lưu trữ, thường được gọi là tarball. Đồng thời, bạn có thể tùy chọn nén trong khi tạo tệp lưu trữ và giải nén trong khi trích xuất nội dung của nó.

Dưới đây là một số ví dụ về việc sử dụng tar:

```
$ tar xvf mydir.tar Trích xuất tất cả các tệp trong mydir.tar vào thư mục mydir  
$ tar zcvf mydir.tar.gz mydir Tạo kho lưu trữ và nén bằng gzip  
$ tar jcvf mydir.tar.bz2 mydir Tạo kho lưu trữ và nén bằng bz2  
$ tar Jcvf mydir.tar.xz mydir Tạo kho lưu trữ và nén với xz  
$ tar xvf mydir.tar.gz Trích xuất tất cả các tệp trong mydir.tar.gz vào thư mục mydir
```

Lưu ý: Bạn không cần phải nói tar ở định dạng gzip

Bạn có thể tách các giai đoạn lưu trữ và nén, như trong:

```
$ tar cvf mydir.tar mydir; gzip mydir.tar  
$ gunzip mydir.tar.gz; tar xvf mydir.tar
```

nhưng điều này chậm hơn và lãng phí không gian bằng cách tạo tệp .tar trung gian không cần thiết.

## 5.19 Lệnh sao chép ổ đĩa (dd)

Chương trình dd rất hữu ích để tạo các bản sao của không gian đĩa thô. Ví dụ: để sao lưu Bản ghi khởi động chính (MBR) (cung 512 byte đầu tiên trên đĩa có chứa bảng mô tả các phân vùng trên đĩa đó), bạn có thể nhập:

```
$ dd if=/dev/sda of=sda.mbr bs=512 đếm=1
```

CẢNH BÁO!

## Gõ lệnh

```
$ dd if=/dev/sda of=/dev/sdb
```

để tạo một bản sao của một đĩa này sang đĩa khác, sẽ xóa mọi thứ đã tồn tại trước đó trên đĩa thứ hai.

Một bản sao chính xác của thiết bị đĩa thứ nhất được tạo trên thiết bị đĩa thứ hai.

Không thử nghiệm với lệnh này như được viết ở trên, vì nó có thể xóa một đĩa cứng!



Chính xác những gì tên dd là viết tắt của một mục thường được tranh luận. Định nghĩa dữ liệu từ là lý thuyết phổ biến nhất và có nguồn gốc từ lịch sử IBM sớm. Thông thường, mọi người nói đùa rằng nó có nghĩa là phá hủy đĩa và các biến thể khác như xóa dữ liệu!

## Bài thực hành LAB

### 1. Thay đổi thư mục /tmp.

Sao chép tệp văn bản vào /tmp. Ví dụ: sao chép /etc/group vào /tmp.

dd không thể chỉ sao chép trực tiếp từ các thiết bị đĩa thô, mà còn từ các tệp thông thường. Hãy nhớ rằng, trong Linux, mọi thứ được coi là một tệp. dd cũng có thể thực hiện các chuyển đổi khác nhau. Ví dụ: tùy chọn conv = ucase sẽ chuyển đổi tất cả các ký tự thành ký tự chữ hoa. Chúng tôi sẽ sử dụng dd để sao chép tệp văn bản sang tệp mới trong /tmp trong khi chuyển đổi các ký tự thành chữ hoa, như trong:

```
$ dd if=/tmp/group of=/tmp/GROUP conv=ucase.
```

Theo trang man cho bản vá, các tùy chọn ưa thích để chuẩn bị một bản vá với diff là -Naur khi so sánh hai cây thư mục theo cách đệ quy. Chúng tôi sẽ bỏ qua tùy chọn -a, có nghĩa là coi tất cả các tệp là văn bản, vì patch và diff chỉ nên được sử dụng trên các tệp văn bản. Vì chúng tôi chỉ so sánh hai tệp, chúng tôi không cần sử dụng tùy chọn N hoặc r để tìm khác biệt, nhưng chúng tôi có thể sử dụng chúng vì dù sao nó sẽ không tạo ra sự khác biệt. So sánh nhóm và NHÓM bằng cách sử dụng diff và chuẩn bị một tệp vá thích hợp.

Sử dụng bản vá để vá tệp gốc, /tmp/nhóm, do đó, nội dung của tệp hiện khớp với nội dung của tệp đã sửa đổi, /tmp/GROUP. Bạn có thể thử với tùy chọn --dry-run trước!

Cuối cùng, để chứng minh rằng tệp gốc của bạn hiện đã được vá thành cùng một tệp với tất cả các ký tự chữ hoa, hãy sử dụng diff trên hai tệp đó. Các tập tin phải giống nhau và bạn sẽ không nhận được bất kỳ đầu ra nào từ diff.

## 2. Khám phá các hệ thống tập tin gắn kết

Thực thi các lệnh sau:

**\$ cat /etc/fstab**

Bây giờ gõ lệnh:

**\$ mount**

So sánh kết quả. Sự khác biệt là gì?

**3. Tìm một cách khác để xem danh sách các hệ thống tập tin được gắn kết, bằng cách kiểm tra /proc giả hệ thống tập tin.**

## 4. Sao lưu thư mục người dùng

Lưu trữ (hoặc sao lưu) các tập tin của bạn theo thời gian là vệ sinh tốt cần thiết. Bạn có thể nhập một lệnh và do đó vô tình các tệp clobber bạn cần và không có nghĩa là thay đổi. Hơn nữa, trong khi phần cứng của bạn có thể được coi là khá đáng tin cậy, cuối cùng tất cả các thiết bị đều thất bại (ngay cả khi đó chỉ là một sự cố mất điện đột xuất). Thông thường, điều này xảy ra vào thời điểm tồi tệ nhất có thể. Định kỳ sao lưu các tập tin là một thói quen tốt để có được vào.

Tất nhiên, điều quan trọng là phải sao lưu vào các hệ thống bên ngoài thông qua mạng hoặc vào bộ lưu trữ ngoài, chẳng hạn như ổ đĩa ngoài hoặc thẻ nhớ USB. Ở đây, chúng tôi sẽ tạo một kho lưu trữ sao lưu trên cùng một hệ thống, rất hữu ích, nhưng đã giành được sự giúp đỡ nếu ổ đĩa bị hỏng nghiêm trọng, hoặc máy tính của bạn bị đánh cắp hoặc tòa nhà bị phá hủy bởi một tiểu hành tinh hoặc hỏa hoạn.

Đầu tiên, sử dụng tar, sao lưu tất cả các tệp và thư mục con trong thư mục chính của bạn. Đặt tệp tarball kết quả vào thư mục / tmp, đặt tên là backup.tar.

Thứ hai, hoàn thành nhiệm vụ tương tự với nén gzip bằng cách sử dụng tùy chọn -z để tar, tạo /tmp/backup.tar.gz.

So sánh kích thước của hai tệp (với ls -l).

Để có thêm kinh nghiệm, hãy tạo các bản sao lưu bằng tùy chọn -j bằng cách sử dụng nén bzip2 và tùy chọn -J để sử dụng nén xz.

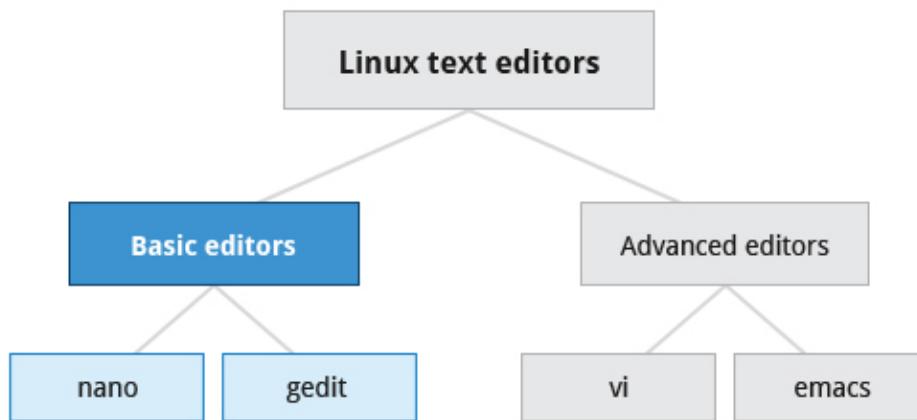
## 5.20 Các trình soạn thảo văn bản text

### Giới thiệu

Tại một số điểm, bạn sẽ cần phải chỉnh sửa thủ công các tệp văn bản. Bạn có thể soạn email ngoại tuyến, viết tập lệnh được sử dụng cho bash hoặc các trình thông dịch lệnh khác, thay đổi tệp cấu hình hệ thống hoặc ứng dụng hoặc phát triển mã nguồn cho ngôn ngữ lập trình như C hoặc Java.

Quản trị viên Linux khá thường xuyên bỏ qua các trình soạn thảo văn bản, bằng cách sử dụng các tiện ích đồ họa để tạo và sửa đổi các tệp cấu hình hệ thống. Tuy nhiên, điều này có thể tốn nhiều công sức hơn so với trực tiếp sử dụng trình soạn thảo văn bản và bị hạn chế hơn về khả năng. Lưu ý rằng các ứng dụng xử lý văn bản bao gồm các ứng dụng là một phần của bộ công cụ văn phòng không thực sự là trình soạn thảo văn bản vì chúng bổ sung nhiều thông tin định dạng bổ sung (thường là vô hình) có thể sẽ khiến các tệp cấu hình quản trị hệ thống không thể sử dụng cho mục đích của chúng. Vì vậy, sử dụng trình soạn thảo văn bản thực sự rất cần thiết trong Linux.

Đến bây giờ, bạn chắc chắn đã nhận ra Linux có rất nhiều sự lựa chọn; Khi nói đến trình soạn thảo văn bản, có rất nhiều sự lựa chọn, từ khá đơn giản đến rất phức tạp, bao gồm: **nano**, **gedit**, **vi**, **emacs**

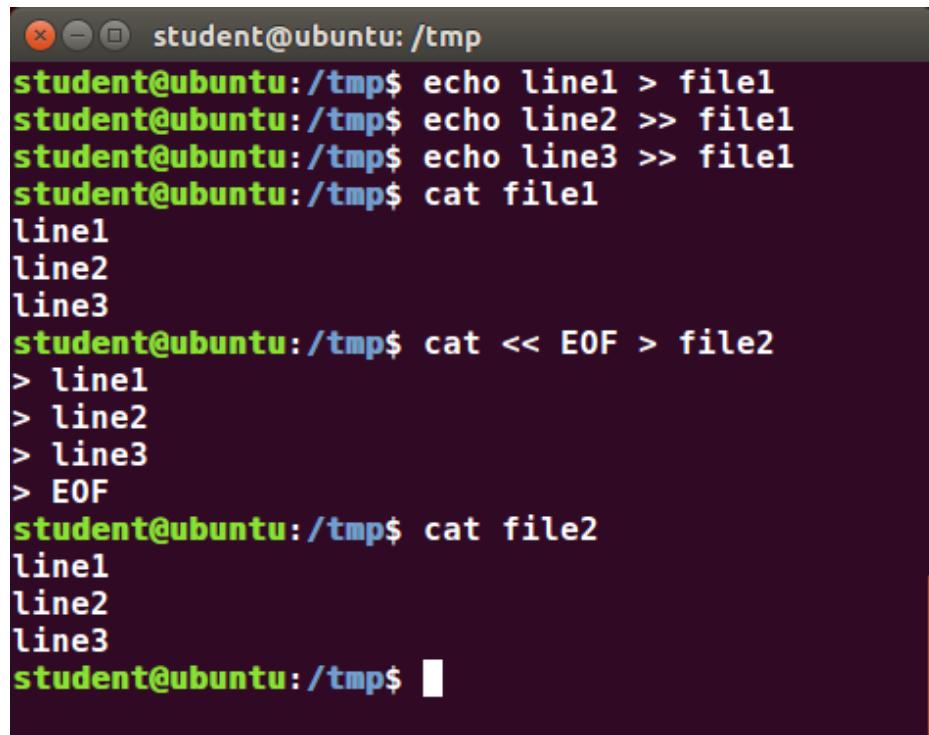


Trong phần này, chúng ta sẽ tìm hiểu về các trình soạn thảo nano và gedit, tương đối đơn giản và dễ học. Trước khi bắt đầu, chúng ta hãy xem xét một số trường hợp không cần trình chỉnh sửa.

### Các lệnh tạo tập tin không dùng trình soạn thảo

Đôi khi, bạn có thể muốn tạo một tệp ngắn và không muốn gọi trình soạn thảo toàn văn. Ngoài ra, làm như vậy có thể khá hữu ích khi được sử dụng từ bên trong các tập lệnh, ngay cả khi tạo các tệp dài hơn. Bạn chắc chắn sẽ thấy mình sử dụng phương pháp này khi bạn bắt đầu vào các chương sau bao gồm kịch bản bash!

Nếu bạn muốn tạo một tệp mà không cần sử dụng trình soạn thảo, có hai cách tiêu chuẩn để tạo một tệp từ dòng lệnh và điền vào đó bằng nội dung.



```
student@ubuntu:/tmp
student@ubuntu:/tmp$ echo line1 > file1
student@ubuntu:/tmp$ echo line2 >> file1
student@ubuntu:/tmp$ echo line3 >> file1
student@ubuntu:/tmp$ cat file1
line1
line2
line3
student@ubuntu:/tmp$ cat << EOF > file2
> line1
> line2
> line3
> EOF
student@ubuntu:/tmp$ cat file2
line1
line2
line3
student@ubuntu:/tmp$
```

Đầu tiên là sử dụng echo nhiều lần:

```
$ echo dòng một> myfile
$ echo dòng hai >> myfile
$ echo dòng ba >> myfile
```

Lưu ý rằng mặc dù một dấu lớn hơn (>) sẽ gửi đầu ra của lệnh tới một tệp, hai trong số chúng (>>) sẽ nối đầu ra mới vào một tệp hiện có.

Cách thứ hai là sử dụng mèo kết hợp với chuyển hướng:

```
$ cat << EOF> myfile
> dòng một
> dòng hai
> dòng ba
> EOF
$
```

Cả hai kỹ thuật tạo ra một tệp có các dòng sau:

```
dòng một
dòng hai
```

dòng ba

và cực kỳ hữu ích khi được sử dụng bởi các kịch bản.

## 5.21 Lệnh nano và gedit

Có một số trình soạn thảo văn bản khá rõ ràng; họ không đòi hỏi kinh nghiệm đặc biệt để học và thực sự khá có khả năng, nếu không mạnh mẽ. Một thứ đặc biệt dễ sử dụng là nano soạn thảo dựa trên thiết bị đầu cuối văn bản. Chỉ cần gọi nano bằng cách đặt tên tệp làm đối số. Tất cả các trợ giúp bạn cần được hiển thị ở dưới cùng của màn hình và bạn sẽ có thể tiến hành mà không gặp vấn đề gì.

Là một trình soạn thảo đồ họa, gedit là một phần của hệ thống máy tính để bàn Gnome (kwrite được liên kết với KDE). Các trình soạn thảo gedit và kwrite rất dễ sử dụng và cực kỳ có khả năng. Họ cũng rất cấu hình. Chúng trông rất giống Notepad trong Windows. Các biến thể khác như kate cũng được KDE hỗ trợ.

File Edit View Search Terminal Help

GNU nano 2.3.1 File: /tmp/lab\_char.h

```

#ifndef _LAB_CHAR_H
#define _LAB_CHAR_H

#include <linux/module.h>
#include <linux/fs.h>
#include <linux/uaccess.h>
#include <linux/sched.h>
#include <linux/init.h>
#include <linux/slab.h>
#include <linux/cdev.h>
#include <linux/device.h>

#define MYDEV_NAME "mycdrv"

static struct device *my_dev;
static char *ramdisk;
static size_t ramdisk_size = (16 * PAGE_SIZE);
static dev_t first;
static unsigned int count = 1; /* number of dev_t needed */
static struct cdev *my_cdev;
static struct class *foo_class;

static const struct file_operations mycdrv_fops;

/* generic entry points */

static inline int mycdrv_generic_open(struct inode *inode, struct file *file)
{
    static int counter;
    dev_info(my_dev, " attempting to open device: %s:\n", MYDEV_NAME);
    dev_info(my_dev, " MAJOR number = %d, MINOR number = %d\n",
            imajor(inode), iminor(inode));
    counter++;

    dev_info(my_dev, " successfully open device: %s:\n\n", MYDEV_NAME);
    [ Read 134 lines ]

```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
 ^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

**nano** rất dễ sử dụng và cần rất ít nỗ lực để học. Để mở tệp bằng nano, nhập nano <tên tệp> và nhấn Enter. Nếu tập tin không tồn tại, nó sẽ được tạo.

nano cung cấp một thanh phím tắt hai dòng, trực tuyến, ở cuối màn hình liệt kê các lệnh có sẵn. Một số lệnh này là:

CTRL-G

Hiển thị màn hình trợ giúp.

CTRL-O

Viết vào một tập tin.

## CTRL-X

Thoát một tập tin.

## CTRL-R

Chèn nội dung từ tệp khác vào bộ đệm hiện tại.

## CTRL-C

Hủy bỏ các lệnh trước đó.

```
#ifndef _LAB_CHAR_H
#define _LAB_CHAR_H

#include <linux/module.h>
#include <linux/fs.h>
#include <linux/uaccess.h>
#include <linux/sched.h>
#include <linux/init.h>
#include <linux/slab.h>
#include <linux/cdev.h>
#include <linux/device.h>

#define MYDEV_NAME "mycdrv"

static struct device *my_dev;
static char *ramdisk;
static size_t ramdisk_size = (16 * PAGE_SIZE);
static dev_t first;
static unsigned int count = 1; /* number of dev_t needed */
static struct cdev *my_cdev;
static struct class *foo_class;

static const struct file_operations mycdrv_fops;

/* generic entry points */

static inline int mycdrv_generic_open(struct inode *inode, struct file **file)
{
    static int counter;
    dev_info(my_dev, " attempting to open device: %s:\n", MYDEV_NAME);
    dev_info(my_dev, " MAJOR number = %d, MINOR number = %d\n",
            imajor(inode), iminor(inode));
    counter++;

    dev_info(my_dev, " successfully open device: %s:\n", MYDEV_NAME);
    dev_info(my_dev, " I have been opened %d times since being loaded\n",
            counter);
    dev_info(my_dev, " ref=%d\n", (int)module_refcount(THIS_MODULE));

    return 0;
}
```

C/C++/ObjC Header ▾ Tab Width: 8 ▾ Ln 7, Col 25 ▾ INS

## gedit

gedit (phát âm 'g-edit') là một trình soạn thảo đồ họa dễ sử dụng, chỉ có thể chạy trong môi trường Đồ họa. Nó khá giống với trình soạn thảo văn bản Notepad trong Windows, nhưng thực sự có khả năng và cấu hình cao hơn rất nhiều và có rất nhiều plugin có sẵn để mở rộng khả năng của nó hơn nữa.

Để mở một tệp mới trong gedit, hãy tìm chương trình trong hệ thống menu của máy tính để bàn của bạn hoặc từ loại dòng lệnh gedit <tên tệp>. Nếu tập tin không tồn tại, nó sẽ được tạo.

Sử dụng gedit khá đơn giản và không cần đào tạo nhiều. Giao diện của nó bao gồm các yếu tố khá quen thuộc.

### Bài tập Thực hành LAB

1. Sử dụng nano, tạo một tệp có tên myname.txt và bao gồm tên của bạn trên dòng đầu tiên và ngày hiện tại trên dòng cuối cùng. Để làm điều này:

Bắt đầu nano bằng cách gõ nano myfile.txt.

Thêm tên của bạn trong dòng đầu tiên của tập tin.

Thêm ngày trong dòng cuối cùng của tập tin.

Đóng tệp.

2, Sử dụng gedit, tạo hoặc sử dụng lại một tệp có tên myname.txt và để nó bao gồm địa chỉ đường phố của bạn trên dòng thứ hai và tên thành phố của bạn trên dòng cuối cùng:

Bắt đầu gedit bằng cách gõ gedit myfile.txt.

Thêm địa chỉ đường phố của bạn trong dòng thứ hai của tập tin.

Thêm tên thành phố của bạn vào dòng cuối cùng của tệp.

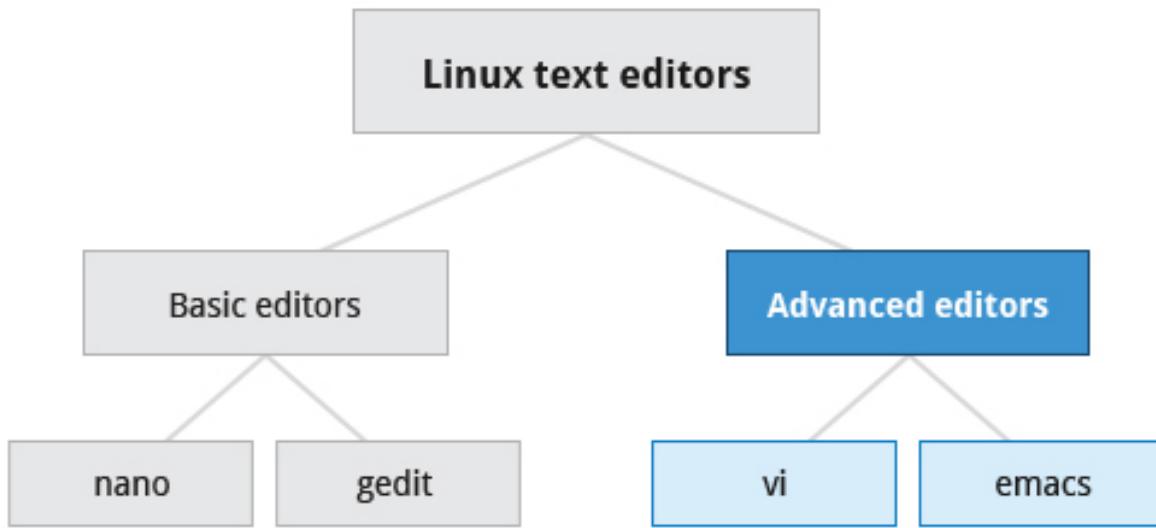
Đóng tệp.

### 5.22 Lệnh vi and emacs

Các nhà phát triển và quản trị viên có kinh nghiệm làm việc trên các hệ thống giống như UNIX hầu như luôn sử dụng một trong hai tùy chọn chỉnh sửa đáng kính: vi và emacs. Cả hai đều có mặt hoặc dễ dàng có sẵn trên tất cả các bản phân phối và hoàn toàn tương thích với các phiên bản có sẵn trên các hệ điều hành khác.

Cả vi và emacs đều có dạng văn bản hoàn toàn cơ bản có thể chạy trong môi trường phi đồ họa. Họ cũng có một hoặc nhiều hình thức giao diện đồ họa với khả năng mở rộng; những thứ này có thể thân thiện hơn với người dùng ít kinh nghiệm Mặc dù vi và emacs có thể có đường cong học tập dốc đáng kể cho người dùng mới, nhưng chúng cực kỳ hiệu quả khi người ta đã học cách sử dụng chúng.

Bạn cần lưu ý rằng những trận đánh giữa những người dùng dày dạn về việc biên tập viên nào tốt hơn có thể khá dữ dội và thường được mô tả như một cuộc chiến thần thánh.



## Lệnh vi

Thông thường, chương trình thực tế được cài đặt trên hệ thống của bạn là vim, viết tắt của VIMproved và được đặt tên là vi. Cái tên này được phát âm là từ vee-eye.

Ngay cả khi bạn không muốn sử dụng vi, vẫn tốt để làm quen với nó: đó là một công cụ tiêu chuẩn được cài đặt trên hầu như tất cả các bản phân phối Linux. Thật vậy, có thể có những lúc không có trình soạn thảo nào khác có sẵn trên hệ thống.

Gnome mở rộng vi với giao diện đồ họa rất nổi tiếng là gvim và KDE cung cấp kvim. Một trong những thứ này có thể dễ sử dụng hơn lúc đầu.

Khi sử dụng vi, tất cả các lệnh được nhập thông qua bàn phím. Bạn không cần phải tiếp tục di chuyển tay để sử dụng một thiết bị con trỏ như chuột hoặc bàn di chuột, trừ khi bạn muốn làm như vậy khi sử dụng một trong các phiên bản đồ họa của trình chỉnh sửa.

Typing vimtutor khởi chạy một hướng dẫn ngắn nhưng rất toàn diện cho những ai muốn tìm hiểu các lệnh vi đầu tiên của họ. Hướng dẫn này là một nơi tốt để bắt đầu học vi.

Mặc dù nó chỉ cung cấp phần giới thiệu và chỉ bảy bài học, nhưng nó có đủ tài liệu để giúp bạn trở thành người dùng vi rất thành thạo, vì nó bao gồm một số lượng lớn các lệnh.

```
File Edit View Search Terminal Help

VIM - Vi IMproved

version 7.4.160
by Bram Moolenaar et al.
Modified by <bugzilla@redhat.com>
Vim is open source and freely distributable

      Help poor children in Uganda!
type :help iccf<Enter>      for information

type :q<Enter>              to exit
type :help<Enter> or <F1>   for on-line help
type :help version7<Enter>  for version info

0,0-1          All
```

## vimtutor

Sau khi học những cái cơ bản này, bạn có thể tìm kiếm các thủ thuật mới để kết hợp vào danh sách các lệnh vi của bạn bởi vì luôn có nhiều cách tối ưu hơn để thực hiện mọi thứ trong vi mà ít gõ hơn.

student@ubuntu: ~/Pictures

```
= Welcome to the VIM Tutor - Version 1.7 =
Vim is a very powerful editor that has many commands, too many to
explain in a tutor such as this. This tutor is designed to describe
enough of the commands that you will be able to easily use Vim as
an all-purpose editor.

The approximate time required to complete the tutor is 25-30 minutes,
depending upon how much time is spent with experimentation.

ATTENTION:
The commands in the lessons will modify the text. Make a copy of this
file to practise on (if you started "vimtutor" this is already a copy).

It is important to remember that this tutor is set up to teach by
use. That means that you need to execute the commands to learn them
properly. If you only read the text, you will forget the commands!

Now, make sure that your Shift-Lock key is NOT depressed and press
the j key enough times to move the cursor so that Lesson 1.1
completely fills the screen.

Lesson 1.1: MOVING THE CURSOR

** To move the cursor, press the h,j,k,l keys as indicated. **
```

Các chế độ

## **lệnh vi**

vi cung cấp ba chế độ, như được mô tả trong bảng dưới đây. Điều quan trọng là không bị mất dấu của chế độ bạn đang ở. Nhiều tổ hợp phím và lệnh hoạt động khá khác nhau trong các chế độ khác nhau.

### **Command**

Theo mặc định, vi bắt đầu trong chế độ Lệnh.

Mỗi khóa là một lệnh biên tập.

Nét bàn phím được hiểu là các lệnh có thể sửa đổi nội dung tệp.

### **Insert**

Nhập i để chuyển sang chế độ Chèn từ chế độ Lệnh.

Chế độ chèn được sử dụng để nhập (chèn) văn bản vào một tệp.

Chế độ chèn được chỉ định bởi một người dùng? CHÈN ?" chỉ báo ở dưới cùng của màn hình.

Nhấn Esc để thoát chế độ Chèn và trở về chế độ Lệnh.

### **Line**

Nhập: để chuyển sang chế độ Dòng từ chế độ Lệnh. Mỗi khóa là một lệnh bên ngoài, bao gồm các thao tác như ghi nội dung tệp vào đĩa hoặc thoát.

Sử dụng các lệnh chỉnh sửa dòng được kế thừa từ các trình soạn thảo dòng cũ hơn. Hầu hết các lệnh này thực sự không còn được sử dụng. Một số lệnh chỉnh sửa dòng rất mạnh mẽ.

Nhấn Esc để thoát chế độ Line và trở về chế độ Command.

### Làm việc với tập tin trong vi

Bảng mô tả các lệnh quan trọng nhất được sử dụng để bắt đầu, thoát, đọc và ghi tệp trong vi. Phím ENTER cần được nhấn sau tất cả các lệnh này

:vi myfile Khởi động trình soạn thảo vi và chỉnh sửa tệp myfile

:vi -r myfile Bắt đầu vi và chỉnh sửa myfile trong chế độ khôi phục từ sự cố hệ thống

:r file2 Đọc trong file2 và chèn vào vị trí hiện tại

:w Ghi vào tập tin

:w myfile Viết tập tin vào myfile

:w! file2 Ghi đè lên file2

:x hoặc: wq Thoát vi và ghi ra tệp sửa đổi

:q Thoát vi

:q! Thoát khỏi vi ngay cả khi sửa đổi chưa được lưu

### Thay đổi trạng thái con trỏ trong vi

Bảng sau mô tả các tổ hợp phím quan trọng nhất được sử dụng khi thay đổi vị trí con trỏ trong vi. Các lệnh chế độ dòng (những dấu hai chấm sau :) yêu cầu nhấn phím ENTER sau khi gõ lệnh.

j hoặc <ret> Để di chuyển xuống, trái và phải

k Để di chuyển một dòng lên

h hoặc Backspace Để di chuyển một ký tự sang trái

l hoặc dấu cách để di chuyển đúng một ký tự

0 Để di chuyển đến đầu dòng

\$ Để di chuyển đến cuối dòng

w Để di chuyển đến đầu từ tiếp theo

: 0 hoặc 1G Để di chuyển đến đầu tập tin

: n hoặc nG Để di chuyển đến dòng n

: \$ hoặc G Để di chuyển đến dòng cuối cùng trong tệp

CTRL-F hoặc Page Down Để di chuyển về phía trước một trang

CTRL-B hoặc Page Up Để di chuyển lùi một trang

^ l Để làm mới và màn hình trung tâm

### **Tìm kiếm văn bản trong vi**

Bảng mô tả các lệnh quan trọng nhất được sử dụng khi tìm kiếm văn bản trong vi. Phím ENTER phải được nhấn sau khi gõ mẫu tìm kiếm.

/ mẫu Tìm kiếm chuyển tiếp cho mẫu

? mẫu Tìm kiếm ngược cho mẫu

Bảng mô tả các tổ hợp phím quan trọng nhất được sử dụng khi tìm kiếm văn bản trong vi.

n Di chuyển đến lần xuất hiện tiếp theo của mẫu tìm kiếm

N Di chuyển đến lần xuất hiện trước của mẫu tìm kiếm

### **Làm việc với văn bản trong vi**

Bảng mô tả các tổ hợp phím quan trọng nhất được sử dụng khi thay đổi, thêm và xóa văn bản trong vi.

Bạn cũng có thể tải xuống tệp PDF hợp nhất với các lệnh cho vi.

Nối văn bản sau con trỏ; dừng lại khi thoát phím

Một văn bản chép thêm ở cuối dòng hiện tại; dừng lại khi thoát phím

i Chèn văn bản trước con trỏ; dừng lại khi thoát phím

I Chèn văn bản ở đầu dòng hiện tại; dừng lại khi thoát phím

o Bắt đầu một dòng mới bên dưới dòng hiện tại, chèn văn bản ở đó; dừng lại khi thoát phím

O Bắt đầu một dòng mới trên dòng hiện tại, chèn văn bản ở đó; dừng lại khi thoát phím

r Thay thế nhân vật ở vị trí hiện tại

R Thay thế văn bản bắt đầu bằng vị trí hiện tại; dừng lại khi thoát phím

x Xóa ký tự ở vị trí hiện tại

Nx Xóa N ký tự, bắt đầu từ vị trí hiện tại

dw Xóa từ ở vị trí hiện tại  
D Xóa phần còn lại của dòng hiện tại  
dd Xóa dòng hiện tại  
Ndd hoặc dNd Xóa N dòng  
u Hoàn tác thao tác trước  
yy Yank (sao chép) dòng hiện tại và đặt nó vào bộ đệm  
Nyy hoặc yNy Yank (sao chép) N dòng và đặt nó vào bộ đệm  
p Dán tại vị trí hiện tại dòng được kéo dài hoặc dòng từ bộ đệm.

### Sử dụng các lệnh bên ngoài

Gõ : lệnh sh mở một vỏ lệnh bên ngoài. Khi bạn thoát khỏi trình bao, bạn sẽ tiếp tục phiên chỉnh sửa vi của mình. Gõ :! thực hiện một lệnh từ bên trong vi. Lệnh tuân theo dấu chấm than. Kỹ thuật này phù hợp nhất cho các lệnh không tương tác, chẳng hạn như :! wc%. Gõ này sẽ chạy lệnh wc (đếm từ) trên tệp; % ký tự đại diện cho tập tin hiện đang được chỉnh sửa.

```

File Edit View Search Terminal Help
/* ***** LFD430:4.9 s_08/lab_miscdev.h ***** */
/*
 * The code herein is: Copyright the Linux Foundation, 2016
 *
 * This Copyright is retained for the purpose of protecting free
 * redistribution of source.
 *
 * URL: http://training.linuxfoundation.org
 * email: trainingquestions@linuxfoundation.org
 *
 * This code is distributed under Version 2 of the GNU General Public
 * License, which you should have received with the source.
 */
/*
 *@*/
#ifndef _LAB_CHAR_H
#define _LAB_CHAR_H

:! wc %

```

```

File Edit View Search Terminal Help
c7:/tmp/s_08>vi lab_miscdev.h
113 348 3075 lab_miscdev.h
Press ENTER or type command to continue

```

## Lệnh emacs

Trình soạn thảo emacs là một đối thủ cạnh tranh phổ biến cho vi. Không giống như vi, nó không hoạt động với các chế độ. emacs có khả năng tùy biến cao và bao gồm một số lượng lớn các tính năng. Ban đầu nó được thiết kế để sử dụng trên bàn điều khiển, nhưng cũng sớm được điều chỉnh để hoạt động với GUI. emacs có nhiều khả năng khác ngoài chỉnh sửa văn bản đơn giản. Ví dụ, nó có thể được sử dụng cho email, gỡ lỗi, v.v.

Thay vì có các chế độ khác nhau cho lệnh và chèn, như vi, emacs sử dụng các phím CTRL và Meta (Alt hoặc Esc) cho các lệnh đặc biệt.

Bảng liệt kê một số tổ hợp phím quan trọng nhất được sử dụng khi bắt đầu, thoát, đọc và ghi tệp trong emacs.

emacs myfile	# Bắt đầu emacs và chỉnh sửa myfile
CTRL-x i	# Chèn được nhắc cho tệp ở vị trí hiện tại
CTRL-x s	# Lưu tất cả các tệp
CTRL-x CTRL-w	

The screenshot shows the GNU Emacs 24.5.1 interface. The main window displays a shell script named `DO_KERNEL.sh`. The script content includes comments about kernel compilation and distribution detection. Below the script, a message from the Emacs manual is visible: "Welcome to [GNU Emacs](#), one component of the [GNU/Linux](#) operating system." A help menu is open, listing various Emacs resources and their descriptions. The status bar at the bottom shows the current buffer is `*GNU Emacs*` and the mode is `(Fundamental)`.

[Emacs Tutorial](#) Learn basic keystroke commands  
[Emacs Guided Tour](#) Overview of Emacs features at gnu.org  
[View Emacs Manual](#) View the Emacs manual using Info  
[Absence of Warranty](#) GNU Emacs comes with ABSOLUTELY NO WARRANTY  
[Copying Conditions](#) Conditions for redistributing and changing Emacs  
[Ordering Manuals](#) Purchasing printed copies of manuals

To quit a partially entered command, type `Control-g`.

This is GNU Emacs 24.5.1 (x86\_64-pc-linux-gnu, GTK+ Version 3.18.9)  
of 2016-04-17 on lgw01-04, modified by Debian  
Copyright (C) 2015 Free Software Foundation, Inc.

Ghi vào tập tin đặt tên mới khi được nhắc

CTRL-x CTRL-s Lưu tệp hiện tại

CTRL-x CTRL-c Thoát sau khi được nhắc lưu bất kỳ tệp sửa đổi nào

Hướng dẫn emacs là một nơi tốt để bắt đầu học các lệnh emacs cơ bản. Nó có sẵn bắt cứ lúc nào khi trong emacs bằng cách chỉ cần gõ `CTRL-h` (để được trợ giúp) và sau đó là chữ `t` để hướng dẫn.

### Thay đổi vị trí con trỏ trong Emacs

Bảng liệt kê một số phím và tổ hợp phím được sử dụng để thay đổi vị trí con trỏ trong emacs.

**Arrow keys** Sử dụng các phím mũi tên để lên, xuống, trái và phải

CTRL-n Một dòng xuống

CTRL-p Một dòng lên

CTRL-f Một ký tự tiến / phải

CTRL-b Một ký tự lùi / trái

CTRL-a Di chuyển đến đầu dòng

CTRL-e Di chuyển đến cuối dòng

Meta-f Di chuyển đến đầu từ tiếp theo

Meta-b Quay trở lại từ đầu của từ trước

Meta- <Di chuyển đến đầu tập tin

Meta-g-g-n Di chuyển đến dòng n (cũng có thể sử dụng 'Esc-x Goto-line n')

Meta-> Di chuyển đến cuối tệp

CTRL-v hoặc Page Down Di chuyển về phía trước một trang

Meta-v hoặc Page Up Di chuyển lùi một trang

CTRL-l Làm mới và màn hình trung tâm

### **Tìm kiếm văn bản trong Emacs**

Bảng liệt kê các tổ hợp phím được sử dụng để tìm kiếm văn bản trong emacs.

CTRL-s Tìm kiếm chuyển tiếp cho mẫu được nhắc hoặc cho mẫu tiếp theo

CTRL-r Tìm kiếm ngược cho mẫu được nhắc hoặc cho mẫu tiếp theo

### **Làm việc với văn bản trong Emacs**

Bảng liệt kê một số tổ hợp phím được sử dụng để thay đổi, thêm và xóa văn bản trong emacs:

CTRL-o Chèn một dòng trống

CTRL-d Xóa ký tự ở vị trí hiện tại

CTRL-k Xóa phần còn lại của dòng hiện tại

CTRL-\_ Hoàn tác thao tác trước đó

CTRL- (dấu cách hoặc CTRL- @) Đánh dấu điểm bắt đầu của vùng đã chọn. Kết thúc sẽ ở vị trí con trỏ

CTRL-w Xóa văn bản được đánh dấu hiện tại và ghi nó vào bộ đệm

CTRL-y Chèn vào vị trí con trỏ hiện tại bất cứ thứ gì được xóa gần đây nhất

## **5.23 Câu hỏi ôn tập Chương 5**

# Chương 6 Môi trường làm việc cho người dùng trên Linux

## 6.1 Xác nhận người dùng hiện hữu trên Linux

Như bạn đã biết, Linux là một hệ điều hành nhiều người dùng, nghĩa là nhiều người dùng có thể đăng nhập cùng một lúc.

Để xác định người dùng hiện tại, nhập **whoami**

Để liệt kê những người dùng đang đăng nhập, hãy nhập **who**.

Cung cấp cho who tùy chọn -a sẽ cung cấp thông tin chi tiết hơn.

### Các tập tin khởi tạo môi trường làm việc của người dùng

Trong Linux, chương trình shell lệnh (thường là bash) sử dụng một hoặc nhiều tệp khởi động để định cấu hình môi trường người dùng. Các tệp trong thư mục /etc xác định cài đặt chung cho tất cả người dùng, trong khi các tệp khởi tạo trong thư mục chính của người dùng có thể bao gồm và / hoặc ghi đè cài đặt chung.

Các tệp khởi động có thể làm bất cứ điều gì mà người dùng muốn làm trong mọi shell lệnh, chẳng hạn như:

Tùy chỉnh lời nhắc

Xác định các phím tắt và bí danh dòng lệnh

Đặt trình soạn thảo văn bản mặc định

Đặt đường dẫn cho nơi tìm chương trình thực thi.

### Thứ tự các tập tin khởi tạo

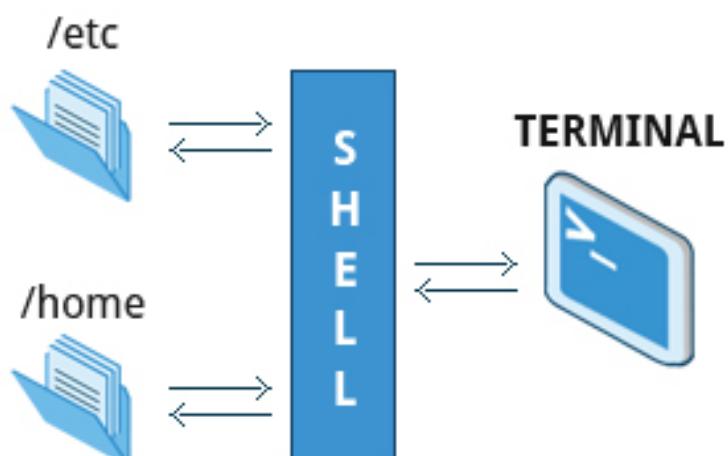
Đơn thuần tiêu chuẩn là khi bạn đăng nhập lần đầu vào Linux, /etc/profile được đọc và đánh giá, sau đó các tệp sau được tìm kiếm (nếu chúng tồn tại) theo thứ tự được liệt kê:

~/.bash\_profile

~/.bash\_login

~/.

trong đó ~/. biểu thị thư mục nhà của người dùng. Shell đăng nhập Linux đánh giá bất kỳ tệp khởi động nào mà nó xuất hiện đầu tiên và bỏ qua phần còn lại. Điều này có nghĩa là

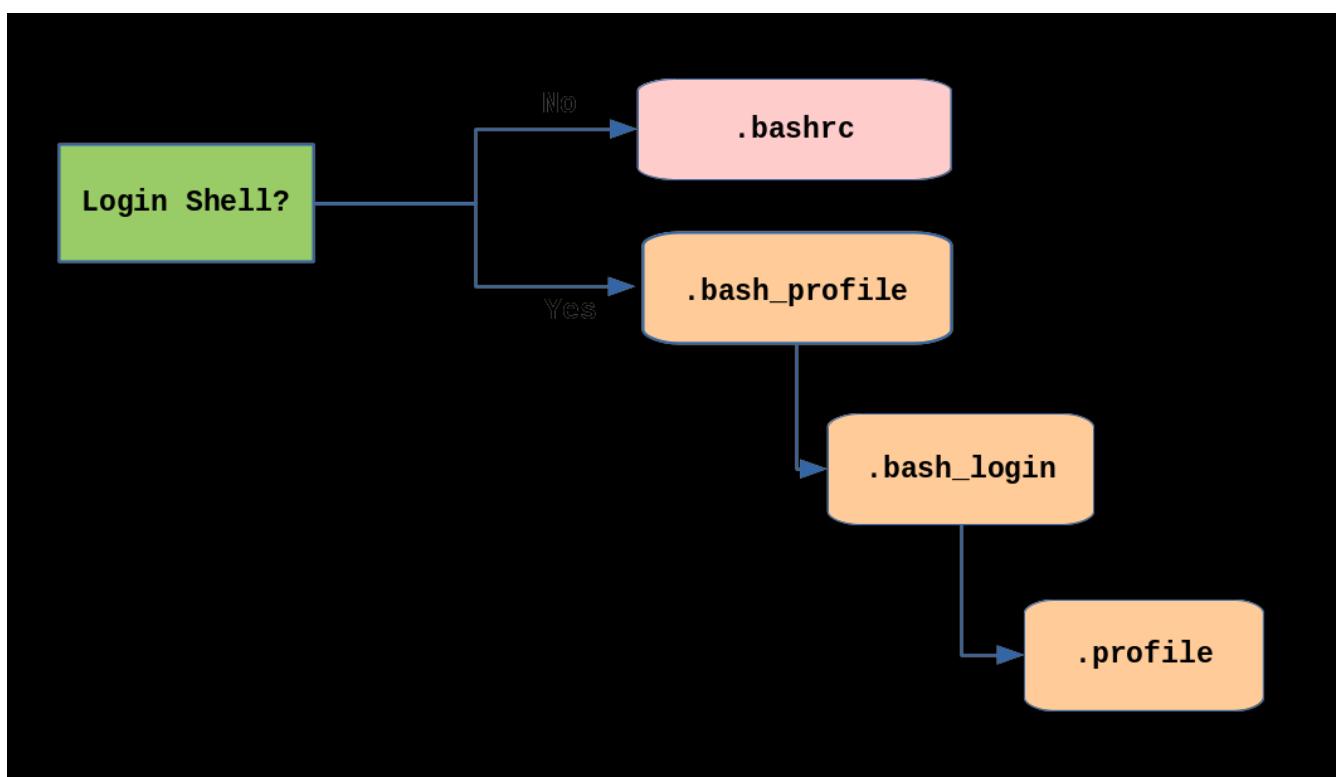


nếu nó tìm thấy `~/.bash_profile`, nó sẽ bỏ qua `~/.bash_login` và `~/.profile`. Các bản phân phối khác nhau có thể sử dụng các tệp khởi động khác nhau.

Tuy nhiên, mỗi khi bạn tạo một shell mới hoặc cửa sổ terminal, v.v., bạn không thực hiện đăng nhập toàn hệ thống; chỉ một tệp có tên `~/.bashrc` được đọc và đánh giá. Mặc dù tệp này không được đọc và đánh giá cùng với vỏ đăng nhập, hầu hết các bản phân phối và / hoặc người dùng bao gồm tệp `~/.bashrc` từ một trong ba tệp khởi động do người dùng sở hữu.

Thông thường nhất, người dùng chỉ sử dụng `~/.bashrc`, vì nó được gọi mỗi khi trình vỏ dòng lệnh mới khởi chạy hoặc chương trình khác được khởi chạy từ cửa sổ đầu cuối, trong khi các tệp khác chỉ được đọc và thực thi khi người dùng đăng nhập lần đầu hệ thống.

Các bản phân phối gần đây đôi khi thậm chí không có `.bash_profile` hoặc `.bash_login` (Ubuntu) và một số chỉ cần làm ít hơn là bao gồm `.bashrc`.

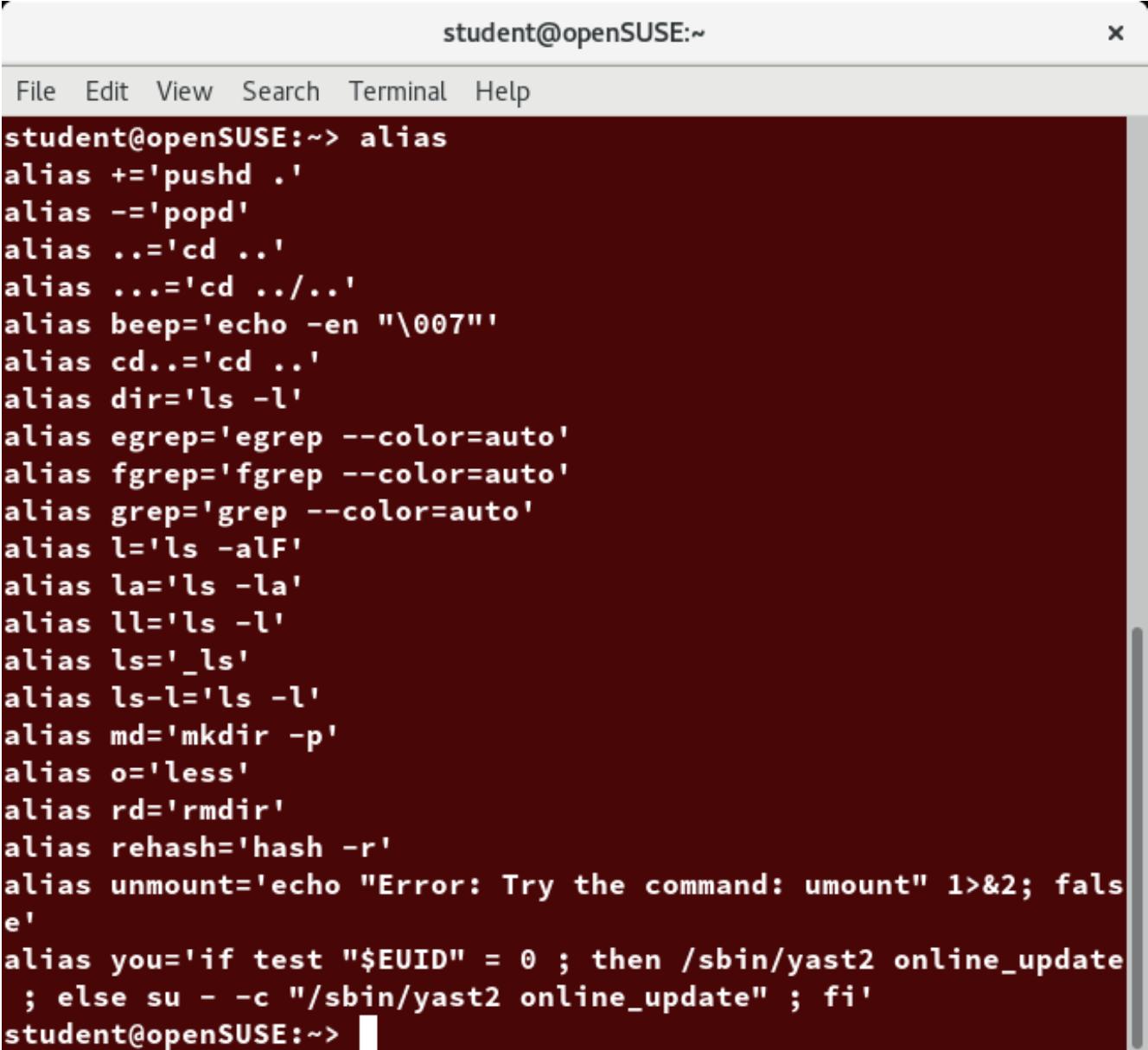


### Tạo các bí danh (Aliases)

Bạn có thể tạo các lệnh tùy chỉnh hoặc sửa đổi hành vi của các lệnh đã có bằng cách tạo bí danh. Thông thường, các bí danh này được đặt trong tệp `~/.bashrc` của bạn để chúng có sẵn cho bất kỳ shell lệnh nào bạn tạo. `unalias` loại bỏ một bí danh.

Nhập bí danh không có đối số sẽ liệt kê các bí danh hiện được xác định.

Xin lưu ý rằng không nên có bất kỳ khoảng trắng nào ở hai bên của dấu bằng và định nghĩa bí danh cần được đặt trong dấu ngoặc đơn hoặc dấu ngoặc kép nếu nó chứa bất kỳ khoảng trắng nào.



The screenshot shows a terminal window titled "student@openSUSE:~". The window contains the output of the "alias" command, which lists various shell aliases defined in the user's environment. The aliases include standard ones like "pushd", "popd", "cd", "ls", "grep", and "less", as well as more specific ones like "beep" (which outputs a beep sound) and "you" (which runs the "/sbin/yast2 online\_update" command). The terminal window has a standard Linux-style interface with a menu bar at the top.

```
student@openSUSE:~> alias
alias +='pushd .'
alias -='popd'
alias ..='cd ..'
alias ...='cd ../../'
alias beep='echo -en "\007"'
alias cd..='cd ..'
alias dir='ls -l'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -alF'
alias la='ls -la'
alias ll='ls -l'
alias ls='_ls'
alias ls-l='ls -l'
alias md='mkdir -p'
alias o='less'
alias rd='rmdir'
alias rehash='hash -r'
alias umount='echo "Error: Try the command: umount" 1>&2; false'
alias you='if test "$EUID" = 0 ; then /sbin/yast2 online_update ; else su - -c "/sbin/yast2 online_update" ; fi'
student@openSUSE:~>
```

## 6.2 Căn bản về người dùng và nhóm người dùng

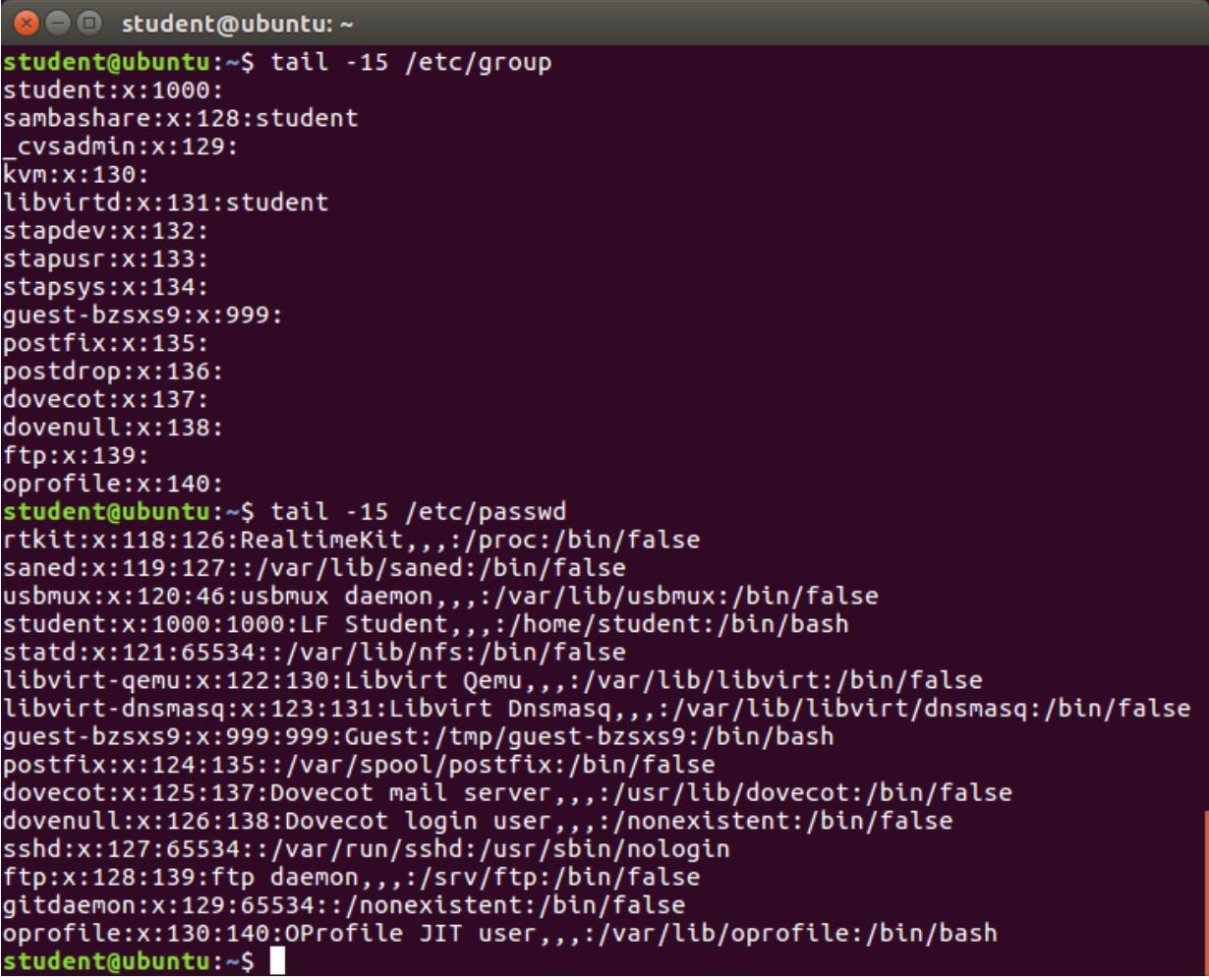
Tất cả người dùng Linux được gán một ID người dùng duy nhất (uid), đây chỉ là một số nguyên; người dùng bình thường bắt đầu với uid từ 1000 trở lên.

Linux sử dụng các nhóm để tổ chức người dùng. Nhóm là tập hợp các tài khoản với các quyền được chia sẻ nhất định. Kiểm soát thành viên nhóm được quản lý thông qua tệp /etc/nجموعات, trong đó hiển thị danh sách các nhóm và thành viên của họ. Theo mặc định, mọi người dùng thuộc về một nhóm mặc định hoặc chính. Khi người dùng đăng nhập,

thành viên nhóm được đặt cho nhóm chính của họ và tất cả các thành viên được hưởng cùng mức truy cập và đặc quyền. Quyền trên các tập tin và thư mục khác nhau có thể được sửa đổi ở cấp độ nhóm.

Người dùng cũng có một hoặc nhiều ID nhóm (gid), bao gồm một ID mặc định giống với ID người dùng. Những con số này được liên kết với tên thông qua các tập tin / etc / passwd và / etc / group. Các nhóm được sử dụng để thiết lập một nhóm người dùng có lợi ích chung cho các mục đích về quyền truy cập, đặc quyền và cân nhắc bảo mật. Quyền truy cập vào tệp (và thiết bị) được cấp trên cơ sở người dùng và nhóm mà họ thuộc về.

Ví dụ: / etc / passwd có thể chứa george: x: 1002: 1002: George Metesky: / home / george: / bin / bash và / etc / group có thể chứa george: x: 1002.



```
student@ubuntu:~$ tail -15 /etc/group
student:x:1000:
sambashare:x:128:student
_cvsadmin:x:129:
kvm:x:130:
libvirtd:x:131:student
stapdev:x:132:
stapusr:x:133:
stapsys:x:134:
guest-bzsxs9:x:999:
postfix:x:135:
postdrop:x:136:
dovecot:x:137:
dovenull:x:138:
ftp:x:139:
oprofile:x:140:
student@ubuntu:~$ tail -15 /etc/passwd
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127::/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
student:x:1000:1000:LF Student,,,:/home/student:/bin/bash
statd:x:121:65534::/var/lib/nfs:/bin/false
libvirt-qemu:x:122:130:Libvirt Qemu,,,:/var/lib/libvirt:/bin/false
libvirt-dnsmasq:x:123:131:Libvirt Dnsmasq,,,:/var/lib/libvirt/dnsmasq:/bin/false
guest-bzsxs9:x:999:999:Guest:/tmp/guest-bzsxs9:/bin/bash
postfix:x:124:135::/var/spool/postfix:/bin/false
dovecot:x:125:137:Dovecot mail server,,,:/usr/lib/dovecot:/bin/false
dovenull:x:126:138:Dovecot login user,,,:/nonexistent:/bin/false
sshd:x:127:65534::/var/run/sshd:/usr/sbin/nologin
ftp:x:128:139:ftp daemon,,,:/srv/ftp:/bin/false
gitdaemon:x:129:65534::/nonexistent:/bin/false
oprofile:x:130:140:OProfile JIT user,,,:/var/lib/oprofile:/bin/bash
student@ubuntu:~$
```

## Thêm và loại bỏ người dùng

Các bản phân phối có giao diện đồ họa đơn giản để tạo và xóa người dùng và nhóm và thao túng thành viên nhóm. Tuy nhiên, thường hữu ích khi thực hiện nó từ dòng lệnh hoặc

tù trong các tập lệnh shell. Chỉ người dùng root mới có thể thêm và xóa người dùng và nhóm.

Thêm người dùng mới được thực hiện với useradd và xóa người dùng hiện tại được thực hiện với userdel. Ở dạng đơn giản nhất, một tài khoản cho người dùng mới bjmoose sẽ được thực hiện với:

```
$ sudo useradd bjmoose
```

```
File Edit View Search Terminal Help
c7:/home/coop>sudo useradd -s /bin/csh -m -k /etc/skel -c "Bullwinkle J Moose" bmoose
c7:/home/coop>ls -la /home/bmoose
ls: cannot open directory /home/bmoose: Permission denied
c7:/home/coop>sudo ls -la /home/bmoose
total 28
drwx----- 3 bmoose bmoose 4096 Dec 14 09:57 .
drwxr-xr-x  3 root    root   4096 Dec 14 09:57 ..
-rw-r--r--  1 bmoose bmoose   18 Oct 17 05:05 .bash_logout
-rw-r--r--  1 bmoose bmoose  193 Oct 17 05:05 .bash_profile
-rw-r--r--  1 bmoose bmoose  231 Oct 17 05:05 .bashrc
-rw-r--r--  1 bmoose bmoose  334 Oct  7 2015 .emacs
drwxr-xr-x  4 bmoose bmoose 4096 Jan 26 2014 .mozilla
c7:/home/coop>sudo passwd bmoose
Changing password for user bmoose.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
c7:/home/coop>su bmoose
Password:
Attempting to create directory /home/bmoose/perl5
[bmoose@c7 coop]$ exit
exit
c7:/home/coop>sudo userdel -r bmoose
c7:/home/coop>ls /home
coop  linux1
c7:/home/coop>
```

Lưu ý rằng đối với openSUSE, useradd không có trong PATH của người dùng thông thường, vì vậy lệnh nên là:

```
$ sudo /usr/sbin/useradd bjmoose
```

theo mặc định, đặt thư mục chính thành /home/bjmoose, điền vào đó một số tệp cơ bản (được sao chép từ /etc/skel) và thêm một dòng vào /etc/passwd, chẳng hạn như:

```
bjmoose: x: 1002: 1002 ::/nhà/bjmoose: /bin/bash
```

và đặt shell mặc định thành / bin / bash. Xóa tài khoản người dùng cũng dễ như gõ userdel bjmoose. Tuy nhiên, điều này sẽ giữ nguyên thư mục / home / bjmoose. Điều này có thể hữu ích nếu nó là bất hoạt tạm thời. Để xóa thư mục chính trong khi xóa tài khoản, người ta cần sử dụng tùy chọn -r cho userdel.

Nhập id không có đối số cung cấp thông tin về người dùng hiện tại, như trong:

```
$ id
```

uid = 1002 (bjmoose) gid = 1002 (bjmoose) nhóm = 106 (cầu chì), 1002 (bjmoose)

Nếu được đặt tên của người dùng khác làm đối số, id sẽ báo cáo thông tin về người dùng khác đó.

### Thêm và loại bỏ nhóm người dùng

Thêm một nhóm mới được thực hiện với groupadd:

```
$ sudo /usr/sbin/groupadd một nhóm mới
```

Nhóm có thể được gỡ bỏ bằng:

```
$ sudo /usr/sbin/groupdel một nhóm mới
```

Thêm người dùng vào một nhóm đã tồn tại được thực hiện với usermod. Ví dụ: trước tiên bạn sẽ xem nhóm người dùng đã thuộc về nhóm nào:

```
$ group rjsquirrel
```

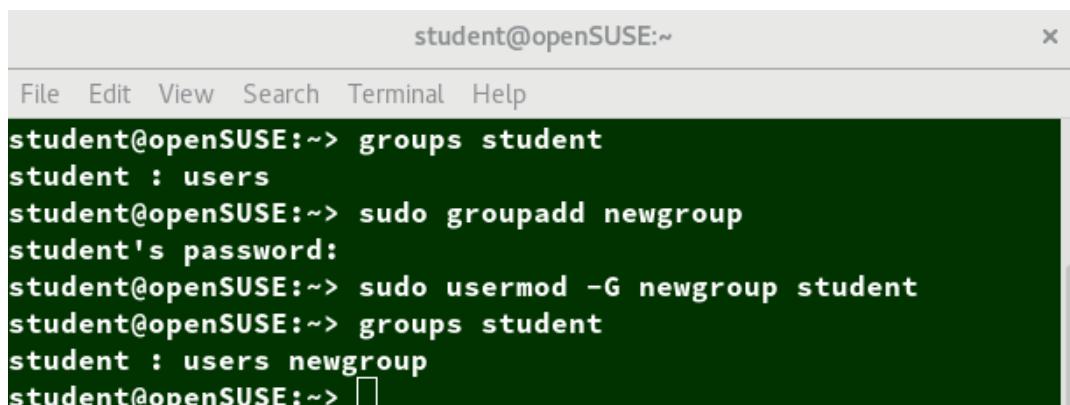
bjmoose: rjsquirrel

và sau đó thêm nhóm mới:

```
$ sudo /usr/sbin/usermod -a -G một nhóm mới rjsquirrel
```

```
$ group rjsquirrel
```

jsquirrel: rjsquirrel một nhóm mới



The screenshot shows a terminal window titled "student@openSUSE:~". The window has a menu bar with File, Edit, View, Search, Terminal, and Help. The terminal content is as follows:

```
student@openSUSE:~> groups student
student : users
student@openSUSE:~> sudo groupadd newgroup
student's password:
student@openSUSE:~> sudo usermod -G newgroup student
student@openSUSE:~> groups student
student : users newgroup
student@openSUSE:~>
```

Những tiện ích cập nhật /etc/group khi cần thiết. Đảm bảo sử dụng tùy chọn -a, để chép thêm, để tránh xóa các nhóm đã tồn tại. groupmod có thể được sử dụng để thay đổi các thuộc tính nhóm, chẳng hạn như ID nhóm (gid) với tùy chọn -g hoặc tên của nó với tùy chọn -n

Việc xóa một người dùng khỏi nhóm có phần phức tạp hơn. Tùy chọn -G cho usermod phải đưa ra một danh sách đầy đủ các nhóm. Vì vậy, nếu bạn làm:

```
$ sudo /usr/sbin/usermod -G rjsquirrel rjsquirrel
```

```
$ nhóm rjsquirrel
```

```
rjsquirrel: rjsquirrel
```

chỉ nhóm rjsquirrel sẽ còn lại.

### 6.3 Tài khoản root

Tài khoản root rất mạnh và có toàn quyền truy cập vào hệ thống. Các hệ điều hành khác thường gọi đây là tài khoản quản trị viên; trong Linux, nó thường được gọi là tài khoản superuser. Bạn phải cực kỳ thận trọng trước khi cấp quyền truy cập root đầy đủ cho người dùng; nó hiếm khi, nếu có, hợp lý. Các cuộc tấn công bên ngoài thường bao gồm các thủ thuật được sử dụng để nâng lên tài khoản root.

Tuy nhiên, bạn có thể sử dụng tính năng sudo để gán các đặc quyền hạn chế hơn cho tài khoản người dùng:

Chỉ trên cơ sở tạm thời

Chỉ cho một tập hợp con cụ thể của lệnh.

### Lệnh su và sudo

Khi gán đặc quyền nâng cao, bạn có thể sử dụng lệnh su (chuyển đổi hoặc người dùng thay thế) để khởi chạy shell mới chạy với tư cách người dùng khác (bạn phải nhập mật khẩu của người dùng mà bạn đang trở thành). Thông thường, người dùng khác này là root và shell mới cho phép sử dụng các đặc quyền nâng cao cho đến khi thoát. Hầu như luôn luôn là một thực hành xấu (nguy hiểm cho cả bảo mật và ổn định) để sử dụng su để trở thành root. Lỗi kết quả có thể bao gồm xóa các tệp quan trọng khỏi hệ thống và vi phạm bảo mật.

Cấp đặc quyền bằng sudo ít nguy hiểm hơn và được ưu tiên. Theo mặc định, sudo phải được bật trên cơ sở mỗi người dùng. Tuy nhiên, một số bản phân phối (như Ubuntu) cho phép mặc định cho ít nhất một người dùng chính hoặc cung cấp tùy chọn này dưới dạng tùy chọn cài đặt.

su và sau đó được nhắc nhập mật khẩu root.

Để thực thi chỉ một lệnh với đặc quyền root, gõ sudo <lệnh>. Khi lệnh hoàn thành, bạn sẽ trở lại là một người dùng không có đặc quyền bình thường.

Các tệp cấu hình sudo được lưu trữ trong tệp / etc / sudoers và trong thư mục /etc/sudoers.d/. Theo mặc định, thư mục sudoers.d trống.

```
student@openSUSE:~> ls -la /root
ls: cannot open directory '/root': Permission denied
student@openSUSE:~> sudo ls -la /root
student's password:
total 12
drwx----- 1 root root 144 Dec 14 10:28 .
drwxr-xr-x 1 root root 156 Dec 14 09:37 ..
-rw----- 1 root root 1269 Dec 14 10:28 .bash_history
drwxr-xr-x 1 root root 0 Oct 7 10:51 bin
drwx----- 1 root root 0 Dec 14 10:03 .cache
drwxr-xr-x 1 root root 20 Dec 14 10:26 .config
drwx----- 1 root root 22 Dec 14 09:49 .dbus
drwx----- 1 root root 0 Dec 14 10:26 .emacs.d
drwx----- 1 root root 0 Oct 7 10:51 .gnupg
drwxr-xr-x 1 root root 46 Dec 14 03:31 inst-sys
-rw----- 1 root root 41 Dec 14 10:09 .lessht
-rw----- 1 root root 1098 Dec 14 10:24 .viminfo
student@openSUSE:~>
```

## Bài tập LAB – Thiết lập bí danh (aliases)

Việc gõ các lệnh dài và tên tệp nhiều lần trở nên khá tẻ nhạt và dẫn đến rất nhiều lỗi nhỏ, chẳng hạn như lỗi chính tả.

Việc triển khai các bí danh cho phép chúng ta xác định các phím tắt để giảm bớt nỗi đau của tất cả các kiểu gõ này.

Giả sử bạn là thành viên của nhóm dự án làm việc trong một thư mục chung, chung cho dự án của bạn. Thư mục này được đặt trong / home / staff / RandD / dự án / projectX / src.

Khi bạn đang làm việc trên Project X, bạn thường cần tạo và sửa đổi các tệp của mình trong thư mục này. Không mất quá nhiều thời gian trước khi nhập: cd / home / staff / RandD / dự án / projectX / src trở nên tẻ nhạt.

Xác định và sử dụng một bí danh có tên "projx" để thực hiện lệnh cd ở trên cho bạn.

## 6.4 Các biến môi trường

Biến môi trường là các đại lượng có các giá trị cụ thể có thể được sử dụng bởi shell lệnh, chẳng hạn như bash hoặc các tiện ích và ứng dụng khác. Một số biến môi trường được cung cấp các giá trị đặt trước bởi hệ thống (thường có thể bị ghi đè), trong khi các biến khác được đặt trực tiếp bởi người dùng, tại dòng lệnh hoặc trong khi khởi động và các tập lệnh khác.

Một biến môi trường thực sự chỉ là một chuỗi ký tự chứa thông tin được sử dụng bởi một hoặc nhiều ứng dụng. Có một số cách để xem các giá trị của các biến môi trường hiện được đặt; người ta có thể gõ set, env hoặc export. Tùy thuộc vào trạng thái hệ thống của bạn, bộ có thể in ra nhiều dòng hơn hai phương thức còn lại.

```
File Edit View Search Terminal Help
c7:/tmp>env | head -5
XDG_VTNR=1
XDG_SESSION_ID=1
HOSTNAME=c7
IMSETTINGS_INTEGRATE_DESKTOP=yes
TERM=xterm-256color
c7:/tmp>
c7:/tmp>set | head -5
ABRT_DEBUG_LOG=/dev/null
BASH=/usr/bin/bash
BASHOPTS=checkwinsize:cmdhist:expand_aliases:extglob:extquote:force_fignore:histappend:interactive_comments:progcomp:promptvars:sourcepath
BASH_ALIASES=()
BASH_ARGC=()
c7:/tmp>
c7:/tmp>export | head -5
declare -x BITS="64"
declare -x CCACHE_COMPRESS="1"
declare -x CCACHE_DIR="/tmp/.ccache"
declare -x CCACHE_TEMPDIR="/tmp/.ccache"
declare -x CCACHE_UMASK="002"
c7:/tmp>
```

Theo mặc định, các biến được tạo trong một tập lệnh chỉ có sẵn cho trình bao hiện tại; các tiến trình con (shell con) sẽ không có quyền truy cập vào các giá trị đã được đặt hoặc sửa đổi. Cho phép các tiến trình con để xem các giá trị yêu cầu sử dụng lệnh xuất.

Hiển thị giá trị của một biến cụ thể echo \$ SHELL

Xuất một giá trị biến mới xuất VARIABLE = value (hoặc VARIABLE = value; export VARIABLE)

Thêm một biến vĩnh viễn

Chỉnh sửa ~ / .bashrc và thêm dòng xuất VARIABLE = value

Nhập nguồn ~ / .bashrc hoặc chỉ. ~ / .bashrc (dấu chấm ~ / .bashrc); hoặc chỉ bắt đầu một shell mới bằng cách gõ bash

Bạn cũng có thể đặt các biến môi trường được cung cấp dưới dạng một lần bắn thành lệnh như trong:

```
$ SDIRS = $_0 * KROOT = / lib / mô-đun / $(uname -r) / xây dựng mô-đun_install  
cung cấp các giá trị của các biến môi trường SDIRS và KROOT cho lệnh tạo mô-  
đun_install.
```

## Biến môi trường HOME

HOME là một biến môi trường đại diện cho thư mục home (hoặc đăng nhập) của người dùng. cd không có đối số sẽ thay đổi thư mục làm việc hiện tại thành giá trị của HOME. Lưu ý ký tự dấu ngã (~) thường được sử dụng làm chữ viết tắt cho \$ HOME. Do đó, cd \$ HOME và cd ~ là các câu lệnh hoàn toàn tương đương.

```
$ echo $ HOME
```

```
/home/my
```

```
$ cd temp
```

```
student@ubuntu: ~$ echo $HOME  
/home/student  
student@ubuntu:~$ cd /usr/bin  
student@ubuntu:/usr/bin$ pwd  
/usr/bin  
student@ubuntu:/usr/bin$ cd $HOME  
student@ubuntu:~$ pwd  
/home/student  
student@ubuntu:~$
```

Hiển thị giá trị của biến môi trường HOME, sau đó thay đổi thư mục (cd) thành / bin.

```
$ pwd
```

```
/temp
```

Chúng ta ở đâu? Sử dụng thư mục làm việc in (hoặc hiện tại) (pwd) để tìm hiểu. Như mong đợi /bin.

```
$ cd Thay đổi thư mục mà không có đối số ...
```

```
$ pwd
```

```
/home/my
```

... Đưa chúng ta trở về TRANG CHỦ, như bây giờ bạn có thể thấy.

Ảnh chụp màn hình chứng minh điều này.

## Biến môi trường PATH

PATH là một danh sách các thư mục (đường dẫn) được quét khi lệnh được đưa ra để tìm chương trình hoặc tập lệnh thích hợp để chạy. Mỗi thư mục trong đường dẫn được phân tách bằng dấu hai chấm (:). Tên thư mục null (trống) (hoặc ./) cho biết thư mục hiện tại tại bất kỳ thời điểm nào.

```
: path1: path2
```

```
path1 :: path2
```

Trong ví dụ: path1: path2, có một thư mục null trước dấu hai chấm đầu tiên (:). Tương tự, đối với path1 :: path2 có một thư mục null giữa path1 và path2.

Để tiện tố một thư mục bin riêng vào đường dẫn của bạn:

```
$ export PATH = $ HOME / thùng: $ PATH
```

```
$ echo $ PATH
```

```
/ home / student / bin: / usr / local / bin: / usr / bin: / bin / usr
```

```
student@ubuntu:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
student@ubuntu:~$ OLDPATH=$PATH
student@ubuntu:~$ PATH=$PATH:/opt/some_application
student@ubuntu:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/some_application
student@ubuntu:~$ PATH=$OLDPATH
student@ubuntu:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
student@ubuntu:~$
```

## Biến môi trường SHELL

Biến môi trường SHELL trả đến trình vỏ lệnh mặc định của người dùng (chương trình đang xử lý bất cứ điều gì bạn gõ trong cửa sổ lệnh, thường là bash) và chứa tên đường dẫn đầy đủ đến trình bao:

```
$ echo $ SHELL
```

```
/bin/bash
```

```
$
```

## Biến môi trường PS1 và dấu nháy trên dòng lệnh

Tuyên bố nháy nhở (PS) được sử dụng để tùy chỉnh chuỗi nháy của bạn trong cửa sổ đầu cuối để hiển thị thông tin bạn muốn.

PS1 là biến dấu nhắc chính điều khiển dấu nhắc dòng lệnh của bạn trông như thế nào. Các ký tự đặc biệt sau đây có thể được bao gồm trong PS1:

- \ u - Tên người dùng
- \ h - Tên máy chủ
- \ w - Thư mục làm việc hiện tại
- \! - Số lịch sử của lệnh này
- \ d - Ngày

Chúng phải được bao quanh trong dấu ngoặc đơn khi chúng được sử dụng, như trong ví dụ sau:

```
$ echo $ PS1  
$  
$ xuất PS1 = '\ u @ \ h: \ w '$  
student@example.com: ~ $ # nhắc mới  
student@example.com: ~ $  
Để hoàn nguyên các thay đổi:  
student@example.com: ~ $ xuất PS1 = '$'  
$
```

Một cách thực hành thậm chí tốt hơn là lưu lại dấu nhắc cũ trước rồi mới khôi phục, như trong:

```
$ OLD_PS1 = $ PS1  
thay đổi lời nhắc và cuối cùng thay đổi lại bằng:  
$ PS1 = $ OLD_PS1  
$
```

### **Bài tập LAB : Thêm /tmp vào đường dẫn PATH**

Tạo một tệp nhỏ / tmp / ls, chỉ chứa dòng:  
echo HELLO, đây là chương trình ls giả mạo.  
Sau đó, làm cho nó thực thi bằng cách làm:  
\$ chmod + x / tmp / ls

Nối / tmp vào đường dẫn của bạn, vì vậy nó chỉ được tìm kiếm sau khi đường dẫn thông thường của bạn được xem xét. Nhập ls và xem chương trình nào đang chạy: / bin / ls hoặc / tmp / ls?

Pre-Pend / tmp vào đường dẫn của bạn, vì vậy nó được tìm kiếm trước khi đường dẫn thông thường của bạn được xem xét. Một lần nữa, gõ ls và xem chương trình nào đang chạy: / bin / ls hoặc / tmp / ls?

Các cân nhắc bảo mật trong việc thay đổi đường dẫn theo cách này là gì?

### Bài tập LAB : Thay đổi dấu nhắc dòng lệnh

Thật tuyệt khi có thư mục làm việc hiện tại của bạn như là một phần của lời nhắc của bạn để một cái nhìn nhanh sẽ cung cấp cho bạn một số thông tin mà không cần gõ pwd mỗi lần.

Nếu bạn thường xuyên làm việc trên nhiều máy tính, đặc biệt nếu bạn nối mạng từ máy này sang máy khác bằng ssh, sẽ rất thuận tiện khi tên máy tính là một phần trong lời nhắc của bạn.

Đặt thư mục làm việc hiện tại của bạn trong dấu nhắc dòng lệnh của bạn.

Đặt tên máy tính (máy) của bạn trong lời nhắc của bạn.

Đặt cả thư mục hiện tại và tên máy tính của bạn trong lời nhắc của bạn.

Làm thế nào bạn có thể thực hiện điều này liên tục, để bắt cứ khi nào bạn bắt đầu một shell lệnh bash, đây là lời nhắc của bạn?

### 6.5 Gọi lại các lệnh đã thực thi

bash theo dõi các lệnh và câu lệnh đã nhập trước đó trong bộ đệm lịch sử. Bạn có thể nhớ lại các lệnh đã sử dụng trước đó chỉ bằng cách sử dụng các phím con trỏ Lên và Xuống. Để xem danh sách các lệnh đã thực hiện trước đó, bạn chỉ cần nhập lịch sử vào dòng lệnh.

Danh sách các lệnh được hiển thị với lệnh gần đây nhất xuất hiện cuối cùng trong danh sách. Thông tin này được lưu trữ trong ~ / .bash\_history. Nếu bạn có nhiều thiết bị đầu cuối mở, các lệnh được nhập trong mỗi phiên sẽ không được lưu cho đến khi phiên kết thúc.

A screenshot of a terminal window titled "student@debian: ~". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area displays the output of the command "history | tail -20". The output shows a list of 21 commands, each preceded by a number (199 to 219). The commands include various system and file operations like sudo poweroff, ifconfig, uname -a, ls /boot, apt-get remove, df -h, cd /var/cache, du -shc, sudo du -shc, cd /tmp, du -shc, sudo su, cd Pictures/, scp, sudo su, history, and history | tail -20. The terminal prompt "student@debian:~\$ " is visible at the bottom.

```
student@debian:~$ history | tail -20
199 sudo poweroff
200 sudo su
201 ifconfig
202 /sbin/ifconfig
203 uname -a
204 sudo su
205 ls /boot
206 sudo apt-get remove 4.9.0-5
207 df -h
208 cd /var/cache
209 du -shc
210 sudo du -shc
211 cd /tmp
212 du -shc
213 sudo su
214 cd Pictures/
215 scp coop@192.168.1.200:/usr/local/images/deat* .
216 sudo su
217 history
218 history | tail -20
student@debian:~$
```

## Sử dụng các biến môi trường lịch sử

Một số biến môi trường liên quan có thể được sử dụng để lấy thông tin về tệp lịch sử.

HISTFILE

Vị trí của tập tin lịch sử.

HISTFILESIZE

Số lượng dòng tối đa trong tệp lịch sử (mặc định 500).

HISTSIZE

Số lượng lệnh tối đa trong tệp lịch sử.

HISTCONTROL

Làm thế nào các lệnh được lưu trữ.

## HISTIGNORE

Những dòng lệnh nào có thể được lưu.

### Tìm và sử dụng các lệnh đã thực thi

#### Up/Down arrow keys:

Các phím mũi tên Lên / Xuống Duyệt qua danh sách các lệnh được thực hiện trước đó

**!! (Phát âm là bang-bang)** Thực hiện lệnh trước đó

**CTRL-R** Tìm kiếm các lệnh được sử dụng trước đó

Nếu bạn muốn gọi lại một lệnh trong danh sách lịch sử, nhưng không muốn nhấn phím mũi tên nhiều lần, bạn có thể nhấn CTRL-R để thực hiện tìm kiếm thông minh ngược.

Khi bạn bắt đầu nhập, tìm kiếm sẽ quay trở lại theo thứ tự ngược lại với lệnh đầu tiên khớp với các chữ cái bạn đã gõ. Bằng cách gõ các chữ cái liên tiếp hơn, bạn làm cho trận đấu ngày càng cụ thể hơn.

Sau đây là một ví dụ về cách bạn có thể sử dụng lệnh CTRL-R để tìm kiếm trong lịch sử lệnh:

\$ ^ R (Tất cả điều này xảy ra trên 1 dòng)

(đảo ngược i-search) 's': ngủ 1000 (Đã tìm kiếm 's'; khớp với "giác ngủ")

\$ ngủ 1000 (Nhấn Enter để thực thi lệnh đã tìm)

\$

### Bài tập LAB: Lịch sử thực thi các lệnh

Bạn đã bận rộn làm việc với máy trạm Linux của mình đủ lâu để gõ khoảng 100 lệnh trong một vỏ lệnh bash cụ thể.

Tại một số thời điểm trước đó, bạn đã sử dụng một lệnh mới, nhưng tên chính xác đã trượt tâm trí của bạn.

Hoặc có lẽ đó là một lệnh khá phức tạp, với một loạt các tùy chọn và đối số và bạn không muốn trải qua quá trình dễ bị lỗi để tìm ra cách gõ lại.

Làm thế nào để bạn xác định được lệnh là gì?

Khi bạn tìm thấy lệnh trong lịch sử của mình, làm thế nào để bạn dễ dàng đưa ra lệnh một lần nữa mà không phải gõ tất cả vào đâu nhỉ?

### 6.6 Quyền sở hữu tập tin và thư mục

Trong Linux và các hệ điều hành dựa trên UNIX khác, mọi tệp đều được liên kết với người dùng là chủ sở hữu. Mỗi tệp cũng được liên kết với một nhóm (tập hợp con của tất cả

người dùng) có mối quan tâm đến tệp và các quyền nhất định hoặc quyền: đọc, viết và thực thi.

Các lệnh tiện ích sau liên quan đến quyền sở hữu và quyền của người dùng và nhóm:

**chown** Được sử dụng để thay đổi quyền sở hữu của người dùng đối với tệp hoặc thư mục

**chgrp** Được sử dụng để thay đổi quyền sở hữu nhóm

**chmod** Được sử dụng để thay đổi các quyền trên tệp, có thể được thực hiện riêng cho chủ sở hữu, nhóm và phần còn lại của thế giới (thường được đặt tên là khác)

## Quyền truy cập tập tin và lệnh chmod

Các tệp có ba loại quyền: đọc (r), ghi (w), thực thi (x). Chúng thường được đại diện như trong rwx. Các quyền này ảnh hưởng đến ba nhóm chủ sở hữu: người dùng / chủ sở hữu (u), nhóm (g) và những người khác (o).

Kết quả là, bạn có ba nhóm ba quyền sau đây:

rwx: rwx: rwx

u: g: o

Có một số cách khác nhau để sử dụng chmod. Chẳng hạn, để cho phép chủ sở hữu và những người khác thực thi quyền và xóa quyền ghi nhóm:

\$ ls -l somefile

-rw-rw-r-- 1 sinh viên sinh viên 1601 Mar 9 15:04 somefile

\$ chmod uo + x, g-w somefile

\$ ls -l somefile

-rwxr - r-x 1 sinh viên sinh viên 1601 Mar 9 15:04 somefile

trong đó u là viết tắt của người dùng (chủ sở hữu), o là viết tắt của khác (thế giới) và g là viết tắt của nhóm.

Loại cú pháp này có thể khó gõ và nhớ, vì vậy người ta thường sử dụng một tốc ký cho phép bạn đặt tất cả các quyền trong một bước. Điều này được thực hiện với một thuật toán đơn giản và một chữ số đủ để chỉ định cả ba bit quyền cho mỗi thực thể. Chữ số này là tổng của:

4 nếu được phép đọc

2 nếu được phép viết

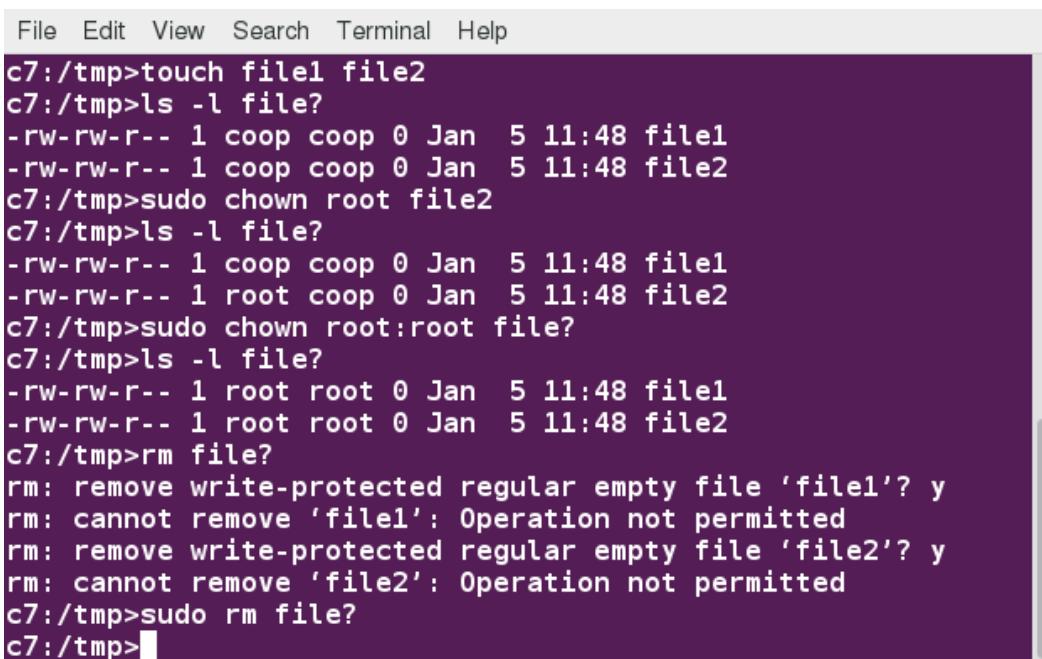
1 nếu quyền thực thi là mong muốn.

Do đó, 7 có nghĩa là đọc / ghi / thực thi, 6 có nghĩa là đọc / ghi và 5 có nghĩa là đọc / thực thi.

Khi bạn áp dụng điều này cho lệnh chmod, bạn phải đưa ra ba chữ số cho mỗi mức độ tự do, chẳng hạn như trong:

```
$ chmod 755  
$ ls -l somefile  
-rwxr-xr-x 1 sinh viên 1601 Mar 9 15:04 somefile
```

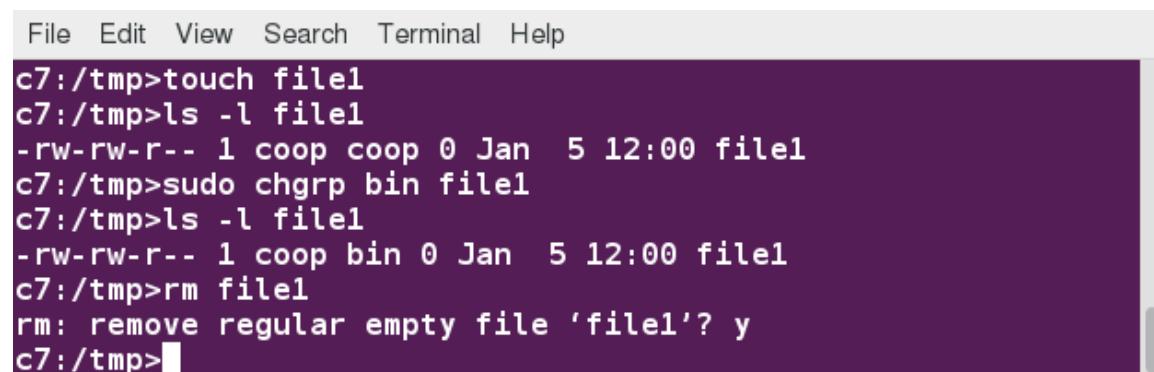
Chúng ta hãy xem một ví dụ về việc thay đổi quyền sở hữu tệp bằng cách sử dụng chown, như được hiển thị trong ảnh chụp màn hình ở bên phải. Đầu tiên, chúng tôi tạo hai tệp trống bằng cách sử dụng cảm ứng.



```
File Edit View Search Terminal Help  
c7:/tmp>touch file1 file2  
c7:/tmp>ls -l file?  
-rw-rw-r-- 1 coop coop 0 Jan  5 11:48 file1  
-rw-rw-r-- 1 coop coop 0 Jan  5 11:48 file2  
c7:/tmp>sudo chown root file2  
c7:/tmp>ls -l file?  
-rw-rw-r-- 1 coop coop 0 Jan  5 11:48 file1  
-rw-rw-r-- 1 root coop 0 Jan  5 11:48 file2  
c7:/tmp>sudo chown root:root file?  
c7:/tmp>ls -l file?  
-rw-rw-r-- 1 root root 0 Jan  5 11:48 file1  
-rw-rw-r-- 1 root root 0 Jan  5 11:48 file2  
c7:/tmp>rm file?  
rm: remove write-protected regular empty file 'file1'? y  
rm: cannot remove 'file1': Operation not permitted  
rm: remove write-protected regular empty file 'file2'? y  
rm: cannot remove 'file2': Operation not permitted  
c7:/tmp>sudo rm file?  
c7:/tmp>
```

Lưu ý rằng nó yêu cầu sudo để thay đổi chủ sở hữu của tập tin 2 thành root.

Lệnh chown thứ hai thay đổi cả chủ sở hữu và nhóm cùng một lúc!



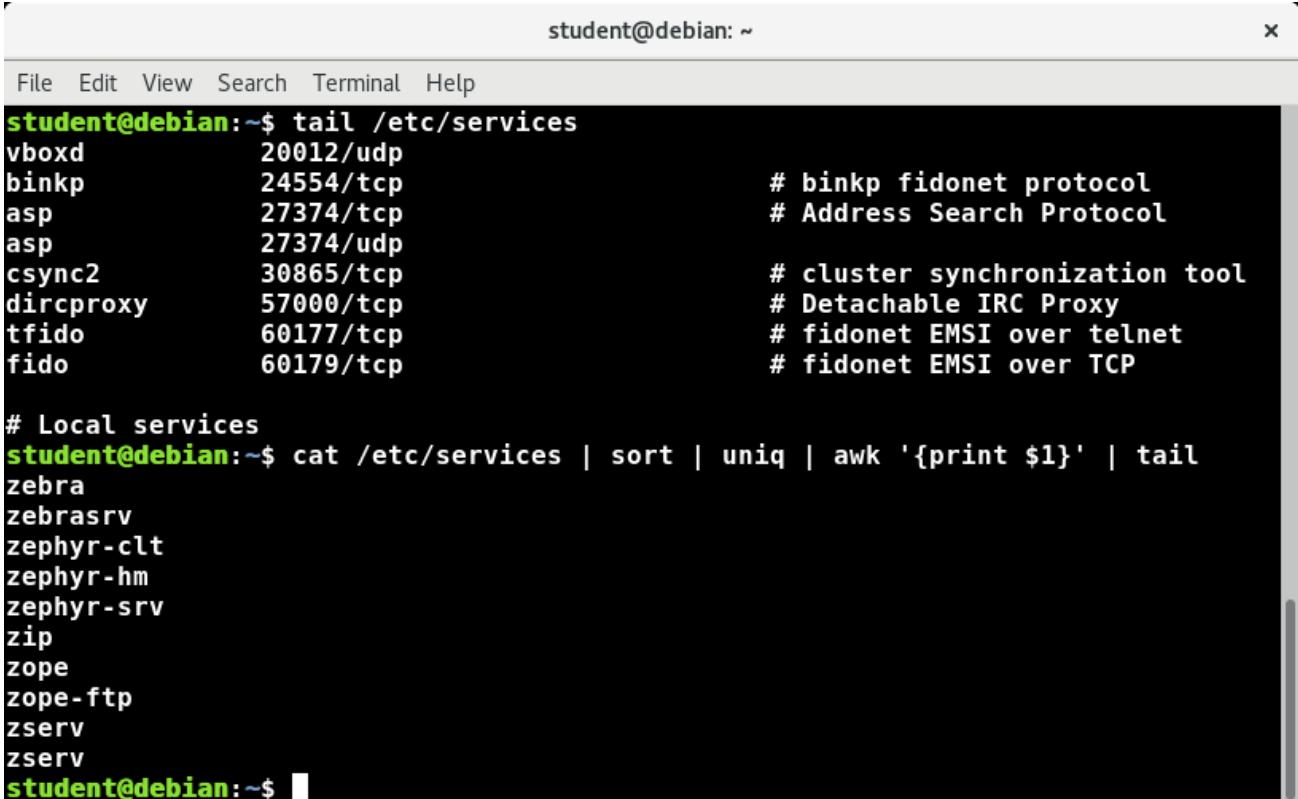
```
File Edit View Search Terminal Help  
c7:/tmp>touch file1  
c7:/tmp>ls -l file1  
-rw-rw-r-- 1 coop coop 0 Jan  5 12:00 file1  
c7:/tmp>sudo chgrp bin file1  
c7:/tmp>ls -l file1  
-rw-rw-r-- 1 coop bin 0 Jan  5 12:00 file1  
c7:/tmp>rm file1  
rm: remove regular empty file 'file1'? y  
c7:/tmp>
```

Cuối cùng, chỉ có siêu người dùng có thể loại bỏ các tập tin.

Bây giờ, hãy để Lôi xem một ví dụ về việc thay đổi quyền sở hữu nhóm bằng chgrp:

## 6.7 Các lệnh tiện ích làm việc với văn bản và dòng lệnh

Bất kể vai trò của bạn với Linux (quản trị viên hệ thống, nhà phát triển hoặc người dùng), bạn thường cần duyệt qua và phân tích các tệp văn bản và / hoặc trích xuất dữ liệu từ chúng. Đây là các thao tác thao tác tập tin. Vì vậy, điều cần thiết là người dùng Linux trở nên lão luyện trong việc thực hiện các hoạt động nhất định trên các tệp.



```
student@debian: ~
student@debian:~$ tail /etc/services
vboxd          20012/udp
binkp          24554/tcp      # binkp fidonet protocol
asp            27374/tcp      # Address Search Protocol
asp            27374/udp
csync2         30865/tcp      # cluster synchronization tool
dircproxy       57000/tcp      # Detachable IRC Proxy
tfido          60177/tcp      # fidonet EMSI over telnet
fido           60179/tcp      # fidonet EMSI over TCP

# Local services
student@debian:~$ cat /etc/services | sort | uniq | awk '{print $1}' | tail
zebra
zebrasrv
zephyr-clt
zephyr-hm
zephyr-srv
zip
zope
zope-ftp
zserv
zserv
student@debian:~$
```

Hầu hết thời gian, thao tác tệp như vậy được thực hiện tại dòng lệnh, cho phép người dùng thực hiện các tác vụ hiệu quả hơn so với khi sử dụng GUI. Hơn nữa, dòng lệnh phù hợp hơn để tự động hóa các tác vụ thường được thực hiện.

Thật vậy, các quản trị viên hệ thống có kinh nghiệm viết các tập lệnh tùy chỉnh để thực hiện các tác vụ lặp đi lặp lại như vậy, được chuẩn hóa cho từng môi trường cụ thể. Chúng tôi sẽ thảo luận về kịch bản như vậy sau này chi tiết hơn.

Trong phần này, chúng tôi sẽ tập trung vào tệp dòng lệnh và các tiện ích liên quan đến thao tác văn bản.

Hiển thị và nối vào nội dung tệp tin bằng cách sử dụng **cat** và **echo**.

Chỉnh sửa và in nội dung tệp tin bằng **sed** và **awk**.

Tìm kiếm các mẫu bằng **grep**.

Sử dụng nhiều tiện ích khác để thao tác tập tin và văn bản.

**cat** là viết tắt của concatenate và là một trong những tiện ích dòng lệnh Linux được sử dụng thường xuyên nhất. Nó thường được sử dụng để đọc và in các tập tin, cũng như chỉ đơn giản là xem nội dung tập tin. Để xem tệp, sử dụng lệnh sau:

```
$ cat <tên tệp>
```

Ví dụ: cat readme.txt sẽ hiển thị nội dung của readme.txt trên thiết bị đầu cuối. Tuy nhiên, mục đích chính của mèo thường là kết hợp (ghép) nhiều tệp lại với nhau. Bạn có thể thực hiện các hành động được liệt kê trong bảng bằng cách sử dụng mèo.

Lệnh tac (cat đánh vần ngược) in các dòng của tệp theo thứ tự ngược lại. Mỗi dòng vẫn giữ nguyên, nhưng thứ tự của các dòng được đảo ngược. Cú pháp của tac hoàn toàn giống với cú pháp của mèo, như trong:

```
$ tac file
```

```
$ tac file1 file2> newfile
```

cat file1 file2 Ghép nhiều tệp và hiển thị đầu ra; tức là toàn bộ nội dung của tệp đầu tiên được theo sau bởi toàn bộ tệp thứ hai

```
cat file1 file2> newfile Kết hợp nhiều tệp và lưu kết quả đầu ra vào một tệp mới
```

```
cat file >> tệp hiện có Nối một tệp vào cuối tệp hiện có
```

```
cat> file Bất kỳ dòng tiếp theo nào được nhập sẽ đi vào tệp, cho đến khi CTRL-D được nhập
```

```
cat>> file
```

cat có thể được sử dụng để đọc từ đầu vào tiêu chuẩn (như cửa sổ đầu cuối) nếu không có tệp nào được chỉ định. Bạn có thể sử dụng toán tử > để tạo và thêm các dòng vào một tệp mới và toán tử >> để nối các dòng (hoặc tệp) vào một tệp hiện có. Chúng tôi đã đề cập đến điều này khi nói về cách tạo tập tin mà không cần trình chỉnh sửa.

Để tạo một tệp mới, tại dấu nhắc lệnh gõ cat> <tên tệp> và nhấn phím Enter.

Lệnh này tạo một tệp mới và chờ người dùng chỉnh sửa / nhập văn bản. Sau khi bạn nhập xong văn bản được yêu cầu, nhấn CTRL-D ở đầu dòng tiếp theo để lưu và thoát chỉnh sửa.

Một cách khác để tạo tệp tại thiết bị đầu cuối là cat> <tên tệp> << EOF. Một tệp mới được tạo và bạn có thể nhập đầu vào cần thiết. Để thoát, nhập EOF ở đầu một dòng.

Lưu ý rằng EOF là trường hợp nhạy cảm. Người ta cũng có thể sử dụng một từ khác, chẳng hạn như STOP.

```
student@FC-25:~  
File Edit View Search Terminal Help  
[student@FC-25 ~]$ cat << EOF > somefile  
> Anything I type will go in the file  
> This is an environment variable: $HOME  
> This is a substitution expression: $(echo Hello Student)  
> This is the last line, have fun!  
> EOF  
[student@FC-25 ~]$ cat somefile  
Anything I type will go in the file  
This is an environment variable: /home/student  
This is a substitution expression: Hello Student  
This is the last line, have fun!  
[student@FC-25 ~]$  
[student@FC-25 ~]$ █
```

echo chỉ đơn giản là hiển thị (tiếng vang) văn bản. Nó được sử dụng đơn giản, như trong:

\$ echo chuỗi

echo có thể được sử dụng để hiển thị một chuỗi trên đầu ra tiêu chuẩn (nghĩa là thiết bị đầu cuối) hoặc để đặt vào một tệp mới (sử dụng toán tử >) hoặc nối vào một tệp đã có sẵn (sử dụng toán tử >>).

Tùy chọn tích hợp, cùng với các công tắc sau, được sử dụng để cho phép các chuỗi ký tự đặc biệt, chẳng hạn như ký tự dòng mới hoặc tab ngang.

\n đại diện cho dòng mới

\t đại diện cho tab ngang.

echo đặc biệt hữu ích để xem các giá trị của các biến môi trường (các biến shell tích hợp). Ví dụ: echo \$ USERNAME sẽ in tên của người dùng đã đăng nhập vào thiết bị đầu cuối hiện tại.

Bảng sau liệt kê các lệnh echo và cách sử dụng của chúng:

chuỗi echo> newfile Chuỗi được chỉ định được đặt trong một tệp mới

chuỗi echo >> currentfile Chuỗi được chỉ định được nối vào cuối tệp đã tồn tại

echo \$ biến Nội dung của biến môi trường đã chỉ định được hiển thị

### Làm việc với các tập tin kích thước lớn

Quản trị viên hệ thống cần làm việc với các tệp cấu hình, tệp văn bản, tệp tài liệu và tệp nhật ký. Một số tệp này có thể lớn hoặc trở nên khá lớn khi chúng tích lũy dữ liệu theo thời gian. Những tập tin này sẽ yêu cầu cả xem và cập nhật hành chính. Trong phần này, bạn sẽ học cách quản lý các tệp lớn như vậy.

Ví dụ: hệ thống ngân hàng có thể duy trì một tệp nhật ký lớn đơn giản để ghi lại chi tiết của tất cả các giao dịch ATM của một ngày. Do một cuộc tấn công bảo mật hoặc trực trặc, quản trị viên có thể buộc phải kiểm tra một số dữ liệu bằng cách điều hướng trong tệp. Trong các trường hợp như vậy, trực tiếp mở tệp trong trình chỉnh sửa sẽ gây ra sự cố, do sử dụng bộ nhớ cao, vì trình chỉnh sửa thường sẽ cố gắng đọc toàn bộ tệp vào bộ nhớ trước. Tuy nhiên, người ta có thể sử dụng ít hơn để xem nội dung của một tệp lớn như vậy, cuộn lên xuống theo từng trang mà không cần hệ thống phải đặt toàn bộ tệp vào bộ nhớ trước khi bắt đầu. Điều này nhanh hơn nhiều so với sử dụng một trình soạn thảo văn bản.

Xem một số tệp có thể được thực hiện bằng cách nhập một trong hai lệnh sau:

\$ ít hơn một chút

\$ cat somefile | ít hơn

Theo mặc định, các trang man được gửi thông qua lệnh ít hơn. Bạn có thể đã gặp tiện ích cũ hơn có cùng chức năng cơ bản nhưng khả năng ít hơn: nghĩa là ít hơn là nhiều hơn!

## head

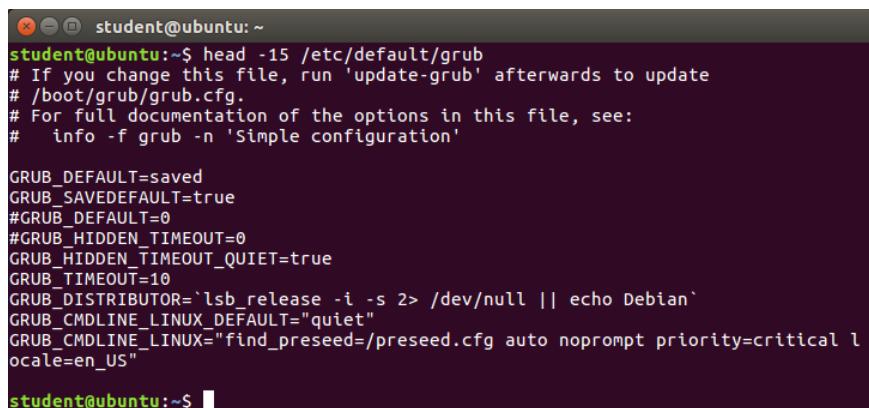
head đọc một vài dòng đầu tiên của mỗi tệp được đặt tên (10 theo mặc định) và hiển thị nó trên đầu ra tiêu chuẩn. Bạn có thể đưa ra một số dòng khác nhau trong một tùy chọn.

Ví dụ: nếu bạn muốn in 5 dòng đầu tiên từ grub.cfg, hãy sử dụng lệnh sau:

**\$ head -n 5 grub.cfg**

Bạn cũng có thể chỉ cần nói

**\$ head -5 grub.cfg**



```
student@ubuntu:~$ head -15 /etc/default/grub
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=saved
GRUB_SAVEDEFAULT=true
#GRUB_DEFAULT=0
#GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX="find_preseed=/preseed.cfg auto noprompt priority=critical locale=en_US"

student@ubuntu:~$
```

## tail

tail in một vài dòng cuối cùng của mỗi tệp được đặt tên và hiển thị nó trên đầu ra tiêu chuẩn. Theo mặc định, nó sẽ hiển thị 10 dòng cuối cùng. Bạn có thể đưa ra một số dòng khác nhau dưới dạng tùy chọn. tail đặc biệt hữu ích khi bạn đang khắc phục sự cố bất kỳ vấn đề nào bằng cách sử dụng tệp nhật ký, vì bạn có thể muốn xem các dòng đầu ra gần đây nhất.

Ví dụ: để hiển thị 15 dòng cuối cùng của somefile.log, hãy sử dụng lệnh sau:

```
$ tail -n 15 somefile.log
```

\* Bạn cũng có thể chỉ cần nói tail -15 somefile.log.

Để liên tục theo dõi đầu ra mới trong một tệp nhật ký đang phát triển:

```
$ tail -f somefile.log
```

Lệnh này sẽ liên tục hiển thị bất kỳ dòng đầu ra mới nào trong atmtrans.log ngay khi chúng xuất hiện. Do đó, nó cho phép bạn theo dõi mọi hoạt động hiện tại đang được báo cáo và ghi lại.

### Xem nội dung các tập tin văn bản nén

Khi làm việc với các tệp nén, nhiều lệnh tiêu chuẩn không thể được sử dụng trực tiếp. Đối với nhiều chương trình xử lý tệp và văn bản thường được sử dụng, cũng có một phiên bản được thiết kế đặc biệt để làm việc trực tiếp với các tệp nén. Các tiện ích liên quan này có chữ "z" được đặt trước tên của chúng. Ví dụ: chúng tôi có các chương trình tiện ích như zcat, zless, zdiff và zgrep.

Dưới đây là bảng liệt kê một số lệnh gia đình z:

```
$ zcat nén-file.txt.gz Để xem tệp nén
```

```
$ zless somefile.gz
```

hoặc là

```
$ zmore somefile.gz Đến trang thông qua tệp nén
```

```
$ zgrep -i less somefile.gz Để tìm kiếm bên trong tệp nén
```

```
$ zdiff file1.txt.gz file2.txt.gz Để so sánh hai tệp nén
```

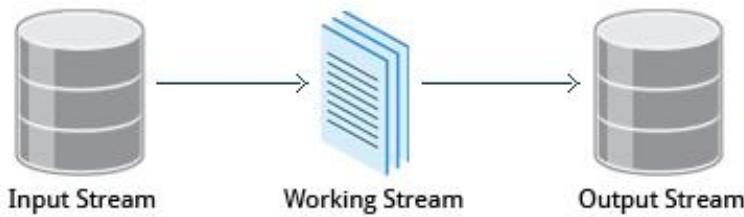
Lưu ý rằng nếu bạn chạy zless trên một tệp không nén, nó sẽ vẫn hoạt động và bỏ qua giai đoạn giải nén. Ngoài ra còn có các chương trình tiện ích tương đương cho các phương thức nén khác ngoài gzip, ví dụ: chúng ta có bzcat và bzless được liên kết với bzip2, và xzcat và xzless được liên kết với xz.

### 6.8 Sử dụng các lệnh sed và awk

Nó là rất phổ biến để tạo và sau đó liên tục chỉnh sửa và / hoặc trích xuất nội dung từ một tập tin. Hãy cùng học cách sử dụng sed và awk để dễ dàng thực hiện các thao tác đó.

Lưu ý rằng nhiều người dùng và quản trị viên Linux sẽ viết các tập lệnh bằng các ngôn ngữ kịch bản toàn diện như Python và perl, thay vì sử dụng sed và awk (và một số tiện ích khác chúng ta sẽ thảo luận sau). Sử dụng các tiện ích như vậy chắc chắn là tốt trong hầu hết các trường hợp; người ta luôn cảm thấy thoải mái khi sử dụng các công cụ mà người

ta có kinh nghiệm. Tuy nhiên, các tiện ích được mô tả ở đây nhẹ hơn nhiều; tức là họ sử dụng ít tài nguyên hệ thống hơn và thực thi nhanh hơn. Có những tình huống (chẳng hạn như trong quá trình khởi động hệ thống), trong đó rất nhiều thời gian sẽ bị lãng phí khi sử dụng các công cụ phức tạp hơn và hệ thống thậm chí có thể không chạy được chúng. Vì vậy, các công cụ đơn giản hơn sẽ luôn luôn cần thiết.



**sed** là một công cụ xử lý văn bản mạnh mẽ và là một trong những tiện ích UNIX lâu đời nhất, sớm nhất và phổ biến nhất. Nó được sử dụng để sửa đổi nội dung của một tệp, thường đặt nội dung vào một tệp mới. Tên của nó là viết tắt cho trình soạn thảo luồng.

sed có thể lọc văn bản, cũng như thực hiện thay thế trong luồng dữ liệu.

Dữ liệu từ một nguồn / tệp đầu vào (hoặc luồng) được lấy và di chuyển đến một không gian làm việc. Toàn bộ danh sách các hoạt động / sửa đổi được áp dụng trên dữ liệu trong không gian làm việc và nội dung cuối cùng được chuyển đến không gian đầu ra tiêu chuẩn (hoặc luồng).

Bạn có thể gọi sed bằng cách sử dụng các lệnh như được liệt kê trong bảng đi kèm.

lệnh sed -e <tên tệp>

Chỉ định các lệnh chỉnh sửa tại dòng lệnh, hoạt động trên tệp và đưa đầu ra ra ngoài tiêu chuẩn (ví dụ: thiết bị đầu cuối)

sed -f scriptfile <filename> Chỉ định scriptfile chứa các lệnh sed, hoạt động trên tệp và đưa đầu ra ra ngoài tiêu chuẩn

Tùy chọn lệnh -e cho phép bạn chỉ định đồng thời nhiều lệnh chỉnh sửa tại dòng lệnh. Nó là không cần thiết nếu bạn chỉ có một hoạt động được gọi.

```

student@fedora:~$ cat /etc/sysconfig/man-db
# Set this to "no" to disable man-db update triggered by installation
# of any package containing manual pages
SERVICE="yes"

# Set this to
# /etc/cron.daily/ma
CRON="yes" # of any package containing manual pages
SERVICE="yes"

# Options used
OPTS="-q"
CRON="yes"
[student@fedor
[student@fedor# Options used by mandb, we use "-q" as default, too much noise without it
OPTS="-q"
[student@fedor# Set this to
[student@fedor# Set this to "no" to disable man-db update triggered by installation
# of any packa
SERVICE="no" # of any package containing manual pages
SERVICE="no"

# Set this to
# Set this to "no" to disable daily man-db update run by
# /etc/cron.daily/man-db.cron
# /etc/cron.da
CRON="no"
CRON="no" # Options used by mandb, we use "-q" as default, too much noise without it
OPTS="-q"
# Options used
[student@fedor ~]$ 
OPTS="-q"
[student@fedora ~]$ 

```

Bây giờ bạn đã biết rằng bạn có thể thực hiện nhiều thao tác chỉnh sửa và lọc với sed, hãy để giải thích một số chi tiết trong số đó. Bảng giải thích một số thao tác cơ bản, trong đó mẫu là chuỗi hiện tại và thay\_ chuỗi là chuỗi mới:

sed s / mẫu / thay\_ chuỗi / tệp Thay thế chuỗi đầu tiên xuất hiện trong mỗi dòng

tệp sed s / mẫu / thay\_ chuỗi / g Thay thế tất cả các lần xuất hiện chuỗi trong mỗi dòng

tệp sed 1,3s / mẫu / thay\_ chuỗi / g Thay thế tất cả các lần xuất hiện chuỗi trong một phạm vi dòng

tệp sed -i s / mẫu / thay\_ chuỗi / g Lưu các thay đổi để thay thế chuỗi trong cùng một tệp

Bạn phải sử dụng tùy chọn -i một cách cẩn thận, vì hành động không thể đảo ngược. Luôn an toàn hơn khi sử dụng sed mà không có tùy chọn HPi và sau đó tự thay thế tệp, như trong ví dụ sau:

\$ sed s / mẫu / thay\_ chuỗi / g file1 > file2

Lệnh trên sẽ thay thế tất cả các lần xuất hiện của mẫu bằng thay thế \_ chuỗi trong tệp1 và di chuyển nội dung sang tệp2. Nội dung của file2 có thể được xem với cat file2. Nếu bạn chấp thuận, bạn có thể ghi đè lên tệp gốc bằng mv file2 file1.

Ví dụ: Để chuyển đổi 01/02 / Đổi thành JAN / FEB / Nhận

```
sed -e 's/01/JAN/' -e 's/02/FEB/' -e 's/03/MAR/' -e 's/04/APR/' -e 's/05/MAY/' \
-e 's/06/JUN/' -e 's/07/JUL/' -e 's/08/AUG/' -e 's/09/SEP/' -e 's/10/OCT/' \
-e 's/11/NOV/' -e 's/12/DEC/'
```

## awk

awk được sử dụng để trích xuất và sau đó in nội dung cụ thể của một tệp và thường được sử dụng để xây dựng các báo cáo. Nó được tạo ra tại Bell Labs vào những năm 1970 và lấy tên từ tên cuối cùng của các tác giả: Alfred Aho, Peter Weinberger và Brian Kernighan.

awk có các tính năng sau:

Nó là một tiện ích mạnh mẽ và ngôn ngữ lập trình diễn giải.

Nó được sử dụng để thao tác với các tệp dữ liệu, truy xuất và xử lý văn bản.

Nó hoạt động tốt với các trường (chứa một đoạn dữ liệu, về cơ bản là một cột) và các bản ghi (một tập hợp các trường, về cơ bản là một dòng trong một tệp).

awk được gọi như sau:

```
student@Linux-Mint-18 ~$ head -10 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
student@Linux-Mint-18 ~$ awk -F: '{print "name: "$1" shell:" $7}' \
> /etc/passwd | head -10
name: root shell:/bin/bash
name: daemon shell:/usr/sbin/nologin
name: bin shell:/usr/sbin/nologin
name: sys shell:/usr/sbin/nologin
name: sync shell:/bin.sync
name: games shell:/usr/sbin/nologin
name: man shell:/usr/sbin/nologin
name: lp shell:/usr/sbin/nologin
name: mail shell:/usr/sbin/nologin
name: news shell:/usr/sbin/nologin
student@Linux-Mint-18 ~$
```

Cũng như sed, các lệnh awk ngắn có thể được chỉ định trực tiếp tại dòng lệnh, nhưng một tập lệnh phức tạp hơn có thể được lưu trong một tệp mà bạn có thể chỉ định bằng tùy chọn -f.

awk ‘command’ file

Chỉ định một lệnh trực tiếp tại dòng lệnh

awk -f scriptfile file

Chỉ định một tập tin có chứa tập lệnh sẽ được thực thi

Bảng giải thích các tác vụ cơ bản có thể được thực hiện bằng awk. Tệp đầu vào được đọc một dòng tại một thời điểm và, đối với mỗi dòng, awk khớp với mẫu đã cho theo thứ tự đã cho và thực hiện hành động được yêu cầu. Tùy chọn -F cho phép bạn chỉ định một ký tự phân tách trường cụ thể. Ví dụ: tệp / etc / passwd sử dụng ":" để phân tách các trường, do đó, tùy chọn -F: được sử dụng với tệp / etc / passwd.

Lệnh / hành động trong awk cần được bao quanh bằng dấu nháy đơn (hoặc trích dẫn ()). awk có thể được sử dụng như sau:

```
awk '{print $ 0}' / etc / passwd In toàn bộ tập tin
```

awk -F: '{print \$ 1}' / etc / passwd In trường đầu tiên (cột) của mỗi dòng, cách nhau bằng khoảng trắng

awk -F: '{in \$ 1 \$ 7}' / etc / passwd In trường đầu tiên và thứ bảy của mỗi dòng

## 6.9 Các lệnh thao tác với tập tin văn bản

Khi quản lý tập của bạn, bạn có thể cần thực hiện nhiều tác vụ, chẳng hạn như sắp xếp dữ liệu và sao chép dữ liệu từ vị trí này sang vị trí khác. Linux cung cấp một số tiện ích thao tác tệp mà bạn có thể sử dụng trong khi làm việc với các tệp văn bản. Trong phần này, bạn sẽ tìm hiểu về các chương trình thao tác tệp sau: **sort**, **uniq**, **paste**, **join** và **split**.

Bạn cũng sẽ tìm hiểu về các biểu thức thông thường và các mẫu tìm kiếm.

**sort** được sử dụng để sắp xếp lại các dòng của tệp văn bản theo thứ tự tăng dần hoặc giảm dần, theo một khóa sắp xếp. Bạn cũng có thể sắp xếp theo các trường cụ thể của một tệp. Khóa sắp xếp mặc định là thứ tự của các ký tự ASCII (nghĩa là về cơ bản theo thứ tự bảng chữ cái).

**sort** có thể được sử dụng như sau:

Cách sử dụng cú pháp

sort <tên tệp> Sắp xếp các dòng trong tệp được chỉ định, theo các ký tự ở đầu mỗi dòng  
tập tin mèo1 tập tin2 | sắp xếp Kết hợp hai tệp, sau đó sắp xếp các dòng và hiển thị đầu ra  
trên thiết bị đầu cuối

sort -r <filename> Sắp xếp các dòng theo thứ tự ngược lại

sort -k 3 <tên tệp> Sắp xếp các dòng theo trường thứ 3 trên mỗi dòng thay vì bắt đầu

Khi được sử dụng với tùy chọn -u, sắp xếp kiểm tra các giá trị duy nhất sau khi sắp xếp các bản ghi (dòng). Nó tương đương với việc chạy uniq (mà chúng ta sẽ thảo luận) trên đầu ra của sort.

### uniq

uniq loại bỏ các dòng liên tiếp trùng lặp trong một tệp văn bản và rất hữu ích để đơn giản hóa việc hiển thị văn bản.

Bởi vì uniq yêu cầu các mục trùng lặp phải liên tiếp, nên người ta thường chạy sắp xếp trước và sau đó chuyển đầu ra thành uniq; nếu sort được sử dụng với tùy chọn -u, nó có thể thực hiện tất cả điều này trong một bước.

```
File Edit View Search Terminal Help
c7:/etc/default>cat grub
GRUB_TIMEOUT="3"
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT="saved"
GRUB_DISABLE_SUBMENU="true"
GRUB_TERMINAL_OUTPUT="console"
#GRUB_CMDLINE_LINUX="vconsole.keymap=us crashkernel=auto vconsole.font=latarcyrheb-sun16 rhgb quiet"
"GRUB_CMDLINE_LINUX="rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
GRUB_TERMINAL=gnome-terminal
GRUB_BACKGROUND="/boot/despair.jpg"
c7:/etc/default>
c7:/etc/default>
c7:/etc/default>sort grub
GRUB_BACKGROUND="/boot/despair.jpg"
GRUB_CMDLINE_LINUX="rhgb quiet"
#GRUB_CMDLINE_LINUX="vconsole.keymap=us crashkernel=auto vconsole.font=latarcyrheb-sun16 rhgb quiet"
"GRUB_DEFAULT="saved"
GRUB_DISABLE_RECOVERY="true"
GRUB_DISABLE_SUBMENU="true"
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_TERMINAL=gnome-terminal
GRUB_TERMINAL_OUTPUT="console"
GRUB_TIMEOUT="3"
c7:/etc/default>
```

```
student@openSUSE:/tmp
File Edit View Search Terminal Help
student@openSUSE:/tmp> sort uniqdemo.txt
beaver
beaver
eddie
eddie
june
larry
lumpy
lumpy
wally
wally
ward
whitey
student@openSUSE:/tmp> sort -u uniqdemo.txt
beaver
eddie
june
larry
lumpy
wally
ward
whitey
student@openSUSE:/tmp>
```

Để xóa các mục trùng lặp khỏi nhiều tệp cùng một lúc, hãy sử dụng lệnh sau:

`sort file1 file2 | uniq > file3`

hoặc là

`sort -u file1 file2 > file3`

Để đếm số lượng mục trùng lặp, sử dụng lệnh sau:

`tên tập tin uniq -c`

**paste**

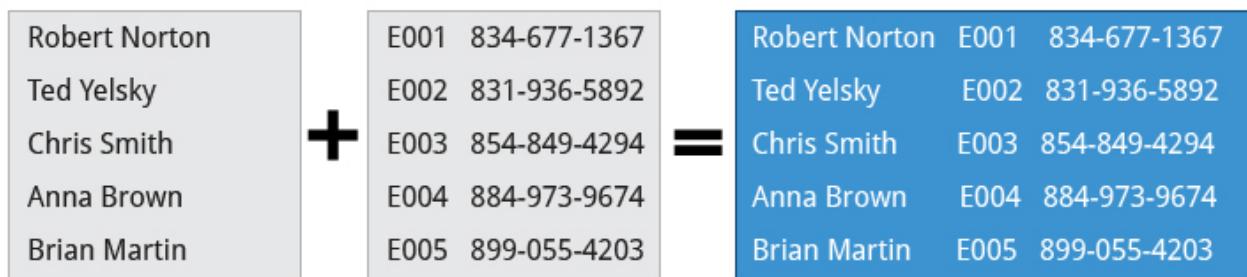
Giả sử bạn có một tệp chứa tên đầy đủ của tất cả nhân viên và một tệp khác liệt kê số điện thoại và ID nhân viên của họ. Bạn muốn tạo một tệp mới chứa tất cả dữ liệu được liệt kê trong ba cột: tên, ID nhân viên và số điện thoại. Làm thế nào bạn có thể làm điều này một cách hiệu quả mà không cần đầu tư quá nhiều thời gian?

**paste** có thể được sử dụng để tạo một tập tin duy nhất chứa cả ba cột. Các cột khác nhau được xác định dựa trên các dấu phân cách (khoảng cách được sử dụng để phân tách hai trường). Ví dụ: dấu phân cách có thể là khoảng trắng, tab hoặc Enter. Trong hình ảnh được cung cấp, một khoảng trắng được sử dụng làm dấu phân cách trong tất cả các tệp.

**past** chấp nhận các tùy chọn sau:

-d dấu phân cách, trong đó chỉ định danh sách các dấu phân cách sẽ được sử dụng thay vì các tab để phân tách các giá trị liên tiếp trên một dòng. Mỗi dấu phân cách được sử dụng lần lượt; khi danh sách đã hết, dán bắt đầu lại ở dấu phân cách đầu tiên.

-s, nguyên nhân gây ra dán để nối dữ liệu theo chuỗi chứ không phải song song; đó là, theo chiều ngang chứ không phải theo chiều dọc.



past có thể được sử dụng để kết hợp các trường (như tên hoặc số điện thoại) từ các tệp khác nhau, cũng như kết hợp các dòng từ nhiều tệp. Ví dụ, dòng một từ tệp1 có thể được kết hợp với dòng một của tệp2, dòng hai từ tệp1 có thể được kết hợp với dòng hai của tệp2, v.v.

Để dán nội dung từ hai tệp người ta có thể làm:

```
$ past file1 file2
```

Cú pháp sử dụng một dấu phân cách khác nhau như sau:

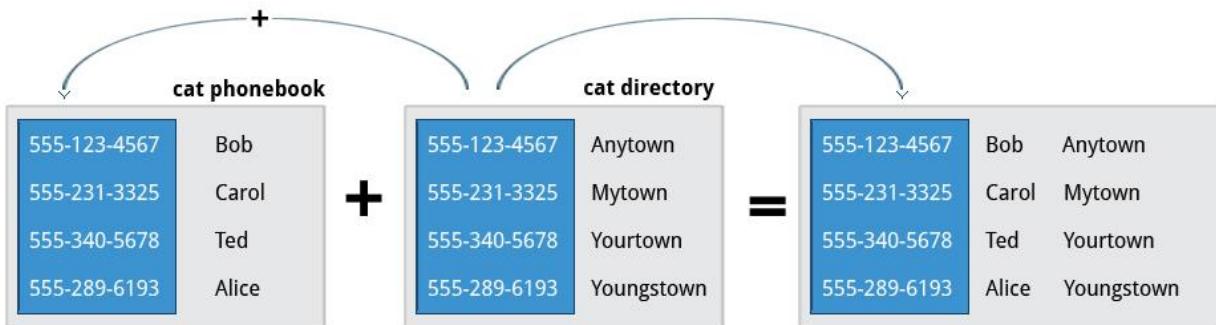
```
$ past -d, file1 file2
```

Các dấu phân cách phổ biến là 'dấu cách', 'tab', '|', 'dấu phẩy', v.v.

### join

Giả sử bạn có hai tệp với một số cột tương tự. Bạn đã lưu nhân viên số điện thoại trong hai tệp, một tệp có tên của họ và tên còn lại có họ. Bạn muốn kết hợp các tệp mà không lặp lại dữ liệu của các cột phổ biến. Làm thế nào để bạn đạt được điều này?

Nhiệm vụ trên có thể đạt được bằng cách sử dụng phép nối, về cơ bản là phiên bản nâng cao của dán. Đầu tiên, nó kiểm tra xem các tệp có chia sẻ các trường chung hay không, chẳng hạn như tên hoặc số điện thoại, sau đó nối các dòng trong hai tệp dựa trên một trường chung.



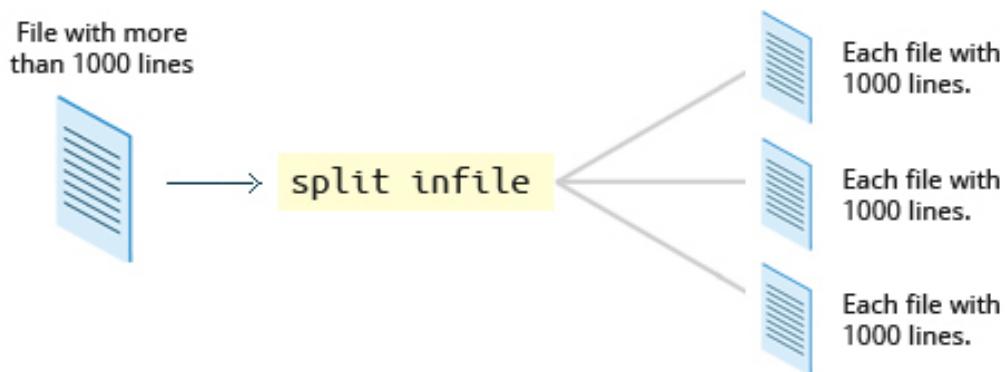
Để kết hợp hai tệp trên một trường chung, tại dấu nhắc lệnh, nhập tệp1 tệp2 và nhấn phím Enter.

Ví dụ: trường chung (nghĩa là nó chứa cùng các giá trị) trong số các danh bạ và tệp thành phố là số điện thoại và kết quả của việc nối hai tệp này được hiển thị trong ảnh chụp màn hình.

### split

`split` được sử dụng để chia (hoặc tách) một tệp thành các phân đoạn có kích thước bằng nhau để xem và thao tác dễ dàng hơn và thường chỉ được sử dụng trên các tệp tương đối lớn. Theo mặc định, phân chia chia một tệp thành các phân đoạn 1000 dòng. Tệp gốc không thay đổi và một tập hợp các tệp mới có cùng tên cộng với tiền tố được thêm vào được tạo. Theo mặc định, tiền tố `x` được thêm vào. Để chia một tập tin thành các phân đoạn, sử dụng lệnh chia nhỏ `infile`.

Để phân chia tệp thành các phân đoạn bằng cách sử dụng tiền tố khác, hãy sử dụng lệnh chia nhỏ `infile <Tiền tố>`.



```
student@ubuntu: /tmp$ wc /usr/share/dict/american-english
99171 99171 938848 /usr/share/dict/american-english
student@ubuntu: /tmp$ split /usr/share/dict/american-english dictionary
student@ubuntu: /tmp$ ls -l dictionary* | wc
100 900 6100
student@ubuntu: /tmp$ ls -l dictionary* | head -10
-rw-rw-r-- 1 student student 8653 Dec 23 17:17 dictionaryaa
-rw-rw-r-- 1 student student 8552 Dec 23 17:17 dictionaryab
-rw-rw-r-- 1 student student 8876 Dec 23 17:17 dictionaryac
-rw-rw-r-- 1 student student 8842 Dec 23 17:17 dictionaryad
-rw-rw-r-- 1 student student 8249 Dec 23 17:17 dictionaryae
-rw-rw-r-- 1 student student 8405 Dec 23 17:17 dictionaryaf
-rw-rw-r-- 1 student student 8494 Dec 23 17:17 dictionaryag
-rw-rw-r-- 1 student student 8010 Dec 23 17:17 dictionaryah
-rw-rw-r-- 1 student student 8222 Dec 23 17:17 dictionaryai
-rw-rw-r-- 1 student student 8566 Dec 23 17:17 dictionaryaj
student@ubuntu: /tmp$
```

Chúng ta sẽ áp dụng phân chia cho một tệp từ điển Mỹ-Anh gồm hơn 99.000 dòng:

\$ wc -l mỹ-tiếng Anh

99171 tiếng Mỹ

nơi chúng tôi đã sử dụng wc (đếm từ, sẽ sớm được thảo luận) để báo cáo về số lượng dòng trong tệp. Sau đó, gõ:

\$ chia từ điển tiếng Anh

sẽ chia tệp Mỹ-Anh thành 100 phân đoạn có kích thước bằng nhau có tên 'dictionaryxx'. Cái cuối cùng tất nhiên sẽ có phần nhỏ hơn.

## 6.10 Các biểu thức chính quy và mẫu tìm kiếm

Biểu thức thông thường là các chuỗi văn bản được sử dụng để khớp với một mẫu cụ thể hoặc để tìm kiếm một vị trí cụ thể, chẳng hạn như bắt đầu hoặc kết thúc của một dòng hoặc một từ. Biểu thức thông thường có thể chứa cả ký tự bình thường hoặc cái gọi là ký tự meta, chẳng hạn như \* và \$.

Nhiều trình soạn thảo văn bản và các tiện ích như vi, sed, awk, find và grep hoạt động rộng rãi với các biểu thức thông thường. Một số ngôn ngữ máy tính phổ biến sử dụng biểu thức chính quy bao gồm Perl, Python và Ruby. Nó có thể trở nên khá phức tạp và có toàn bộ sách viết về các biểu thức thông thường; do đó, chúng tôi sẽ không làm gì ngoài việc lướt qua bề mặt ở đây.

Các biểu thức chính quy này khác với các ký tự đại diện (hoặc ký tự meta) được sử dụng trong khớp tên tệp trong các lệnh như bash (được trình bày trong Chương 7: Hoạt động dòng lệnh). Bảng liệt kê các mẫu tìm kiếm và cách sử dụng của chúng.

(chấm) Phù hợp với bất kỳ ký tự đơn

a | z Ghép a hoặc z

\$ Kết thúc chuỗi

^ Khớp đầu chuỗi

\* Khớp mục trước 0 lần trở lên

Ví dụ, hãy xem xét câu sau: the quick brown fox jumped over the lazy dog

Một số mẫu có thể được áp dụng cho câu này như sau:

.. matches azy

b.j. matches both br and ju

..\$ matches og

l.\* matches lazy dog

l.\*y matches lazy

the.\* matches the whole sentence

### Bài tập LAB: xử lý tập tin văn bản với awk, sort, uniq

Tạo một cột chứa một danh sách duy nhất của tất cả các shell được sử dụng cho người dùng trong / etc / passwd.

Bạn có thể cần tham khảo trang hướng dẫn cho / etc / passwd như trong:

\$ man 5 passwd

Trường nào trong /etc/passwd chứa shell mặc định của tài khoản (trình thông dịch lệnh người dùng)?

Làm thế nào để bạn tạo một danh sách các mục duy nhất (không lặp lại)?

### grep

grep được sử dụng rộng rãi như một công cụ tìm kiếm văn bản chính. Nó quét các tệp cho các mẫu được chỉ định và có thể được sử dụng với các biểu thức thông thường, cũng như các chuỗi đơn giản, như được hiển thị trong bảng:

grep [mẫu] <tên tệp> Tìm kiếm một mẫu trong một tệp và in tất cả các dòng khớp

grep -v [mẫu] <tên tệp> In tất cả các dòng không khớp với mẫu

grep [0-9] <tên tệp> In các dòng có chứa các số từ 0 đến 9

grep -C 3 [mẫu] <tên tệp> In bối cảnh của các dòng (số dòng được chỉ định ở trên và bên dưới mẫu) để khớp với mẫu. Ở đây, số lượng dòng được chỉ định là 3

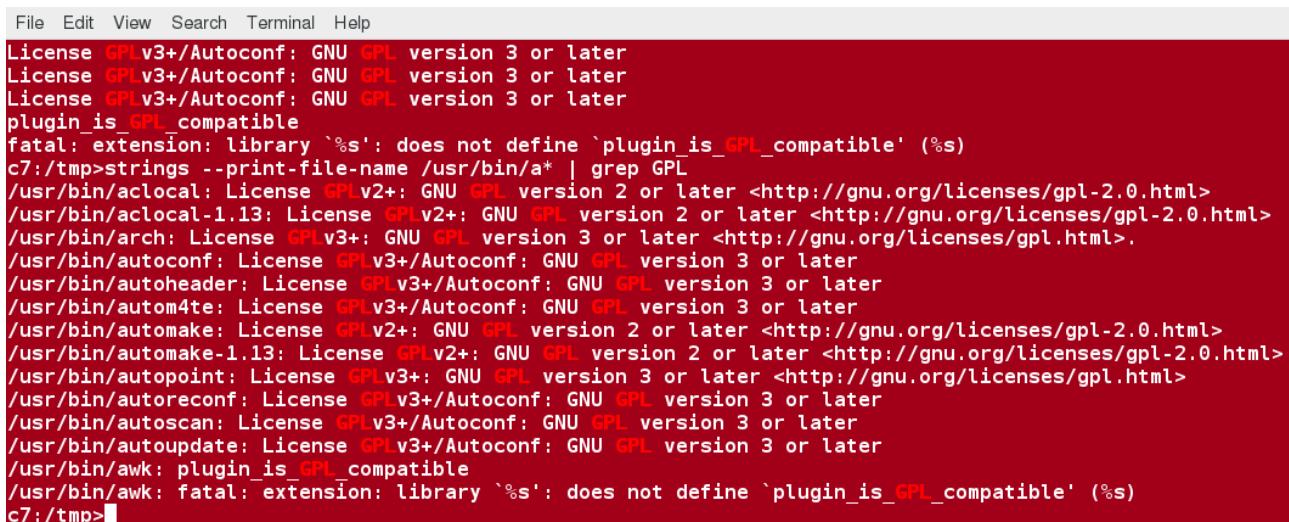
## strings

**strings** được sử dụng để trích xuất tất cả các chuỗi ký tự có thể in được tìm thấy trong tệp hoặc tệp được cung cấp dưới dạng đối số. Nó rất hữu ích trong việc định vị nội dung có thể đọc được của con người được nhúng trong các tệp nhị phân; Đối với các tệp văn bản người ta chỉ có thể sử dụng grep.

Ví dụ: để tìm kiếm chuỗi my\_string trong bảng tính:

```
$ strings book1.xls | grep my_string
```

Ảnh chụp màn hình cho thấy tìm kiếm một số chương trình để xem chương trình nào có giấy phép GPL của các phiên bản khác nhau.



```
File Edit View Search Terminal Help
License GPLv3+/Autoconf: GNU GPL version 3 or later
License GPLv3+/Autoconf: GNU GPL version 3 or later
License GPLv3+/Autoconf: GNU GPL version 3 or later
plugin_is_GPL_compatible
fatal: extension: library `%' does not define `plugin_is_GPL_compatible' (%s)
c7:/tmp>strings --print-file-name /usr/bin/a* | grep GPL
/usr/bin/aclocal: License GPLv2+: GNU GPL version 2 or later <http://gnu.org/licenses/gpl-2.0.html>
/usr/bin/aclocal-1.13: License GPLv2+: GNU GPL version 2 or later <http://gnu.org/licenses/gpl-2.0.html>
/usr/bin/arch: License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
/usr/bin/autoconf: License GPLv3+/Autoconf: GNU GPL version 3 or later
/usr/bin/autoheader: License GPLv3+/Autoconf: GNU GPL version 3 or later
/usr/bin/autom4te: License GPLv3+/Autoconf: GNU GPL version 3 or later
/usr/bin/automake: License GPLv2+: GNU GPL version 2 or later <http://gnu.org/licenses/gpl-2.0.html>
/usr/bin/automake-1.13: License GPLv2+: GNU GPL version 2 or later <http://gnu.org/licenses/gpl-2.0.html>
/usr/bin/autopoint: License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
/usr/bin/autoreconf: License GPLv3+/Autoconf: GNU GPL version 3 or later
/usr/bin/autoscanc: License GPLv3+/Autoconf: GNU GPL version 3 or later
/usr/bin/autoupdate: License GPLv3+/Autoconf: GNU GPL version 3 or later
/usr/bin/awk: plugin_is_GPL_compatible
/usr/bin/awk: fatal: extension: library `%' does not define `plugin_is_GPL_compatible' (%s)
c7:/tmp>
```

## Bài tập LAB – Sử dụng lệnh grep

Sau đây chúng tôi đưa ra một số ví dụ về những điều bạn có thể làm với lệnh grep; nhiệm vụ của bạn là thử nghiệm những ví dụ này và mở rộng chúng.

Tìm kiếm tên người dùng của bạn trong tập tin / etc / passwd.

Tìm tất cả các mục trong / etc / services bao gồm chuỗi ftp:

Hạn chế cho những người sử dụng giao thức tcp.

Bây giờ giới hạn ở những người không sử dụng giao thức tcp, trong khi in ra số dòng

Nhận tất cả các chuỗi bắt đầu bằng ts hoặc kết thúc bằng st.

tr

Trong phần này, bạn sẽ tìm hiểu về một số tiện ích văn bản bổ sung mà bạn có thể sử dụng để thực hiện các hành động khác nhau trên các tệp Linux của mình, chẳng hạn như thay đổi trường hợp chữ cái hoặc xác định số lượng từ, dòng và ký tự trong một tệp.

The screenshot shows three terminal windows side-by-side. The top window has the title 'student@FC-25:/tmp' and displays the command: [student@FC-25 tmp]\$ cat trinput | tr 'a-l' 'A-L' > troutput [student@FC-25 tmp]\$ . The middle window has the title 'student@FC-25:/tmp' and displays the output of the 'head -20 trinput' command, which lists various kernel module names. The bottom window has the title 'student@FC-25:/tmp' and displays the output of the 'head -20 troutput' command, where the lowercase letters in the middle window's output have been converted to uppercase. The status bar at the bottom right indicates '1 / 2'.

```
[student@FC-25 tmp]$ cat trinput | tr 'a-l' 'A-L' > troutput
[student@FC-25 tmp]$ 

[student@FC-25 tmp]$ head -20 trinput
personal_ws-1.1 en 8252
pcidevice
getstate
vivante
oldmm
ranslation
savesigs
wblayer
adjtimex
subfolders
khugepaged
getstats
lenovo
omain
oprofile
oprofile
gitweb
mwait

[student@FC-25 tmp]$ head -20 troutput
pErsonAL_ws-1.1 En 8252
pCIDEvICe
GEtstAtE
vIvAntE
oLDmm
rAnsLAtIon
sAvEsIGs
wBLAyEr
ADJtImEx
suBFoLDers
KHuGEpAGED
GEtstAts
LEnovo
oMAIn
oproFILE
oproFILE
GITwEB
mwAIT
```

Tiện ích tr được sử dụng để dịch các ký tự được chỉ định sang các ký tự khác hoặc xóa chúng. Cú pháp chung như sau:

```
$ tr [tùy chọn] set1 [set2]
```

Các mục trong ngoặc vuông là tùy chọn. tr yêu cầu ít nhất một đối số và chấp nhận tối đa hai đối số. Tập đầu tiên, được chỉ định set1 trong ví dụ, liệt kê các ký tự trong văn bản sẽ được thay thế hoặc xóa. Thứ hai, set2, liệt kê các ký tự sẽ được thay thế cho các ký tự được liệt kê trong đối số đầu tiên. Đôi khi các bộ này cần được bao quanh bởi dấu nháy đơn (hoặc dấu ngoặc đơn ()) để có vở bỏ qua rằng chúng có ý nghĩa gì đó đặc biệt đối với vỏ. Thông thường sẽ an toàn (và có thể được yêu cầu) để sử dụng dấu ngoặc đơn xung quanh mỗi bộ như bạn sẽ thấy trong các ví dụ bên dưới.

Ví dụ: giả sử bạn có một tệp có tên thành phố chứa một số dòng văn bản trong trường hợp hỗn hợp. Để dịch tất cả các ký tự chữ thường sang chữ hoa, tại dấu nháy lệnh gõ cat city | tr a-z A-Z và nhấn phím Enter.

\$ tr abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ Chuyển đổi chữ thường thành chữ hoa

\$ tr '{}' ''<inputfile> outputfile Dịch dấu ngoặc vào ngoặc đơn

\$ echo "Đây là để thử nghiệm" | tr [: space:] '\t' Dịch khoảng trắng sang các tab

\$ echo "Đây là để thử nghiệm" | tr -s [: không gian:]

Bóp lặp lại các ký tự bằng cách sử dụng -s

\$ echo "công cụ đam mê" | tr -d 't' Xóa các ký tự được chỉ định bằng tùy chọn -d

\$ echo "tên người dùng của tôi là 432234" | tr -cd [: chữ số:] Bổ sung cho các bộ bằng tùy chọn -c

\$ tr -cd [: print:] <file.txt Xóa tất cả các ký tự không in được khỏi tệp

\$ tr -s '\n' '' <file.txt Tham gia tất cả các dòng trong một tệp vào một dòng

### tee

tee lấy đầu ra từ bất kỳ lệnh nào, và trong khi gửi nó đến đầu ra tiêu chuẩn, nó cũng lưu nó vào một tệp. Nói cách khác, nó "khởi động" luồng đầu ra từ lệnh: một luồng được hiển thị trên đầu ra tiêu chuẩn và luồng còn lại được lưu vào một tệp.

Ví dụ, để liệt kê nội dung của một thư mục trên màn hình và lưu kết quả đầu ra vào một tệp, tại dấu nháy lệnh gõ ls -l | tee newfile và nhấn phím Enter.

Gõ cat newfile sau đó sẽ hiển thị đầu ra của ls -l.

```
File Edit View Search Terminal Help

c7:/etc>sudo find . -name "g*cfg" | tee /tmp/tee_output
./grub-customizer/grub.cfg
./grub2-efi.cfg
./grub2.cfg
./grub.d/backup/boot_grub/grub.cfg
c7:/etc>cat /tmp/tee_output
./grub-customizer/grub.cfg
./grub2-efi.cfg
./grub2.cfg
./grub.d/backup/boot_grub/grub.cfg
c7:/etc>
```

## wc

wc (đếm từ) đếm số lượng dòng, từ và ký tự trong một tệp hoặc danh sách các tệp. Các tùy chọn được đưa ra trong bảng dưới đây.

- l Hiển thị số lượng dòng
- c Hiển thị số byte
- w Hiển thị số lượng từ

Theo mặc định, cả ba tùy chọn này đều hoạt động.

Ví dụ: để chỉ in số dòng có trong tệp, nhập tên tệp wc -l và nhấn phím Enter.

```
File Edit View Search Terminal Help
c7:/usr/src/linux/kernel/sched>wc *h
 64   181  1514 auto_group.h
 17    42   359 cpuacct.h
 34    88   733 cpudeadline.h
 31    78   674 cpupri.h
 71   296  1940 features.h
1817  5899 47470 sched.h
 269  1128  8236 stats.h
2303 7712 60926 total
c7:/usr/src/linux/kernel/sched>
```

## Cut

cut được sử dụng để thao tác các tệp dựa trên cột và được thiết kế để trích xuất các cột cụ thể. Dấu phân cách cột mặc định là ký tự tab. Một dấu phân cách khác nhau có thể được cung cấp dưới dạng tùy chọn lệnh.

Ví dụ: để hiển thị cột thứ ba được phân cách bằng khoảng trắng, tại dấu nhắc lệnh gõ

```
$ ls -l | cut -d "" -f3
```

và nhấn phím Enter.

### **Bài tập LAB: sử dụng lệnh tee**

Tiện ích tee rất hữu ích để lưu một bản sao đầu ra của bạn trong khi bạn đang xem nó được tạo.

Thực hiện một lệnh như thực hiện một danh sách thư mục của thư mục / etc:

```
$ ls -l / etc
```

trong khi cả hai lưu đầu ra trong một tập tin và hiển thị nó tại thiết bị đầu cuối của bạn.

### **Bài tập LAB: sử dụng lệnh wc**

Sử dụng wc (đếm từ), tìm hiểu có bao nhiêu dòng, từ và ký tự có trong tất cả các tệp trong / var / log có phần mở rộng .log.

## **6.11 Câu hỏi ôn tập Chương 6**

1. Cách sử dụng và cấu hình tài khoản người dùng và nhóm người dùng.
2. Cách sử dụng và thiết lập các biến môi trường.
3. Cách sử dụng lịch sử lệnh shell trước đó.
4. Cách sử dụng phím tắt.
5. Cách sử dụng và xác định bí danh.
6. Cách sử dụng và thiết lập quyền và quyền sở hữu tập tin.

## Chương 7 Cấu hình và làm việc mạng với Linux

### 7.1 Môi trường làm việc mạng trên Linux

Mạng là một nhóm các máy tính và thiết bị máy tính được kết nối với nhau thông qua các kênh truyền thông, chẳng hạn như cáp hoặc phương tiện không dây. Các máy tính được kết nối qua mạng có thể được đặt trong cùng một khu vực địa lý hoặc trải rộng trên toàn thế giới.

Một mạng được sử dụng để:

Cho phép các thiết bị được kết nối giao tiếp với nhau

Cho phép nhiều người dùng chia sẻ thiết bị qua mạng, chẳng hạn như máy in và máy quét

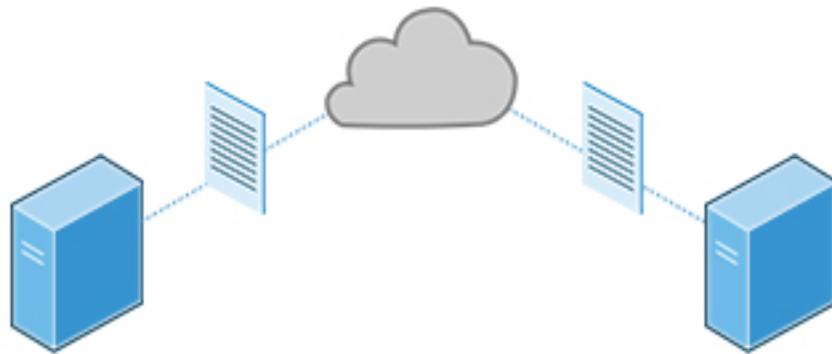
Chia sẻ và quản lý thông tin trên các máy tính dễ dàng.

Hầu hết các tổ chức đều có cả mạng nội bộ và kết nối Internet để người dùng giao tiếp với máy móc và những người bên ngoài tổ chức. Internet là mạng lớn nhất trên thế giới và có thể được gọi là "mạng của các mạng".

### 7.2 Địa chỉ IP

Các thiết bị được gắn vào mạng phải có ít nhất một mã định danh địa chỉ mạng duy nhất được gọi là địa chỉ IP (Giao thức Internet). Địa chỉ này rất cần thiết để định tuyến các gói thông tin qua mạng.

Trao đổi thông tin trên mạng đòi hỏi phải sử dụng các luồng gói nhỏ, mỗi gói chứa một phần thông tin đi từ máy này sang máy khác. Các gói này chứa bộ đệm dữ liệu cùng với các tiêu đề chứa thông tin về nơi gói đến và đến và nơi phù hợp với chuỗi các gói tạo thành luồng. Các giao thức và phần mềm mạng khá phức tạp do sự đa dạng của máy móc và hệ điều hành mà chúng phải xử lý, cũng như thực tế là ngay cả các tiêu chuẩn rất cũ cũng phải được hỗ trợ.



### Phiên bản IPv4 và IPv6

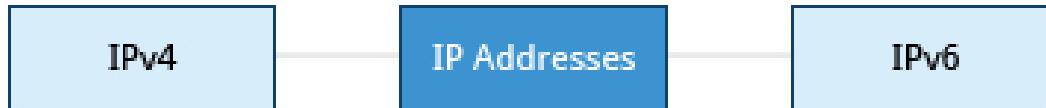
Có hai loại địa chỉ IP khác nhau có sẵn: IPv4 (phiên bản 4) và IPv6 (phiên bản 6). IPv4 cũ hơn và được sử dụng rộng rãi hơn, trong khi IPv6 mới hơn và được thiết kế để vượt qua các giới hạn vốn có trong tiêu chuẩn cũ hơn và cung cấp nhiều địa chỉ khả dĩ hơn.

IPv4 sử dụng 32 bit cho địa chỉ; chỉ có 4,3 tỷ địa chỉ duy nhất có sẵn. Hơn nữa, nhiều địa chỉ được phân bổ và dành riêng, nhưng không thực sự được sử dụng. IPv4 được coi là không đủ để đáp ứng nhu cầu trong tương lai vì số lượng thiết bị có sẵn trên mạng toàn cầu đã tăng lên rất nhiều trong những năm gần đây.

IPv6 sử dụng 128 bit cho địa chỉ; điều này cho phép  $3,4 \times 10^{38}$  địa chỉ duy nhất. Nếu bạn có một mạng máy tính lớn hơn và muốn thêm nhiều hơn, bạn có thể muốn chuyển sang IPv6, vì nó cung cấp nhiều địa chỉ độc đáo hơn. Tuy nhiên, việc chuyển sang IPv6 có thể phức tạp; hai giao thức không phải lúc nào cũng hoạt động tốt. Do đó, việc chuyển thiết bị và địa chỉ sang IPv6 đòi hỏi nỗ lực đáng kể và không hoàn toàn nhanh như dự định ban đầu. Chúng tôi sẽ thảo luận về IPv4 nhiều hơn IPv6 vì bạn có nhiều khả năng đối phó với nó.

Một lý do khiến IPv4 không biến mất là có nhiều cách để cung cấp hiệu quả nhiều địa chỉ hơn bằng các phương thức như NAT (Dịch địa chỉ mạng). NAT cho phép chia sẻ một địa chỉ IP giữa nhiều máy tính được kết nối cục bộ, mỗi máy tính có một địa chỉ duy nhất chỉ

thấy trên mạng cục bộ. Trong khi điều này được sử dụng trong cài đặt tổ chức, nó cũng được sử dụng trong các mạng gia đình đơn giản. Ví dụ: nếu bạn có một bộ định tuyến được nối với Nhà cung cấp dịch vụ Internet của bạn (chẳng hạn như hệ thống cáp), nó sẽ



Uses 32-bits  
Only 4.3 billion unique addresses.

Uses 128-bits  
allows  $3.4 \times 10^{38}$  unique addresses.

cung cấp cho bạn một địa chỉ hiển thị bên ngoài, nhưng cấp cho mỗi thiết bị trong nhà bạn một địa chỉ cục bộ riêng.

### Giải mã địa chỉ IPv4

Địa chỉ IPv4 32 bit được chia thành bốn phần 8 bit được gọi là octet.

Thí dụ:

Địa chỉ IP → 172. 16. 31. 46

Định dạng bit → 10101100.00010000.00011111.00101110

Lưu ý: Octet chỉ là một từ khác cho byte.

Địa chỉ mạng được chia thành năm lớp: A, B, C, D và E. Các lớp A, B và C được phân thành hai phần: Địa chỉ mạng (ID mạng) và Địa chỉ máy chủ (ID máy chủ). Net ID được sử dụng để xác định mạng, trong khi ID máy chủ được sử dụng để xác định máy chủ trong mạng. Lớp D được sử dụng cho các ứng dụng phát đa hướng đặc biệt (thông tin được phát đồng thời tới nhiều máy tính) và Lớp E được dành riêng để sử dụng trong tương lai. Trong phần này, bạn sẽ tìm hiểu về các lớp A, B và C.

	Octet 1	Octet 2	Octet 3	Octet 4
Class A	Network ID	Host ID	Host ID	Host ID
Class B	Network ID	Network ID	Host ID	Host ID
Class C	Network ID	Network ID	Network ID	Host ID
Class D		Multicast addresses		
Class E		Reserved for future use		

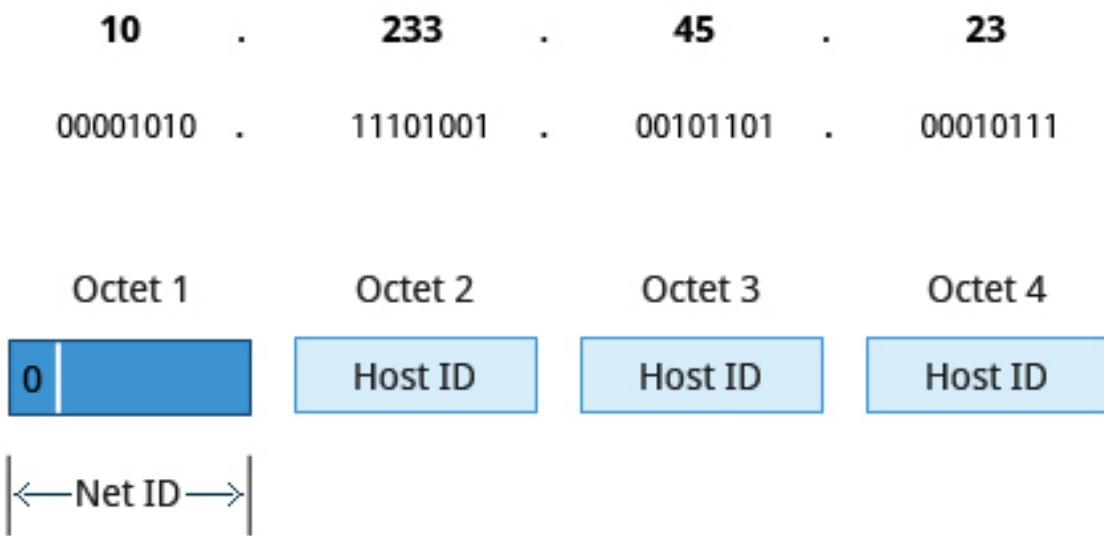
### Địa chỉ mạng lớp A

Địa chỉ lớp A sử dụng octet đầu tiên của địa chỉ IP làm ID Net của họ và sử dụng ba octet khác làm ID máy chủ. Bit đầu tiên của octet đầu tiên luôn được đặt thành 0. Vì vậy, bạn chỉ có thể sử dụng 7 bit cho các số mạng duy nhất. Kết quả là, có tối đa 126 mạng loại A khả dụng (địa chỉ 0000000 và 1111111 được bảo lưu). Không có gì đáng ngạc nhiên, điều này chỉ khả thi khi có rất ít mạng duy nhất có số lượng lớn máy chủ. Khi việc sử dụng Internet được mở rộng, Lớp B và C đã được thêm vào để đáp ứng nhu cầu ngày càng tăng đối với các mạng độc lập.

Mỗi mạng Lớp A có thể có tới 16,7 triệu máy chủ duy nhất trên mạng của mình. Phạm vi địa chỉ máy chủ là từ 1.0.0.0 đến 127.255.255.255.

Lưu ý: Giá trị của một octet, hoặc 8 bit, có thể nằm trong khoảng từ 0 đến 255.

An example of a Class A address is:

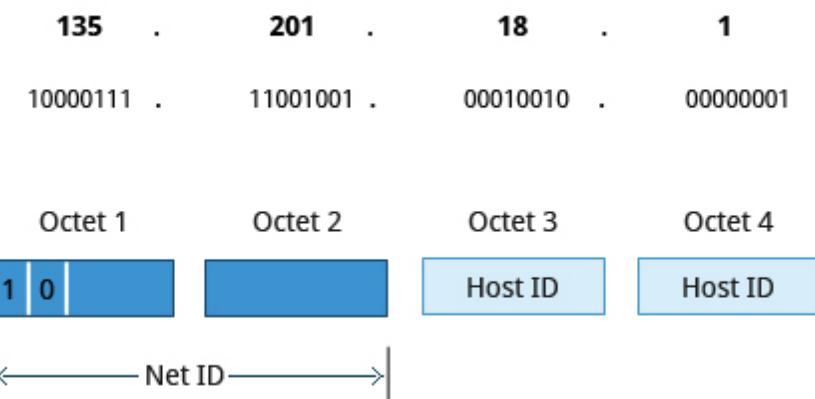


### Địa chỉ mạng lớp B

Địa chỉ lớp B sử dụng hai octet đầu tiên của địa chỉ IP làm Net ID của chúng và hai octet cuối cùng làm ID máy chủ. Hai bit đầu tiên của octet đầu tiên luôn được đặt thành nhị phân 10, do đó, có tối đa 16.384 (14 bit) mạng Lớp B. Octet đầu tiên của địa chỉ Class B có các giá trị từ 128 đến 191. Việc giới thiệu các mạng Class B đã mở rộng số lượng mạng nhưng nó đã sớm trở nên rõ ràng rằng sẽ cần một mức độ cao hơn.

Mỗi mạng Lớp B có thể hỗ trợ tối đa 65.536 máy chủ duy nhất trên mạng của mình. Phạm vi của địa chỉ máy chủ là từ 128.0.0.0 đến 191.255.255.255.

An example of a Class B address is:

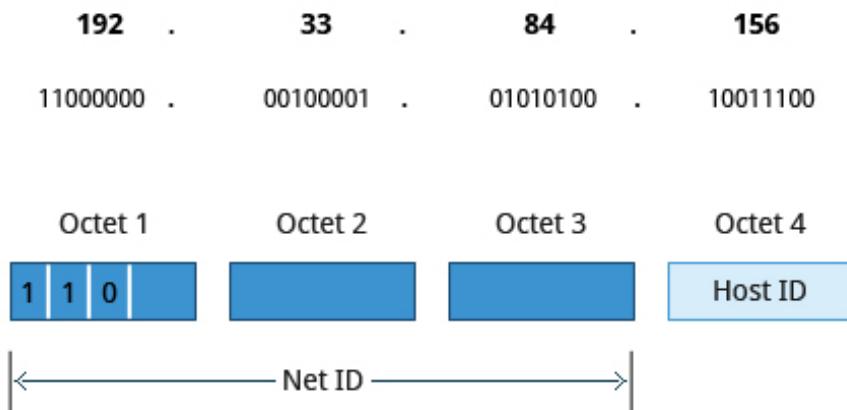


### Địa chỉ mạng lớp C

Địa chỉ lớp C sử dụng ba octet đầu tiên của địa chỉ IP làm ID Net và octet cuối cùng làm ID máy chủ của họ. Ba bit đầu tiên của octet đầu tiên được đặt thành nhị phân 110, do đó, gần như 2,1 triệu (21 bit) mạng Class C có sẵn. Octet đầu tiên của địa chỉ Class C có các giá trị từ 192 đến 223. Đây là những phô biến nhất cho các mạng nhỏ hơn không có nhiều máy chủ duy nhất.

Mỗi mạng Lớp C có thể hỗ trợ tối đa 256 (8 bit) máy chủ duy nhất. Phạm vi của địa chỉ máy chủ là từ 192.0.0.0 đến 223.255.255.255.

An example of a Class C address is:

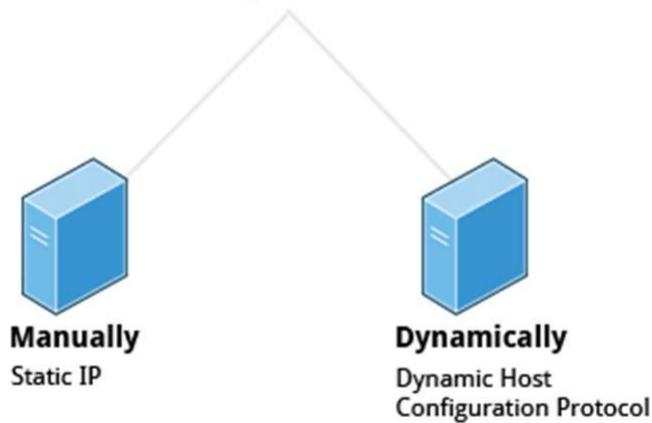


### 7.3 Phân hoạch và cấp phát địa chỉ IP

Thông thường, một loạt các địa chỉ IP được yêu cầu từ Nhà cung cấp dịch vụ Internet (ISP) của bạn bởi quản trị viên mạng của tổ chức. Thông thường, việc bạn chọn loại địa chỉ IP nào bạn được cung cấp tùy thuộc vào kích thước mạng của bạn và nhu cầu tăng trưởng dự kiến. Nếu NAT đang hoạt động, chẳng hạn như trong mạng gia đình, bạn chỉ nhận được một địa chỉ hiển thị bên ngoài!

Bạn có thể gán địa chỉ IP cho các máy tính qua mạng theo cách thủ công hoặc động. Gán thủ công thêm địa chỉ tĩnh (không bao giờ thay đổi) vào mạng. Địa chỉ được gán động có thể thay

#### Allocating IP Addresses



đổi mỗi khi bạn khởi động lại hoặc thậm chí thường xuyên hơn; Giao thức cấu hình máy chủ động (DHCP) được sử dụng để gán địa chỉ IP.

#### 7.4 Phân giải tên miền qua địa chỉ IP

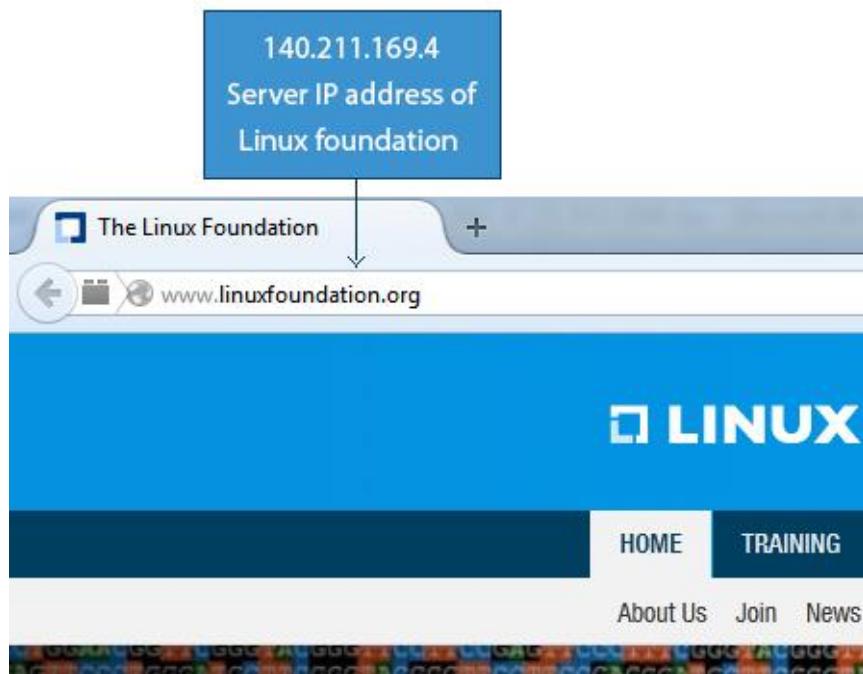
Tên Độ phân giải được sử dụng để chuyển đổi các giá trị địa chỉ IP số thành định dạng có thể đọc được của con người được gọi là tên máy chủ. Ví dụ: 104,95,85,15 là địa chỉ IP số để cập đến tên máy chủ Whitehouse.gov. Tên máy chủ dễ nhớ hơn nhiều!

Cho một địa chỉ IP, bạn có thể có được tên máy chủ tương ứng. Truy cập máy qua mạng trở nên dễ dàng hơn khi bạn có thể nhập tên máy chủ thay vì địa chỉ IP.

Bạn có thể xem tên máy chủ của hệ thống của bạn chỉ bằng cách nhập tên máy chủ mà không có đốii số.

Lưu ý: Nếu bạn đưa ra một đốii số, hệ thống sẽ cố gắng thay đổi tên máy chủ của nó để khớp với nó, tuy nhiên, chỉ người dùng root mới có thể làm điều đó.

Tên máy chủ lưu trữ đặc biệt localhost được liên kết với địa chỉ IP 127.0.0.1 và mô tả máy bạn hiện đang sử dụng (thường có thêm địa chỉ IP liên quan đến mạng).

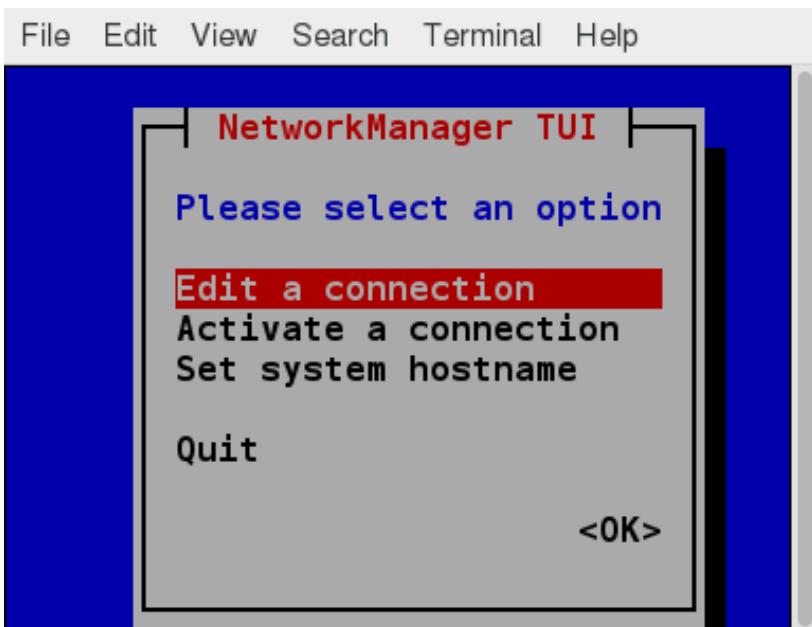


#### 7.5 Các tập tin cấu hình mạng

Các tập tin cấu hình mạng là rất cần thiết để đảm bảo các giao diện hoạt động chính xác. Chúng được đặt trong cây thư mục / etc. Tuy nhiên, các tệp chính xác được sử dụng trước đây phụ thuộc vào bản phân phối và phiên bản Linux cụ thể đang được sử dụng.

Đối với cấu hình họ Debian, các tệp cấu hình mạng cơ bản có thể được tìm thấy trong /etc/network/, trong khi đối với các hệ thống họ Fedora và SUSE, người ta cần kiểm tra /etc/sysconfig/mạng.

Các hệ thống hiện đại nhấn mạnh việc sử dụng Trình quản lý mạng, chúng tôi đã thảo luận ngắn gọn khi chúng tôi xem xét quản trị hệ thống đồ họa, thay vì cố gắng theo kịp sự mở rộng của các tệp trong /etc. Mặc dù các phiên bản đồ họa của Trình quản lý mạng trông hơi khác nhau trong các bản phân phối khác nhau, nhưng tiện ích nmtui (hiển thị trong ảnh chụp màn hình) hầu như không thay đổi, cũng như tiện ích nmcli (giao diện dòng lệnh) thậm chí còn thua thót hơn. Nếu bạn thành thạo trong việc sử dụng GUI, bằng mọi cách, hãy sử dụng chúng. Nếu bạn đang làm việc trên nhiều hệ thống khác nhau, các tiện ích cấp thấp hơn có thể giúp cuộc sống dễ dàng hơn.



## 7.6 Các giao tiếp với thiết bị mạng

Giao diện mạng là kênh kết nối giữa thiết bị và mạng. Về mặt vật lý, giao diện mạng có thể tiến hành thông qua thẻ giao diện mạng (NIC) hoặc có thể được triển khai trùm tượng hơn dưới dạng phần mềm. Bạn có thể có nhiều giao diện mạng hoạt động cùng một lúc. Các giao diện cụ thể có thể được đưa lên (kích hoạt) hoặc đưa xuống (khử kích hoạt) bất cứ lúc nào.

Thông tin về một giao diện mạng cụ thể hoặc tất cả các giao diện mạng có thể được báo cáo bởi các tiện ích ip và ifconfig, mà bạn có thể phải chạy như siêu người dùng, hoặc ít nhất, đưa ra đường dẫn đầy đủ, tức là /sbin/ifconfig, trên một số bản phân phối. ip mới hơn ifconfig và có nhiều khả năng hơn, nhưng đầu ra của nó xấu hơn mắt người. Một số

bản phân phối Linux mới không cài đặt gói công cụ mạng cũ hơn mà ifconfig thuộc về, và vì vậy bạn sẽ phải cài đặt nó nếu bạn muốn sử dụng nó.

```
File Edit View Search Terminal Help
c7:/tmp>ifconfig eno1
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.200 netmask 255.255.255.0 broadcast 192.168.1.255
        ether 08:62:66:45:4d:16 txqueuelen 1000 (Ethernet)
          RX packets 777139 bytes 698855238 (666.4 MiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 533909 bytes 330980403 (315.6 MiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
          device interrupt 20 memory 0xf7d00000-f7d20000

c7:/tmp>ip -s link show eno1
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 08:62:66:45:4d:16 brd ff:ff:ff:ff:ff:ff
      RX: bytes packets errors dropped overrun mcast
        698855238 777139 0 0 0 109
      TX: bytes packets errors dropped carrier collsns
        330980403 533909 0 0 0 0
c7:/tmp>ip address show eno1
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:62:66:45:4d:16 brd ff:ff:ff:ff:ff:ff
      inet 192.168.1.200/24 brd 192.168.1.255 scope global dynamic eno1
        valid_lft 74195sec preferred_lft 74195sec
c7:/tmp>
```

## 7.7 Các lệnh tiện ích làm việc mạng

### Lệnh ip

Để xem địa chỉ IP:

\$ /sbin / ip addr hiển thị

Để xem thông tin định tuyến:

\$ /sbin / ip hiển thị tuyến đường

```
File Edit View Search Terminal Help
c7:/home/coop>ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
      inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:62:66:45:4d:16 brd ff:ff:ff:ff:ff:ff
      inet 192.168.1.200/24 brd 192.168.1.255 scope global dynamic eno1
        valid_lft 72470sec preferred_lft 72470sec
3: vmnet1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 00:50:56:c0:00:01 brd ff:ff:ff:ff:ff:ff
      inet 172.16.43.1/24 brd 172.16.43.255 scope global vmnet1
        valid_lft forever preferred_lft forever
4: vmnet8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 00:50:56:c0:00:08 brd ff:ff:ff:ff:ff:ff
      inet 172.16.2.1/24 brd 172.16.2.255 scope global vmnet8
        valid_lft forever preferred_lft forever
c7:/home/coop>
```

**ip** là một chương trình rất mạnh mẽ có thể làm nhiều việc. Các tiện ích cũ hơn (và cũ hơn) như ifconfig và tuyến đường thường được sử dụng để thực hiện các tác vụ tương tự. Nhìn vào các trang **man** có liên quan có thể cho bạn biết nhiều hơn về các tiện ích này.

## Lệnh ping

ping được sử dụng để kiểm tra xem một máy được gắn vào mạng có thể nhận và gửi dữ liệu hay không; tức là nó xác nhận rằng máy chủ từ xa đang trực tuyến và đang phản hồi.

Để kiểm tra trạng thái của máy chủ từ xa, tại dấu nhắc lệnh, nhập ping <tên máy chủ>.

ping thường được sử dụng để kiểm tra và quản lý mạng; tuy nhiên, việc sử dụng nó có thể làm tăng tải mạng không được chấp nhận. Do đó, bạn có thể hủy bỏ việc thực thi ping bằng cách nhập CTRL-C hoặc bằng cách sử dụng tùy chọn -c, giới hạn số lượng gói mà ping sẽ gửi trước khi thoát. Khi thực hiện dừng, một bản tóm tắt được hiển thị.

```
File Edit View Search Terminal Help
c7:/tmp>ping -c 10 linuxfoundation.com
PING linuxfoundation.com (140.211.169.4) 56(84) bytes of data.
From corv-car1-gw.nero.net (207.98.64.17) icmp_seq=1 Packet filtered
From corv-car1-gw.nero.net (207.98.64.17) icmp_seq=2 Packet filtered
From corv-car1-gw.nero.net (207.98.64.17) icmp_seq=3 Packet filtered
From corv-car1-gw.nero.net (207.98.64.17) icmp_seq=4 Packet filtered
From corv-car1-gw.nero.net (207.98.64.17) icmp_seq=5 Packet filtered
From corv-car1-gw.nero.net (207.98.64.17) icmp_seq=6 Packet filtered
From corv-car1-gw.nero.net (207.98.64.17) icmp_seq=7 Packet filtered
From corv-car1-gw.nero.net (207.98.64.17) icmp_seq=8 Packet filtered
From corv-car1-gw.nero.net (207.98.64.17) icmp_seq=9 Packet filtered
From corv-car1-gw.nero.net (207.98.64.17) icmp_seq=10 Packet filtered

--- linuxfoundation.com ping statistics ---
10 packets transmitted, 0 received, +10 errors, 100% packet loss, time 9011ms
c7:/tmp>
```

## Lệnh route

Một mạng yêu cầu kết nối của nhiều nút. Dữ liệu di chuyển từ nguồn đến đích bằng cách đi qua một loạt các bộ định tuyến và có khả năng trên nhiều mạng. Máy chủ duy trì các bảng định tuyến chứa địa chỉ của từng nút trong mạng. Các giao thức định tuyến IP cho phép các bộ định tuyến xây dựng bảng chuyển tiếp tương quan các điểm đến cuối cùng với các địa chỉ bước nhảy tiếp theo.

```

student@ubuntu: /etc
student@ubuntu:/etc$ route -n
Kernel IP routing table
Destination      Gateway      Genmask      Flags Metric Ref    Use Iface
0.0.0.0          192.168.1.1  0.0.0.0      UG     100    0        0 ens33
169.254.0.0      0.0.0.0      255.255.0.0  U      1000   0        0 ens33
192.168.1.0      0.0.0.0      255.255.255.0 U      100    0        0 ens33
192.168.122.0    0.0.0.0      255.255.255.0 U      0       0        0 virbr0
student@ubuntu:/etc$ ip route
default via 192.168.1.1 dev ens33 proto static metric 100
169.254.0.0/16 dev ens33 scope link metric 1000
192.168.1.0/24 dev ens33 proto kernel scope link src 192.168.1.24 metric 100
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown
student@ubuntu:/etc$ 

```

Người ta có thể sử dụng tiện ích định tuyến hoặc lệnh ip tuyến mới hơn để xem hoặc thay đổi bảng định tuyến IP để thêm, xóa hoặc sửa đổi các tuyến (tĩnh) cụ thể thành các máy chủ hoặc mạng cụ thể. Bảng giải thích một số lệnh có thể được sử dụng để quản lý định tuyến IP:

### Lệnh route

Hiển thị bảng định tuyến hiện tại \$ tuyến đường lộ trình hoặc ip

Thêm tuyến tĩnh \$ tuyến thêm địa chỉ -net hoặc thêm tuyến ip

Xóa tuyến tĩnh \$ route del -net address hoặc ip route del

### Lệnh traceroute

traceroute được sử dụng để kiểm tra tuyến đường mà gói dữ liệu đi đến máy chủ đích, điều này làm cho nó khá hữu ích để khắc phục sự cố chậm trễ và lỗi mạng. Bằng cách sử dụng traceroute, bạn có thể cách ly các vấn đề kết nối giữa các bước nhảy, giúp giải quyết chúng nhanh hơn.

Để in tuyến được thực hiện bởi gói đến máy chủ mạng, tại dấu nhắc lệnh, nhập traceroute <address>.

```

File Edit View Search Terminal Help
c7:/tmp>traceroute www.linuxfoundation.org
traceroute to www.linuxfoundation.org (140.211.169.4), 30 hops max, 60 byte packets
 1 gateway (192.168.1.1)  0.386 ms  0.455 ms  0.510 ms
 2 * * *
 3 dtr02ftbgwi-tge-0-6-0-3.ftbg.wi.charter.com (96.34.24.122)  24.020 ms  24.476 ms  24.555 ms
 4 crr01ftbgwi-bue-101.ftbg.wi.charter.com (96.34.30.246)  24.273 ms  24.371 ms  24.623 ms
 5 crr01euclwi-bue-400.eucl.wi.charter.com (96.34.24.224)  26.629 ms  26.521 ms  26.707 ms
 6 crr02sgnwmi-tge-0-0-0-4.sgnw.mi.charter.com (96.34.2.153)  32.672 ms  17.489 ms  18.903 ms
 7 bbr02euclwi-bue-5.eucl.wi.charter.com (96.34.0.7)  21.910 ms  22.064 ms  21.976 ms
 8 bbr01chcgil-bue-1.chcg.il.charter.com (96.34.0.9)  47.642 ms  47.696 ms  47.540 ms
 9 prr01chcgil-bue-2.chcg.il.charter.com (96.34.3.9)  50.501 ms  50.580 ms  50.366 ms
10 ge-7-2-3.0.rtr.eqch.net.internet2.edu (64.57.20.49)  50.277 ms  56.285 ms ge-7-2-0.0.rtr.eqch.
net.internet2.edu (162.252.69.153)  40.472 ms
11 ae-1.80.rtsw.chic.net.internet2.edu (64.57.20.150)  39.146 ms  39.147 ms  39.155 ms
12 ae-0.80.rtsw.kans.net.internet2.edu (64.57.20.148)  50.416 ms  48.018 ms  52.376 ms
13 ae-0.80.rtsw.salt.net.internet2.edu (64.57.20.146)  81.953 ms  79.413 ms  84.802 ms
14 ae-2.80.rtsw.losa.net.internet2.edu (64.57.20.144)  86.829 ms  86.982 ms  84.276 ms
15 nero-cps1.ptck-core1-gw.oregon-gigapop.net (198.32.165.198)  93.184 ms  93.248 ms  94.575 ms
16 ptck-p2-gw.nero.net (207.98.64.170)  96.579 ms ptck-p1-gw.nero.net (207.98.64.168)  97.037 ms
ptck-p2-gw.nero.net (207.98.64.170)  95.485 ms
17 corv-p2-gw.nero.net (207.98.64.27)  97.880 ms corv-p1-gw.nero.net (207.98.64.25)  98.413 ms
corv-p2-gw.nero.net (207.98.64.27)  102.088 ms
18 corv-car1-gw.nero.net (207.98.64.19)  98.220 ms  95.977 ms corv-car1-gw.nero.net (207.98.64.17)
)  98.659 ms
19 corv-car1-gw.nero.net (207.98.64.17)  95.829 ms !X  93.986 ms !X *
c7:/tmp>

```

## Các công cụ mạng khác

Bây giờ, hãy để học hỏi về một số công cụ mạng bổ sung. Các công cụ mạng rất hữu ích để theo dõi và gỡ lỗi các sự cố mạng, chẳng hạn như kết nối mạng và lưu lượng mạng.

**ethtool** Truy vấn và cũng có thể đặt các tham số khác nhau như tốc độ

**netstat** Hiển thị tất cả các kết nối hoạt động và bảng định tuyến. Hữu ích cho việc theo dõi hiệu suất và xử lý sự cố

**nmap** Quét các cổng mở trên mạng. Quan trọng đối với phân tích bảo mật

**tcpdump** Kết xuất lưu lượng mạng để phân tích

**iptraf** Màn hình lưu lượng truy cập mạng ở chế độ văn bản

**mtr** Kết hợp chức năng của ping và traceroute và hiển thị liên tục được cập nhật

**dig** Kiểm tra hoạt động DNS. Một sự thay thế tốt cho máy chủ và nslookup

## 7.8 Các trình duyệt dùng dòng lệnh (phi đồ họa)

Các trình duyệt được sử dụng để truy xuất, truyền tải và khám phá các tài nguyên thông tin, thường là trên World Wide Web. Người dùng Linux thường sử dụng cả ứng dụng trình duyệt đồ họa và phi đồ họa.

Các trình duyệt đồ họa phổ biến được sử dụng trong Linux là: **Firefox, Google Chrome, Chromium, Konqueror, Opera**.

Đôi khi, bạn không có môi trường đồ họa để làm việc (hoặc có lý do không sử dụng nó) nhưng vẫn cần truy cập tài nguyên web. Trong trường hợp như vậy, bạn có thể sử dụng các trình duyệt phi đồ họa, chẳng hạn như sau:

**wget** Trình duyệt web dựa trên văn bản có thể cấu hình; trình duyệt sớm nhất và vẫn còn được sử dụng

**curl** Dựa trên Lynx. Nó có thể hiển thị bảng và khung

**w3m** Một trình duyệt web dựa trên văn bản khác với nhiều tính năng.

## **wget**

Đôi khi, bạn cần tải xuống tệp và thông tin, nhưng trình duyệt không phải là lựa chọn tốt nhất, vì bạn muốn tải xuống nhiều tệp và / hoặc thư mục hoặc bạn muốn thực hiện hành động từ một dòng lệnh hoặc tập lệnh. wget là một tiện ích dòng lệnh có thể xử lý các loại tải xuống sau:

Tải tập tin lớn

Tải xuống đệ quy, trong đó một trang web để cập đến các trang web khác và tất cả được tải xuống cùng một lúc

Tải mật khẩu yêu cầu

Nhiều tập tin tải về.

Để tải xuống một trang web, bạn chỉ cần nhập wget <url> và sau đó bạn có thể đọc trang đã tải xuống dưới dạng tệp cục bộ bằng trình duyệt đồ họa hoặc phi đồ họa.

```
File Edit View Search Terminal Help
c7:/tmp>wget https://training.linuxfoundation.org/cm/prep/ready-for.sh
--2017-01-10 11:42:56-- https://training.linuxfoundation.org/cm/prep/ready-for.sh
Resolving training.linuxfoundation.org (training.linuxfoundation.org)... 140.211.169.4
Connecting to training.linuxfoundation.org (training.linuxfoundation.org)|140.211.169.4|:443...
... connected.
HTTP request sent, awaiting response... 200 OK
Length: 131804 (129K) [application/octet-stream]
Saving to: 'ready-for.sh.1'

100%[=====] 131,804      442KB/s   in 0.3s

2017-01-10 11:42:57 (442 KB/s) - 'ready-for.sh.1' saved [131804/131804]
c7:/tmp>
```

## **curl**

Bên cạnh việc tải xuống, bạn có thể muốn có được thông tin về một URL, chẳng hạn như mã nguồn đang được sử dụng. curl có thể được sử dụng từ dòng lệnh hoặc tập lệnh để đọc thông tin đó. curl cũng cho phép bạn lưu nội dung của một trang web vào một tập tin, cũng như wget.

Bạn có thể đọc một URL bằng cách sử dụng curl <URL>. Ví dụ: nếu bạn muốn đọc <http://www.linuxfoundation.org>, hãy nhập curl <http://www.linuxfoundation.org>.

Để lấy nội dung của một trang web và lưu trữ nó vào một tệp, hãy nhập curl -o đã lưu.html <http://www.mysite.com>. Nội dung của tệp chỉ mục chính tại trang web sẽ được lưu trong tệp.html.

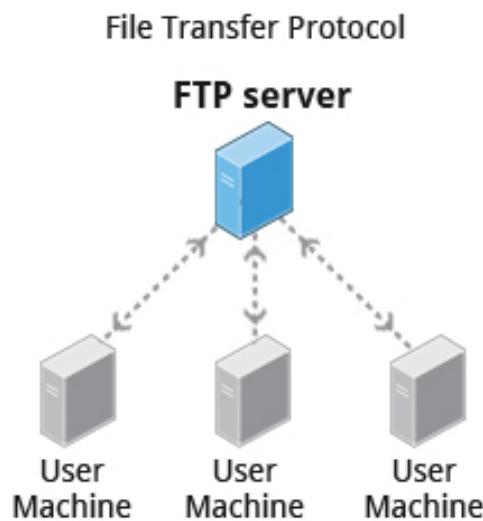
```
File Edit View Search Terminal Help
c7:/tmp>curl -o lf.html https://www.linuxfoundation.org
  % Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
               Dload  Upload Total   Spent   Left Speed
100 326k    0 326k    0     0  330k      0 --:--:-- --:--:-- --:--:-- 330k
c7:/tmp>wc lf.html
 4322 13180 334463 lf.html
c7:/tmp>w3m -dump lf.html | head
Skip to main content
Home
Change Language [English]
• Membership
  □ Corporate Members
  □ Individual Supporters
  □ Become a Member
  □ Success Stories
• Offerings
  □ Compliance
c7:/tmp>
```

## 7.9 Các lệnh di chuyển tập tin

### FTP

Khi bạn được kết nối với mạng, bạn có thể cần phải truyền tệp từ máy này sang máy khác. Giao thức truyền tệp (FTP) là một phương pháp nổi tiếng và phổ biến để truyền tệp giữa các máy tính sử dụng Internet. Phương pháp này được xây dựng trên mô hình máy khách-máy chủ. FTP có thể được sử dụng trong trình duyệt hoặc với các chương trình máy khách độc lập.

FTP là một trong những phương thức truyền dữ liệu mạng lâu đời nhất, có từ đầu những năm 1970. Như vậy, nó được coi là không đủ cho nhu cầu hiện đại, cũng như không an toàn về bản chất. Tuy nhiên, nó vẫn đang được sử dụng và khi bảo mật không phải là vấn đề đáng lo ngại (như với cái gọi là FTP ẩn danh) thì nó có thể có ý nghĩa. Tuy nhiên, nhiều trang web, chẳng hạn như kernel.org, đã từ bỏ việc sử dụng nó.



## FTP Client

Máy khách FTP cho phép bạn truyền tệp với máy tính từ xa bằng giao thức FTP. Những máy khách này có thể là công cụ đồ họa hoặc dòng lệnh. Ví dụ, Filezilla cho phép sử dụng phương pháp kéo và thả để truyền tệp giữa các máy chủ. Tất cả các trình duyệt web đều hỗ trợ FTP, tất cả những gì bạn phải làm là cung cấp một URL như <ftp://ftp.kernel.org> trong đó http:// thông thường trở thành ftp://.

Một số máy khách FTP dòng lệnh là:

ftp

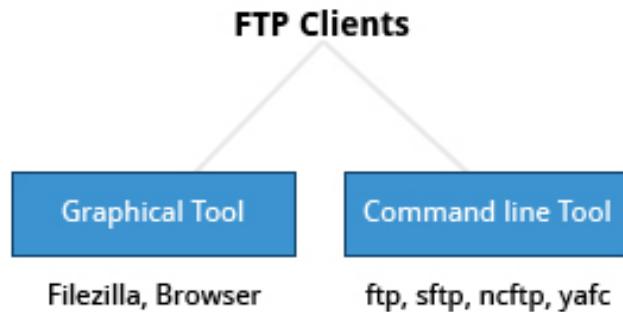
sftp

ncftp

yafc (Một khách hàng FTP khác).

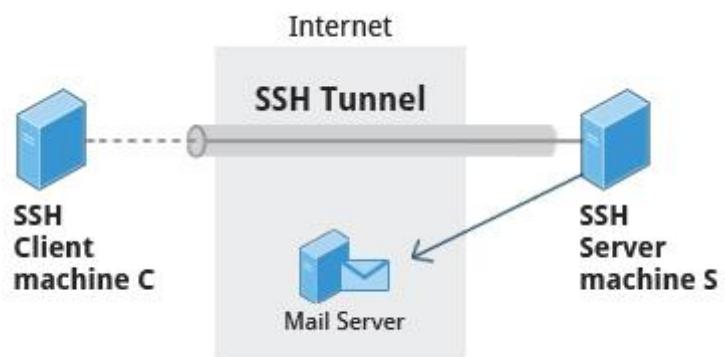
FTP đã rơi vào tình trạng không hài lòng trên các hệ thống hiện đại, vì về bản chất nó không an toàn, vì mật khẩu là thông tin người dùng có thể được truyền đi mà không cần mã hóa và do đó dễ bị chặn. Do đó, nó đã bị xóa vì sử dụng rsync và truy cập https của trình duyệt web chẳng hạn. Thay vào đó, sftp là một chế độ kết nối rất

an toàn, sử dụng giao thức Secure Shell (ssh), chúng ta sẽ thảo luận ngay sau đây. sftp mã hóa dữ liệu của nó và do đó thông tin nhạy cảm được truyền an toàn hơn. Tuy nhiên, nó không hoạt động với cái gọi là FTP ẩn danh (thông tin người dùng khách).



## 7.10 SSH - Thực thi lệnh trên Linux từ xa

Secure Shell (SSH) là một giao thức mạng mật mã được sử dụng để liên lạc dữ liệu an toàn. Nó cũng được sử dụng cho các dịch vụ từ xa và các dịch vụ bảo mật khác giữa hai thiết bị trên mạng và rất hữu ích để quản trị các hệ thống không dễ dàng có thể hoạt động trên thực tế, nhưng bạn có quyền truy cập từ xa.



Để đăng nhập vào hệ thống từ xa bằng cùng tên người dùng của bạn, bạn chỉ cần nhập ssh some\_system và nhấn Enter. ssh sau đó nhắc bạn nhập mật khẩu từ xa. Bạn cũng có thể

định cấu hình ssh để cho phép truy cập từ xa một cách an toàn mà không cần nhập mật khẩu mỗi lần.

Nếu bạn muốn chạy như một người dùng khác, bạn có thể thực hiện ssh -l someone some\_system hoặc ssh someone @ some\_system. Để chạy lệnh trên hệ thống từ xa thông qua SSH, tại dấu nhắc lệnh, bạn có thể nhập ssh some\_system my\_command.

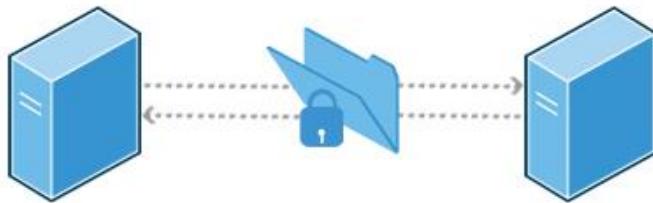
### 7.11 Sao chép tập tin an toàn với scp

Chúng tôi cũng có thể di chuyển các tệp một cách an toàn bằng cách sử dụng Sao chép bảo mật (scp) giữa hai máy chủ được nối mạng. scp sử dụng giao thức SSH để truyền dữ liệu.

Để sao chép tệp cục bộ sang hệ thống từ xa, tại dấu nhắc lệnh, nhập scp <localfile> <user @ remotesystem>: / home / user / và nhấn Enter.

Bạn sẽ nhận được một lời nhắc cho mật khẩu từ xa. Bạn cũng có thể định cấu hình scp để nó không nhắc mật khẩu cho mỗi lần chuyển.

```
scp <localfile> <user@remotesystem>:/home/user/
```



### 7.12 Câu hỏi ôn tập Chương 7

#### + Kiểm tra cấu hình mạng và khử lỗi mạng

Khắc phục sự cố mạng là vấn đề mà bạn sẽ thường gặp phải nếu chưa có. Chúng tôi sẽ thực hành một số công cụ được thảo luận trước đây, có thể giúp bạn cách ly, khắc phục sự cố và khắc phục các sự cố trong mạng của bạn.

Tệp giải pháp chứa quy trình từng bước để thực hiện nhiều công cụ chúng tôi đã nghiên cứu. Vui lòng lặp lại các bước, thay thế tên giao diện mạng thực tế của bạn, địa chỉ mạng và trang web thay thế, v.v.

#### + Sử dụng các trình duyệt phi đồ họa

Đôi khi trình duyệt đồ họa không có sẵn, nhưng bạn cần tra cứu hoặc tải xuống tài nguyên. Trong bài tập này, chúng tôi sẽ thử nghiệm sử dụng các trình duyệt web phi đồ họa.

Tệp giải pháp chứa quy trình từng bước để thực hiện các công cụ được thảo luận. Vui lòng lặp lại các bước, thay thế các trang web, vv

## Câu hỏi ôn tập

1. Giải thích các khái niệm mạng cơ bản, bao gồm các loại mạng và giải quyết các vấn đề.
2. Định cấu hình giao diện mạng và sử dụng các tiện ích mạng cơ bản, chẳng hạn như ifconfig, ip, ping, tuyến đường và traceroute.
3. Sử dụng các trình duyệt đồ họa và phi đồ họa, như Lynx, w3m, Firefox, Chrome và Epiphany.
4. Chuyển tệp đến và từ máy khách và máy chủ bằng cả ứng dụng chế độ đồ họa và văn bản, chẳng hạn như Filezilla, ftp, sftp, curl và wget.

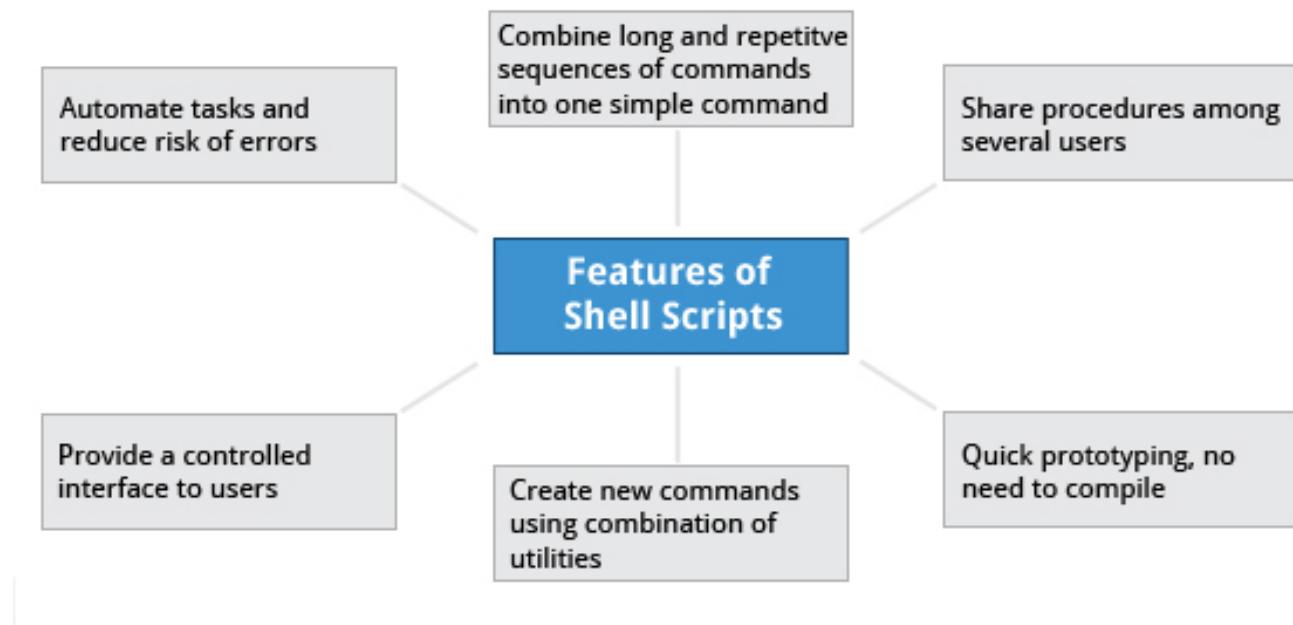


## Chương 8 Lập trình Bash Shell

### 8.1 Giới thiệu lập trình Shell Scripting

Giả sử bạn muốn tra cứu tên tệp, kiểm tra xem tệp liên quan có tồn tại không, sau đó trả lời tương ứng, hiển thị thông báo xác nhận hoặc không xác nhận sự tồn tại của tệp. Nếu bạn chỉ cần thực hiện một lần, bạn có thể chỉ cần gõ một chuỗi lệnh tại một thiết bị đầu cuối. Tuy nhiên, nếu bạn cần làm điều này nhiều lần, tự động hóa là cách để đi. Để tự động hóa các bộ lệnh, bạn sẽ cần học cách viết shell script, trong đó phổ biến nhất được sử dụng với bash. Đồ họa minh họa một số lợi ích của việc triển khai các tập lệnh.

Lưu ý: Nhiều chủ đề được thảo luận trong phần này và chương tiếp theo đã được giới thiệu trước đó, trong khi thảo luận về những điều có thể được thực hiện tại dòng lệnh. Chúng tôi đã chọn để lặp lại một số cuộc thảo luận đó để làm cho các phần về kịch bản tự đứng vững, vì vậy việc lặp lại là có chủ ý, không chỉ là chỉnh sửa câu thả.



### 8.2 Các lựa chọn Command Shell

Trình thông dịch lệnh được giao nhiệm vụ thực thi các câu lệnh tuân theo nó trong kịch bản lệnh. Các trình thông dịch thường được sử dụng bao gồm: /usr/bin/perl, /bin/bash, /bin/csh, /usr/bin/python và /bin/sh.

Việc gõ một chuỗi lệnh dài ở cửa sổ terminal có thể phức tạp, tốn thời gian và dễ bị lỗi. Bằng cách triển khai các tập lệnh shell, sử dụng dòng lệnh trở thành một cách hiệu quả và nhanh chóng để khởi chạy các chuỗi phức tạp của các bước. Thực tế là các tập lệnh shell

được lưu trong một tệp cũng giúp bạn dễ dàng sử dụng chúng để tạo các biến thể tập lệnh mới và chia sẻ các tiến trình chuẩn với một số người dùng.

Linux cung cấp nhiều lựa chọn hệ vỏ; chính xác những gì có sẵn trên hệ thống được liệt kê trong /etc/shell. Các lựa chọn điển hình là:

**/bin/sh**

**/bin/bash**

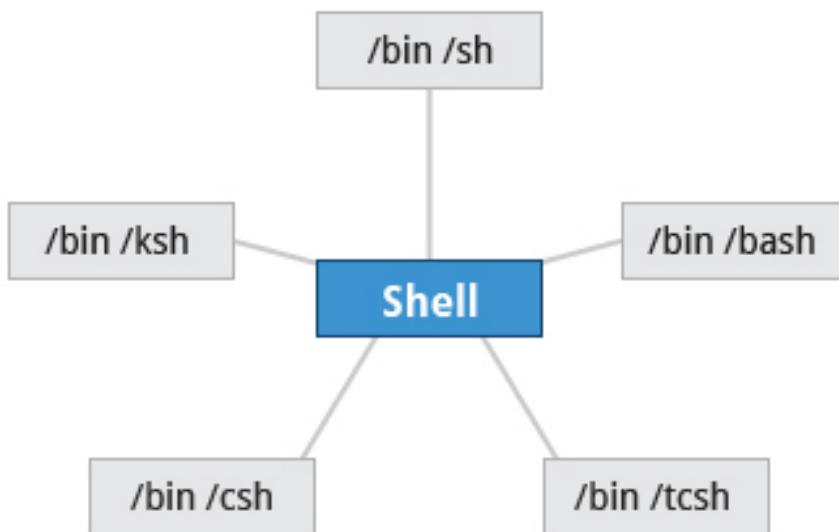
**/bin/tcsh**

**/bin/csh**

**/bin/ksh**

**/bin/zsh**

Hầu hết người dùng Linux sử dụng shell bash mặc định, nhưng những người có nền UNIX dài với các shell khác có thể muốn ghi đè mặc định.



### 8.3 Shell Scripts

Hãy nhớ từ cuộc thảo luận trước đây của chúng tôi, shell là trình thông dịch dòng lệnh cung cấp giao diện người dùng cho các cửa sổ đầu cuối. Nó cũng có thể được sử dụng để chạy các tập lệnh, ngay cả trong các phiên không tương tác mà không có cửa sổ đầu cuối,

như thế các lệnh được nhập trực tiếp vào. Ví dụ: gõ: find. -name "\* .c" -ls tại dòng lệnh thực hiện điều tương tự như thực thi tệp tập lệnh chứa các dòng:

```
#!/bin/bash
```

```
tìm thấy . -name "* .c" -ls
```

Bất cứ ai đã phát triển bất kỳ loại tập lệnh nào trong môi trường UNIX đều phải nhận ra #! / Bin / bash trong dòng đầu tiên. Dòng đầu tiên của tập lệnh, bắt đầu bằng #!, Chứa đường dẫn đầy đủ của trình thông dịch lệnh (trong trường hợp này / bin / bash) sẽ được sử dụng trên tệp. Như chúng tôi đã lưu ý, bạn có khá nhiều lựa chọn cho ngôn ngữ kịch bản mà bạn có thể sử dụng, chẳng hạn như / usr / bin / perl, / bin / csh, / usr / bin / python, v.v.

```
c7:/usr/src/linux/kernel>find . -name "* .c" -ls
50826 24 -rw-r--r-- 1 root root 24389 Dec 12 07:37 ./livepatch/core.c
50799 8 -rw-r--r-- 1 root root 6030 Dec 12 07:37 ./hung_task.c
50761 68 -rw-r--r-- 1 root root 65972 Dec 12 07:37 ./auditsc.c
50784 24 -rw-r--r-- 1 root root 23128 Jul 21 07:46 ./audit_tree.c
54752 4 -rw-r--r-- 1 root root 1704 Jun 1 2016 ./stacktrace.c
53367 8 -rw-r--r-- 1 root root 6347 Oct 15 07:48 ./ksysfs.c
51098 4 -rw-r--r-- 1 root root 2265 Dec 12 07:37 ./sched/cpufreq.c
51095 220 -rw-r--r-- 1 root root 222157 Dec 12 07:37 ./sched/core.c
51100 28 -rw-r--r-- 1 root root 25027 Dec 12 07:37 ./sched/cputime.c
51102 24 -rw-r--r-- 1 root root 23565 Dec 12 07:37 ./sched/debug.c
51101 48 -rw-r--r-- 1 root root 47208 Dec 12 07:37 ./sched/deadline.c
54738 12 -rw-r--r-- 1 root root 11361 Jul 25 06:43 ./sched/loadavg.c
51692 4 -rw-r--r-- 1 root root 2996 Jul 21 07:46 ./sched/stop_task.c
51099 16 -rw-r--r-- 1 root root 15023 Dec 12 07:37 ./sched/cpufreq_schedutil.c
51586 12 -rw-r--r-- 1 root root 8533 Oct 15 07:48 ./sched/cpuacct.c
51093 8 -rw-r--r-- 1 root root 6630 Dec 12 07:37 ./sched/auto_group.c
```

### Thí dụ Shell Script:

Hãy viết một tập lệnh bash đơn giản hiển thị thông báo một dòng trên màn hình. Một trong hai loại:

```
$ cat > hello.sh
#!/bin/bash
echo "Hello Linux Foundation Student"
```

The screenshot shows a terminal window titled "student@openSUSE:~". The window contains the following text:

```
student@openSUSE:~> cat hello.sh
#!/bin/bash
echo "Hello Linux Foundation Student"
student@openSUSE:~> bash hello.sh
Hello Linux Foundation Student
student@openSUSE:~> chmod +x hello.sh
student@openSUSE:~> ./hello.sh
Hello Linux Foundation Student
student@openSUSE:~> █
```

và nhấn ENTER và CTRL-D để lưu tệp hoặc chỉ cần tạo hello.sh trong trình chỉnh sửa văn bản yêu thích của bạn. Sau đó, nhập chmod + x hello.sh để tất cả người dùng thực thi tệp.

Sau đó, bạn có thể chạy tập lệnh bằng cách nhập ./hello.sh hoặc bằng cách thực hiện:

```
$ bash hello.sh
```

Hello Linux

Lưu ý: Nếu bạn sử dụng mẫu thứ hai, bạn không phải làm cho tệp thực thi.

### Thí dụ tương tác sử dụng Shell Script

Bây giờ, hãy xem cách tạo một ví dụ tương tác nhiều hơn bằng cách sử dụng tập lệnh bash. Người dùng sẽ được nhắc nhập một giá trị, sau đó được hiển thị trên màn hình. Giá trị được lưu trữ trong một biến tạm thời, tên. Chúng ta có thể tham chiếu giá trị của biến shell bằng cách sử dụng \$ ở phía trước tên biến, chẳng hạn như \$ name. Để tạo tập lệnh này, bạn cần tạo một tệp có tên getname.sh trong trình chỉnh sửa yêu thích của bạn với nội dung sau:

```
#!/bin/bash
# Đọc tương tác của một biến
echo "NHẬP TÊN CỦA BẠN"
read name
# Hiển thị đầu vào biến
echo "Your name is name: $ name"
```

Một lần nữa, làm cho nó thực thi được bằng cách thực hiện chmod + x getname.sh.

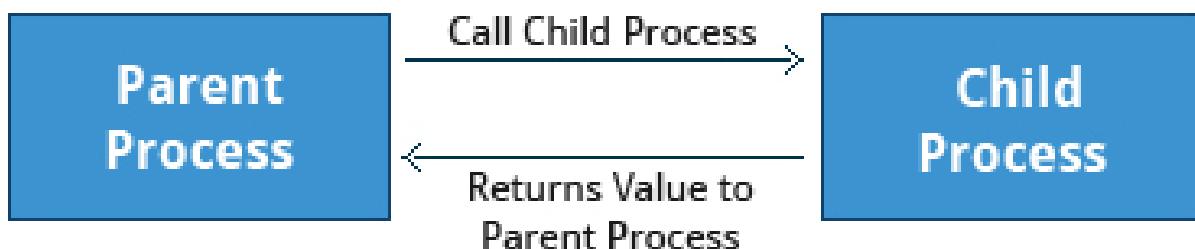
Trong ví dụ trên, khi người dùng gõ. / Getname.sh và tập lệnh được thực thi, người dùng sẽ được nhắc với chuỗi ENTER TÊN CỦA BẠN. Sau đó, người dùng cần nhập một giá trị và nhấn phím Enter. Giá trị sau đó sẽ được in ra.

Lưu ý: Thể băm / bảng hiệu / ký hiệu số (#) được sử dụng để bắt đầu nhận xét trong tập lệnh và có thể được đặt ở bất kỳ đâu trong dòng (phần còn lại của dòng được coi là nhận xét). Tuy nhiên, lưu ý sự kết hợp ma thuật đặc biệt của #!, Được sử dụng trên dòng đầu tiên, là một ngoại lệ duy nhất cho quy tắc này.

```
student@openSUSE:/tmp
File Edit View Search Terminal Help
student@openSUSE:/tmp> cat getname.sh
#!/bin/bash
# Interactive reading of a variable
echo "ENTER YOUR NAME"
read name
# Display variable input
echo The name given was: $name
student@openSUSE:/tmp> ./getname.sh
ENTER YOUR NAME
Sal Paradise
The name given was: Sal Paradise
student@openSUSE:/tmp>
```

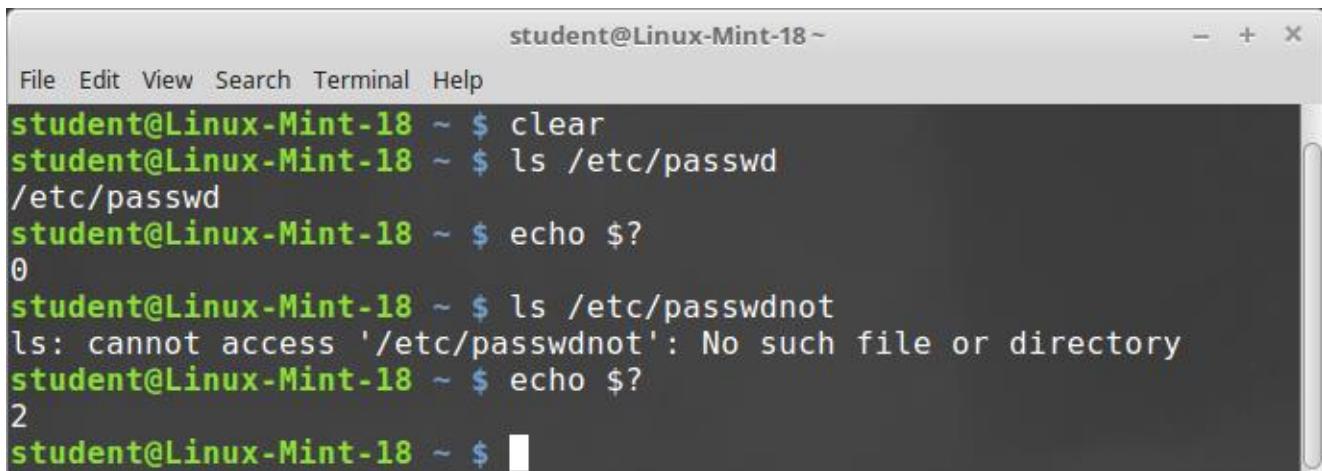
## 8.4 Giá trị trả lại từ Shell Script

Tất cả các tập lệnh shell tạo ra một giá trị trả về khi thực hiện kết thúc, có thể được đặt rõ ràng bằng câu lệnh thoát. Các giá trị trả về cho phép một tiến trình giám sát trạng thái thoát của một tiến trình khác, thường là trong mối quan hệ cha-con. Biết làm thế nào quá trình chấm dứt cho phép thực hiện bất kỳ bước thích hợp cần thiết hoặc phụ thuộc vào thành công hay thất bại.



### Xem giá trị trả lại

Tất cả các tập lệnh shell tạo ra một giá trị trả về khi thực hiện kết thúc, có thể được đặt rõ ràng bằng câu lệnh thoát. Các giá trị trả về cho phép một tiến trình giám sát trạng thái thoát của một tiến trình khác, thường là trong mối quan hệ cha-con. Biết làm thế nào quá trình chấm dứt cho phép thực hiện bất kỳ bước thích hợp cần thiết hoặc phụ thuộc vào thành công hay thất bại.



```
student@Linux-Mint-18 ~ $ clear
student@Linux-Mint-18 ~ $ ls /etc/passwd
/etc/passwd
student@Linux-Mint-18 ~ $ echo $?
0
student@Linux-Mint-18 ~ $ ls /etc/passwdnot
ls: cannot access '/etc/passwdnot': No such file or directory
student@Linux-Mint-18 ~ $ echo $?
2
student@Linux-Mint-18 ~ $
```

### Thực hành LAB:

Viết một shell script:

Có ls cho một tệp không tồn tại, và sau đó hiển thị trạng thái thoát kết quả.

Tạo một tệp và thực hiện ls cho nó, và sau đó một lần nữa hiển thị trạng thái thoát kết quả.

## 8.5 Cú pháp cơ bản và ký tự đặc biệt

Các tập lệnh yêu cầu bạn tuân theo một cú pháp ngôn ngữ tiêu chuẩn. Các quy tắc mô tả cách xác định các biến và cách xây dựng và định dạng các câu lệnh được phép, v.v. Bảng liệt kê một số cách sử dụng ký tự đặc biệt trong các tập lệnh bash:

Mô tả nhân vật

# Được sử dụng để thêm nhận xét, ngoại trừ khi được sử dụng dưới dạng \# hoặc #! khi bắt đầu một kịch bản

\ Được sử dụng ở cuối dòng để biểu thị tiếp tục cho dòng tiếp theo

; Được sử dụng để giải thích những gì tiếp theo như một lệnh mới sẽ được thực hiện tiếp theo

\$ Chỉ ra những gì sau đây là một biến môi trường

> Chuyển hướng đầu ra

>> Nối đầu ra

<Chuyển hướng đầu vào

| Được sử dụng để dẫn kết quả vào lệnh tiếp theo

Có các ký tự và cấu trúc ký tự đặc biệt khác mà các tập lệnh hiểu, như (..), {..}, [..], &&, ||, ', ", \$ ((...)), một số trong đó chúng ta sẽ thảo luận sau.

+ Tách các lệnh dài trên nhiều dòng

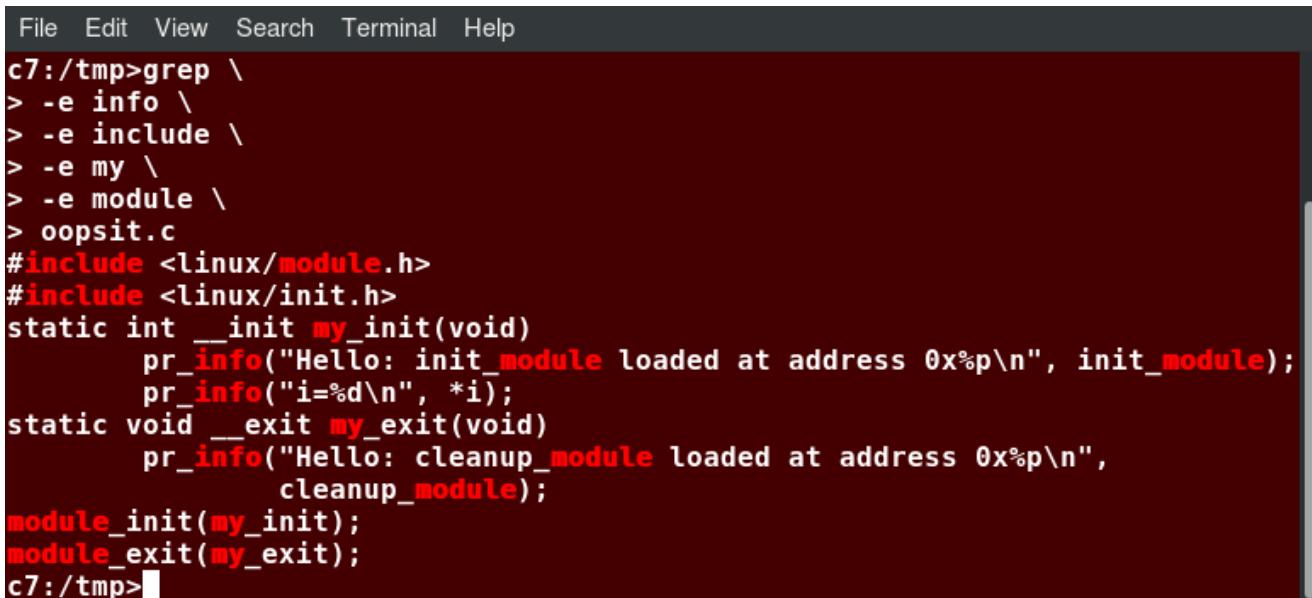
Đôi khi, các lệnh quá dài để dễ dàng gõ trên một dòng hoặc để nắm và hiểu (mặc dù không có giới hạn thực tế đối với độ dài của một dòng lệnh).

Trong trường hợp này, toán tử ghép (\), ký tự dấu gạch chéo ngược, được sử dụng để tiếp tục các lệnh dài trên một số dòng.

Dưới đây là ví dụ về lệnh cài đặt danh sách dài các gói trên hệ thống bằng quản lý gói Debian:

```
$ ~/> cd $ TRANG CHỦ  
$ ~/> sudo apt-get install autoconf automake bison build-Essential  
    chrpath curl diffstat emacs flex gcc-multilib g ++ - multilib \  
    libsdl1.2-dev libtool lzop tạo mc patch \  
    màn hình socat sudo tar texinfo tofrodos u-boot-tools giải nén \  
    vim wget xterm zip
```

Lệnh được chia thành nhiều dòng để làm cho nó dễ đọc và dễ hiểu hơn. Toán tử \ ở cuối mỗi dòng làm cho trình bao kết hợp (nối) nhiều dòng và thực thi chúng như một lệnh đơn.



```
File Edit View Search Terminal Help  
c7:/tmp>grep \  
> -e info \  
> -e include \  
> -e my \  
> -e module \  
> oopsit.c  
#include <linux/module.h>  
#include <linux/init.h>  
static int __init my_init(void)  
    pr_info("Hello: init_module loaded at address 0x%p\n", init_module);  
    pr_info("i=%d\n", *i);  
static void __exit my_exit(void)  
    pr_info("Hello: cleanup_module loaded at address 0x%p\n",  
           cleanup_module);  
module_init(my_init);  
module_exit(my_exit);  
c7:/tmp>
```

### + Đặt nhiều lệnh trên một dòng

Người dùng đôi khi cần kết hợp một số lệnh và câu lệnh và thậm chí có điều kiện thực thi chúng dựa trên hành vi của các toán tử được sử dụng ở giữa chúng. Phương pháp này được gọi là chuỗi các lệnh.

Có một số cách khác nhau để làm điều này, tùy thuộc vào những gì bạn muốn làm. Các ; Ký tự (dấu chấm phẩy) được sử dụng để phân tách các lệnh này và thực hiện chúng một

cách tuân tự, như thể chúng đã được gõ trên các dòng riêng biệt. Mỗi lệnh tiếp theo được thực thi cho dù các lệnh trước có thành công hay không.

Do đó, ba lệnh trong ví dụ sau sẽ thực hiện, ngay cả khi các lệnh trước chúng không thành công:

\$ thực hiện; thực hiện cài đặt; Làm sạch

Tuy nhiên, bạn có thể muốn hủy bỏ các lệnh tiếp theo khi một lệnh trước đó không thành công. Bạn có thể thực hiện việc này bằng cách sử dụng toán tử **&&** (và) như trong:

A screenshot of a Linux terminal window titled "student@Linux-Mint-18 ~". The terminal shows the following command and its output:

```
student@Linux-Mint-18 ~ $ cd / ; echo doing ls on / ; ls ; cd $HOME ; echo doing ls on $HOME ; ls
doing ls on /
bin dev initrd.img lost+found opt run sys var
boot etc lib media proc sbin tmp vmlinuz
cdrom home lib64 mnt root srv usr
doing ls on /home/student
Desktop Documents Downloads LFT Music Pictures Public Templates Videos
student@Linux-Mint-18 ~ $
```

\$ make && thực hiện cài đặt && làm sạch

Nếu lệnh đầu tiên thất bại, lệnh thứ hai sẽ không bao giờ được thực thi. Một sàng lọc cuối cùng là sử dụng **||** (hoặc) toán tử, như trong:

\$ tập tin mèo1 || tập tin mèo2 || tập tin mèo3

Trong trường hợp này, bạn tiến hành cho đến khi một cái gì đó thành công và sau đó bạn ngừng thực hiện bất kỳ bước tiếp theo.

### + Chuyển hướng đầu ra

A screenshot of a Linux terminal window titled "student@ubuntu: ~". The terminal shows the following command and its output:

```
student@ubuntu:~$ ls /etc/grub.d > /tmp/grubd
student@ubuntu:~$ cat /tmp/grubd
00_header
05_debian_theme
10_linux
20_linux_xen
20_memtest86+
30_os-prober
30_uefi-firmware
40_custom
41_custom
README
student@ubuntu:~$
```

Hầu hết các hệ điều hành chấp nhận đầu vào từ bàn phím và hiển thị đầu ra trên thiết bị đầu cuối. Tuy nhiên, trong kịch bản shell, bạn có thể gửi đầu ra tới một tệp. Quá trình chuyển hướng đầu ra sang một tệp

được gọi là chuyển hướng đầu ra. Chúng tôi đã sử dụng cơ sở này trong các phần trước của chúng tôi về cách sử dụng dòng lệnh.

Ký tự > được sử dụng để ghi đầu ra vào một tệp. Ví dụ: lệnh sau sẽ gửi đầu ra của lệnh free tới /tmp/free.out:

```
$ free > /tmp/free.out
```

Để kiểm tra nội dung của /tmp/free.out, tại dấu nhắc lệnh gõ cat /tmp/free.out.

Hai > ký tự (>>) sẽ nối đầu ra vào một tệp nếu nó tồn tại và hoạt động giống như > nếu tệp chưa tồn tại.

### + Chuyển hướng đầu vào

Giống như đầu ra có thể được chuyển hướng đến một tệp, đầu vào của lệnh có thể được đọc từ một tệp. Quá trình đọc đầu vào từ một tệp được gọi là chuyển hướng đầu vào và sử dụng <ký tự.

Ba lệnh sau (sử dụng chương trình wc để đếm số lượng dòng, từ và ký tự trong một tệp) là hoàn toàn tương đương và liên quan đến chuyển hướng đầu vào và một lệnh hoạt động trên nội dung của tệp:

```
$ wc </etc/passwd
```

```
49 105 2678 /etc/passwd
```

```
$ wc /etc/passwd
```

```
49 105 2678 /etc/passwd
```

```
$ cat /etc/passwd | wc
```

```
49 105 2678
```

## 8.6 Các lệnh tích hợp trong Shell

Các kịch bản lệnh Shell thực thi các chuỗi lệnh và các loại câu lệnh khác. Các lệnh này có thể là:

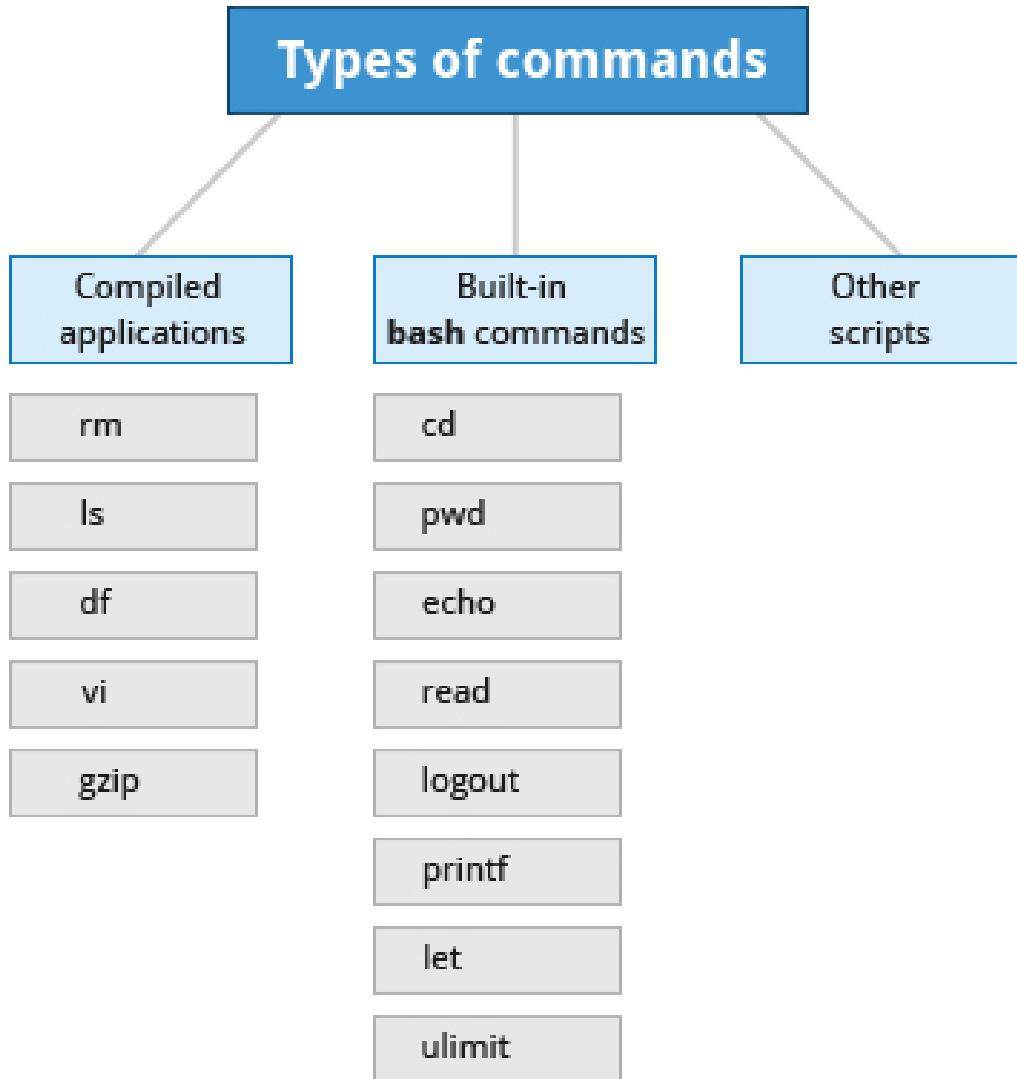
Ứng dụng tổng hợp

Các lệnh bash tích hợp

Shell script hoặc script từ các ngôn ngữ được giải thích khác, chẳng hạn như perl và Python.

Các ứng dụng được biên dịch là các tệp thực thi nhị phân, thường nằm trên hệ thống tệp trong các thư mục nổi tiếng như /usr/bin. Các tập lệnh Shell luôn có quyền truy cập vào

các ứng dụng như rm, ls, df, vi và gzip, là các chương trình được biên dịch từ các ngôn ngữ lập trình cấp thấp hơn như C.



Ngoài ra, bash có nhiều lệnh dựng sẵn, chỉ có thể được sử dụng để hiển thị đầu ra trong một shell hoặc script. Đôi khi, các lệnh này có cùng tên với các chương trình thực thi trên hệ thống, chẳng hạn như tiếng vang có thể dẫn đến các vấn đề tinh vi. Các lệnh tích hợp bash bao gồm và cd, pwd, echo, read, logout, printf, let và ulimit. Do đó, hành vi hơi khác nhau có thể được mong đợi từ phiên bản tích hợp của lệnh như echo so với /bin/echo.

Một danh sách đầy đủ các lệnh tích hợp bash có thể được tìm thấy trong trang bash man hoặc chỉ cần gõ trợ giúp, khi chúng tôi xem xét trên trang tiếp theo.

## 8.7 Các lệnh được xây dựng để chạy Bash

Chúng tôi đã liệt kê các lệnh có phiên bản được tích hợp sẵn để bash, trong cuộc thảo luận trước đây của chúng tôi về cách nhận trợ giúp trên các hệ thống Linux. Một lần nữa, đây là một ảnh chụp màn hình liệt kê chính xác các lệnh có sẵn.

```
File Edit View Search Terminal Help
c7:/home/coop>help
GNU bash, version 4.2.46(1)-release (x86_64-redhat-linux-gnu)
These shell commands are defined internally. Type 'help' to see this list.
Type 'help name' to find out more about the function `name'.
Use `info bash` to find out more about the shell in general.
Use `man -k` or `info` to find out more about commands not in this list.

A star (*) next to a name means that the command is disabled.

job_spec [&]
(( expression ))
. filename [arguments]
:
[ arg... ]
[[ expression ]]
alias [-p] [name[=value] ... ]
bg [job_spec ...]
bind [-lpvsPVS] [-m keymap] [-f filename] [-q name] [>
break [n]
builtin [shell-builtin [arg ...]]
caller [expr]
case WORD in [PATTERN [| PATTERN]...) COMMANDS ;;;...>
cd [-L|[-P [-e]]] [dir]
command [-pVv] command [arg ...]
compgen [-abcfdefgjksuv] [-o option] [-A action] [-G >
complete [-abcfdefgjksuv] [-pr] [-DE] [-o option] [-A >
compopt [-o]+o option] [-DE] [name ...]
continue [n]
coproc [NAME] command [redirections]
declare [-aAfFgilrtux] [-p] [name[=value] ...]
dirs [-clpv] [+N] [-N]
disown [-h] [-ar] [jobspec ...]
echo [-neE] [arg ...]
enable [-a] [-dnp] [-f filename] [name ...]
eval [arg ...]
exec [-cl] [-a name] [command [arguments ...]] [redir>
exit [n]
export [-fn] [name[=value] ...] or export -p
false
fc [-e ename] [-lnr] [first] [last] or fc -s [pat=rep>
fg [job_spec]
for NAME [in WORDS ... ] ; do COMMANDS; done
for (( expl; exp2; exp3 )); do COMMANDS; done
function name { COMMANDS ; } or name () { COMMANDS ; >
getopts optstring name [arg]
hash [-lr] [-p pathname] [-dt] [name ...]
help [-dms] [pattern ...]
c7:/home/coop>
```

## 8.8 Tham số tập lệnh

Người dùng thường cần truyền các giá trị tham số cho tập lệnh, chẳng hạn như tên tệp, ngày, v.v. Các tập lệnh sẽ có các đường dẫn khác nhau hoặc đến các giá trị khác nhau theo các tham số (đối số lệnh) được truyền cho chúng. Các giá trị này có thể là văn bản hoặc số như trong:

```
$ ./script.sh / tmp  
$ ./script.sh 100 200
```

Trong một tập lệnh, tham số hoặc đối số được biểu thị bằng \$ và một số hoặc ký tự đặc biệt. Bảng liệt kê một số các tham số này.

Ý nghĩa tham số

\$ 0 Tên tập lệnh

\$ 1 Tham số đầu tiên

\$ 2, \$ 3, v.v. Tham số thứ hai, thứ ba, v.v.

\$ \* Tất cả các tham số

\$ # Số lượng đối số

#### + Sử dụng tham số tập lệnh

Nếu bạn nhập tập lệnh được hiển thị trong hình, hãy tạo tập lệnh thực thi với chmod + x param.sh. Sau đó, chạy tập lệnh cho nó một vài đối số, như được hiển thị. Kịch bản được xử lý như sau:

\$ 0 in tên tập lệnh: param.sh

\$ 1 in tham số đầu tiên: một

\$ 2 in tham số thứ hai: hai

\$ 3 in tham số thứ ba: ba

\$ \* in tất cả các tham số: một hai ba bốn năm

Tuyên bố cuối cùng trở thành: Tất cả được thực hiện với param.sh

#### + Bộ chỉ huy thay thế

Đôi khi, bạn có thể cần phải thay thế kết quả của một lệnh như là một phần của lệnh khác. Nó có thể được thực hiện theo hai cách:

Bằng cách đặt lệnh bên trong trong \$ ()

Bằng cách kèm theo lệnh bên trong với backticks ()

Dạng thứ hai, backticks, không được dùng trong các tập lệnh và lệnh mới. Bất kể phương thức nào được sử dụng, lệnh được chỉ định sẽ được thực thi trong môi trường shell mới được khởi chạy và đầu ra tiêu chuẩn của shell sẽ được chèn vào nơi thực hiện thay thế lệnh.

```

File Edit View Search Terminal Help
c7:/tmp>cat param.sh
#!/bin/bash
echo "The name of this program is: $0"
echo "The first argument passed from the command line is: $1"
echo "The second argument passed from the command line is: $2"
echo "The third argument passed from the command line is: $3"
echo "All of the arguments passed from the command line are : $*"
echo
echo "All done with $0"

c7:/tmp>./param.sh one two three four five
The name of this program is: ./param.sh
The first argument passed from the command line is: one
The second argument passed from the command line is: two
The third argument passed from the command line is: three
All of the arguments passed from the command line are : one two three four five

All done with ./param.sh
c7:/tmp>

```

Hầu như bất kỳ lệnh nào cũng có thể được thực thi theo cách này. Trong khi cả hai phương thức này đều cho phép thay thế lệnh, phương thức \$ () cho phép lồng lệnh. Các kịch bản mới nên luôn luôn sử dụng phương pháp hiện đại hơn này. Ví dụ:

```
$ ls / lib / mô-đun / $ (uname -r) /
```

Trong ví dụ trên, đầu ra của lệnh uname hèr (sẽ giống như 4.18.2) được chèn vào đối số cho lệnh ls.

```

File Edit View Search Terminal Help
c7:/tmp>uname -r
4.9.0
c7:/tmp>ls /lib/modules/`uname -r`/
build      modules.alias.bin    modules.dep.bin  modules.symbols
kernel     modules.builtin     modules.devname  modules.symbols.bin
misc       modules.builtin.bin modules.order    source
modules.alias modules.dep      modules.softdep
c7:/tmp>ls /lib/modules/${(uname -r)}/
build      modules.alias.bin    modules.dep.bin  modules.symbols
kernel     modules.builtin     modules.devname  modules.symbols.bin
misc       modules.builtin.bin modules.order    source
modules.alias modules.dep      modules.softdep
c7:/tmp>

```

## 8.9 Biến môi trường

Hầu hết các tập lệnh sử dụng các biến chứa một giá trị, có thể được sử dụng ở bất cứ đâu trong tập lệnh. Các biến này có thể là người dùng hoặc hệ thống xác định. Nhiều ứng dụng sử dụng các biến môi trường như vậy (đã được đề cập chi tiết trong Chương 12: Môi trường người dùng) để cung cấp đầu vào, xác nhận và kiểm soát hành vi.

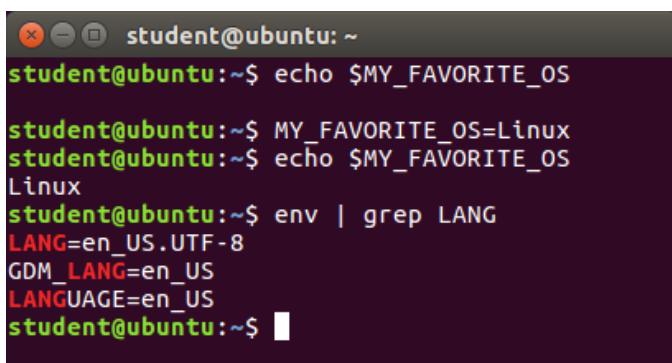
Như chúng ta đã thảo luận trước đó, một số ví dụ về các biến môi trường tiêu chuẩn là HOME, PATH và HOST. Khi được tham chiếu, các biến môi trường phải được thêm tiền tố với ký hiệu \$, như trong \$ HOME. Bạn có thể xem và đặt giá trị của các biến môi trường. Ví dụ: lệnh sau sẽ hiển thị giá trị được lưu trữ trong biến PATH:

```
$ echo $ PATH
```

Tuy nhiên, không có tiền tố được yêu cầu khi cài đặt hoặc sửa đổi giá trị biến. Ví dụ: lệnh sau đặt giá trị của biến MYCOLOR thành màu xanh:

```
$ MYCOLOR = màu xanh
```

Bạn có thể nhận được một danh sách các biến môi trường với các lệnh env, set hoặc printenv.



A screenshot of a terminal window titled "student@ubuntu: ~". The terminal shows the following commands and outputs:

```
student@ubuntu:~$ echo $MY_FAVORITE_OS
student@ubuntu:~$ MY_FAVORITE_OS=Linux
student@ubuntu:~$ echo $MY_FAVORITE_OS
Linux
student@ubuntu:~$ env | grep LANG
LANG=en_US.UTF-8
GDM_LANG=en_US
LANGUAGE=en_US
student@ubuntu:~$
```

### + Xuất khẩu biến môi trường

Mặc dù chúng tôi đã thảo luận về việc xuất các biến môi trường trong phần "Môi trường người dùng", đáng để xem xét chủ đề này trong bối cảnh viết các tập lệnh bash.

Theo mặc định, các biến được tạo trong một tập lệnh chỉ khả dụng cho các bước tiếp theo của tập lệnh đó. Bất kỳ tiến trình con (shell phụ) nào cũng không có quyền truy cập tự động vào các giá trị của các biến này. Để làm cho chúng có sẵn cho các tiến trình con, chúng phải được thăng cấp lên các biến môi trường bằng cách sử dụng câu lệnh xuất, như trong:

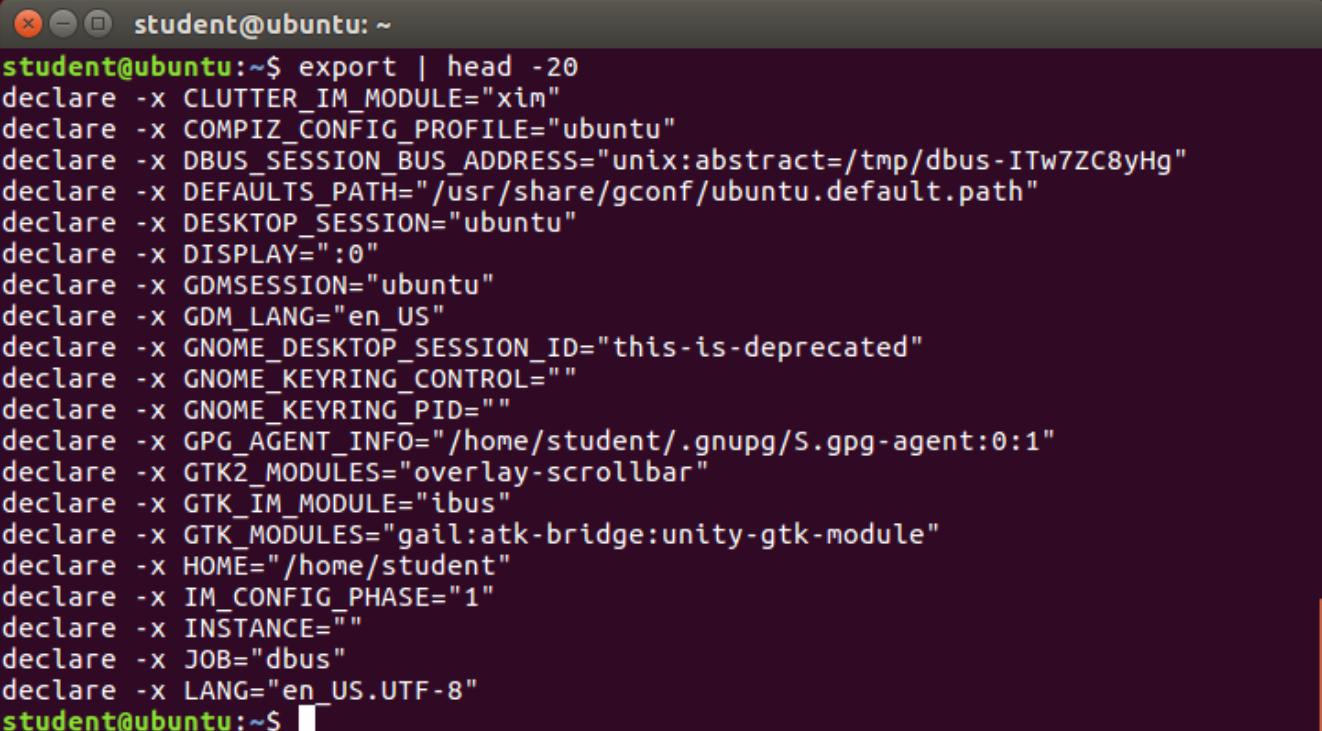
```
export VAR = value
```

hoặc là

```
VAR = value; export VAR
```

Trong khi các biến con được phép sửa đổi giá trị của các biến được xuất, thì cha mẹ sẽ không thấy bất kỳ thay đổi nào; các biến xuất không được chia sẻ, chúng chỉ được sao chép và kế thừa.

Nhập xuất không có đối số sẽ đưa ra một danh sách tất cả các biến môi trường hiện đang xuất.



```
student@ubuntu:~$ export | head -20
declare -x CLUTTER_IM_MODULE="xim"
declare -x COMPIZ_CONFIG_PROFILE="ubuntu"
declare -x DBUS_SESSION_BUS_ADDRESS="unix:abstract=/tmp/dbus-ITw7ZC8yHg"
declare -x DEFAULTS_PATH="/usr/share/gconf/ubuntu.default.path"
declare -x DESKTOP_SESSION="ubuntu"
declare -x DISPLAY=:0"
declare -x GDMSESSION="ubuntu"
declare -x GDM_LANG="en_US"
declare -x GNOME_DESKTOP_SESSION_ID="this-is-deprecated"
declare -x GNOME_KEYRING_CONTROL=""
declare -x GNOME_KEYRING_PID=""
declare -x GPG_AGENT_INFO="/home/student/.gnupg/S.gpg-agent:0:1"
declare -x GTK2_MODULES="overlay-scrollbar"
declare -x GTK_IM_MODULE="ibus"
declare -x GTK_MODULES="gail:atk-bridge:unity-gtk-module"
declare -x HOME="/home/student"
declare -x IM_CONFIG_PHASE="1"
declare -x INSTANCE=""
declare -x JOB="dbus"
declare -x LANG="en_US.UTF-8"
student@ubuntu:~$
```

## 8.10 Định nghĩa hàm trong Shell Scripting

Hàm là một khối mã thực hiện một tập hợp các hoạt động. Các hàm rất hữu ích để thực hiện các thủ tục nhiều lần, có lẽ với các biến đầu vào khác nhau. Các hàm cũng thường được gọi là chương trình con. Sử dụng các hàm trong tập lệnh yêu cầu hai bước:

Khai báo một hàm

Gọi một chức năng.

Khai báo hàm yêu cầu một tên được sử dụng để gọi nó. Cú pháp thích hợp là:

```
function_name () {
    commands
}
```

Ví dụ, hàm sau được đặt tên là display:

```
display () {  
    echo "Đây là một hàm mẫu"  
}
```

Hàm này có thể dài như mong muốn và có nhiều câu lệnh. Sau khi xác định, hàm có thể được gọi sau nhiều lần nếu cần. Trong ví dụ đây đủ được hiển thị trong hình, chúng tôi cũng đang hiển thị một sàng lọc thường được sử dụng: làm thế nào để truyền một đối số cho hàm. Đối số đầu tiên có thể được gọi là \$1, đối số thứ hai là \$2, v.v.

```
student@ubuntu:/tmp$ cat testbashfunc.sh  
#!/bin/bash  
  
showmess(){  
    echo My favorite Linux Distribution is: $1  
}  
echo ""  
showmess Ubuntu  
showmess Fedora  
showmess openSUSE  
showmess GENTOO  
showmess Slackware  
student@ubuntu:/tmp$ ./testbashfunc.sh  
  
My favorite Linux Distribution is: Ubuntu  
My favorite Linux Distribution is: Fedora  
My favorite Linux Distribution is: openSUSE  
My favorite Linux Distribution is: GENTOO  
My favorite Linux Distribution is: Slackware  
student@ubuntu:/tmp$
```

## Bài tập LAB

### 1. Viết một shell script:

1. Nhắc người dùng cho một tên thư mục và sau đó tạo nó với mkdir.
2. Thay đổi thư mục mới và in ra nơi nó đang sử dụng pwd.
3. Sử dụng cảm ứng, tạo một số tệp trống và chạy ls trên chúng để xác minh chúng trống.
4. Đặt một số nội dung trong đó bằng cách sử dụng tiếng vang và chuyển hướng.
5. Hiển thị nội dung của họ bằng cách sử dụng con mèo.

6. Nói lời tạm biệt với người dùng và làm sạch sau đó.

## 2. Viết một shell script

có chính xác một đối số và in nó trở lại đầu ra tiêu chuẩn. Hãy chắc chắn rằng tập lệnh tạo một thông báo sử dụng nếu nó được chạy mà không đưa ra một đối số.

## 3. Viết một shell script

Yêu cầu người dùng cho một số, phải là "1" hoặc "2". Bất kỳ điều vào khác sẽ dẫn đến một báo cáo lỗi.

Đặt biến môi trường thành "Có" nếu đó là "1" và "Không" nếu đó là "2".

Xuất khẩu biến môi trường và hiển thị nó.

### Viết một shell script

Yêu cầu người dùng cho một số (1, 2 hoặc 3).

Gọi một hàm với số đó trong tên của nó. Các chức năng sẽ hiển thị một thông báo với tên của nó bao gồm.

## 8.11 Kiểm tra điều kiện if

Ra quyết định có điều kiện, sử dụng câu lệnh if, là một cấu trúc cơ bản mà bất kỳ ngôn ngữ lập trình hoặc kịch bản hữu ích nào cũng phải có.

Khi một câu lệnh if được sử dụng, các hành động tiếp theo phụ thuộc vào việc đánh giá các điều kiện được chỉ định, chẳng hạn như:

So sánh số hoặc chuỗi, Trả về giá trị của lệnh  
(0 để thành công)

Tệp tồn tại hoặc quyền.

Ở dạng rút gọn, cú pháp của câu lệnh if là:

nếu KIỂM TRA; sau đó là  
CONSEQUENT-THÔNG TIN; fi

Một định nghĩa chung hơn là:

**if condition**

**then**

**statements**

**else**

**statements**



**fi**

Trong ví dụ sau, một câu lệnh if kiểm tra xem có tồn tại một tệp nào đó không và nếu tìm thấy tệp, nó sẽ hiển thị một thông báo cho biết thành công hay thất bại:

```
if [ -f "$1" ]
then
    echo file "$1 exists"
else
    echo file "$1" does not exist
fi
```

Trước tiên chúng ta cũng nên kiểm tra xem có một đối số được truyền cho tập lệnh (\$ 1) và hủy bỏ nếu không.

Lưu ý việc sử dụng dấu ngoặc vuông ([] ) để phân định điều kiện thử nghiệm. Có nhiều loại thử nghiệm khác mà bạn có thể thực hiện, chẳng hạn như kiểm tra xem hai số có bằng nhau, lớn hơn hoặc nhỏ hơn nhau hay không và đưa ra quyết định phù hợp; chúng tôi sẽ thảo luận về các thử nghiệm khác.

Trong các tập lệnh hiện đại, bạn có thể thấy dấu ngoặc kép như trong [[-f / etc / passwd]]. Đây không phải là một lỗi. Không bao giờ sai khi làm như vậy và nó tránh được một số vấn đề tinh tế, chẳng hạn như để cập đến một biến môi trường trống mà không bao quanh nó trong dấu ngoặc kép; chúng tôi sẽ không nói về điều này ở đây.

#### + Lệnh kiểm tra elif

Bạn có thể sử dụng câu lệnh elif để thực hiện các bài kiểm tra phức tạp hơn và thực hiện các hành động phù hợp. Cú pháp cơ bản là:

```
if [ sometest ] ; then
    echo Passed test1
elif [ somothertest ] ; then
    echo Passed test2
fi
```

Trong ví dụ hiển thị, chúng tôi sử dụng các bài kiểm tra chuỗi mà chúng tôi sẽ giải thích ngay sau đây và chỉ ra cách kéo vào một biến môi trường bằng câu lệnh đọc.

```
File Edit View Search Terminal Help
c7:/tmp>cat ./show_elif.sh
#!/bin/bash

echo Give your name

read name

if [ "$name" == John ] ; then
    echo Hello $name
elif [ "$name" == Ringo ] ; then
    echo Hello $name
elif [ "$name" == George ] || [ "$name" == Paul ] ; then
    echo Hello $name
else
    echo Forget it $name, You are not a Beatle
fi
c7:/tmp>./show_elif.sh
Give your name
John
Hello John
c7:/tmp>./show_elif.sh
Give your name
George
Hello George
c7:/tmp>./show_elif.sh
Give your name
Marlon
Forget it Marlon, You are not a Beatle
c7:/tmp>
```

## 8.12 Lệnh kiểm tra tập tin test

bash cung cấp một tập hợp các điều kiện tệp, có thể được sử dụng với câu lệnh if, bao gồm cả các câu lệnh trong bảng.

Bạn có thể sử dụng câu lệnh if để kiểm tra các thuộc tính tệp, chẳng hạn như:

Sự tồn tại của tập tin hoặc thư mục

Đọc hoặc viết cho phép

Thực thi cho phép.

Ví dụ, trong ví dụ sau:

**if [ -x /etc/passwd ] ; then**

**ACTION**

**fi**

câu lệnh **if** kiểm tra xem tập tin /etc/passwd có thể thực thi được không. Lưu ý thực hành rất phổ biến của việc đặt **; then** trên cùng một dòng với câu lệnh **if**.

Bạn có thể xem danh sách đầy đủ các điều kiện tập tin gõ:

### **man 1 test**

Ý nghĩa điều kiện

-e tập tin Kiểm tra nếu tập tin tồn tại.

Tập tin -d Kiểm tra nếu tập tin là một thư mục.

-f tệp Kiểm tra xem tệp có phải là tệp thông thường không (nghĩa là không phải là liên kết tượng trưng, nút thiết bị, thư mục, v.v.)

Tập tin -s Kiểm tra xem tập tin có kích thước khác không.

Tập tin -g Kiểm tra xem tập tin có tập sgid không.

-u tập tin Kiểm tra xem tập tin có được đặt suid không.

Tập tin -r Kiểm tra xem tập tin có thể đọc được không.

-w tập tin Kiểm tra nếu tập tin có thể ghi.

Tập tin -x Kiểm tra xem tập tin có thực thi được không.

### **8.13 Các biểu thức luận lý Boolean**

Các biểu thức Boolean ước tính thành TRUE hoặc FALSE và kết quả thu được bằng cách sử dụng các toán tử Boolean khác nhau được liệt kê trong bảng.

Ý nghĩa hoạt động của người vận hành

&& VÀ Hành động sẽ chỉ được thực hiện nếu cả hai điều kiện đánh giá là đúng.

|| HOẶC Hành động sẽ được thực hiện nếu bất kỳ một trong các điều kiện đánh giá là đúng.

! KHÔNG Hành động sẽ chỉ được thực hiện nếu điều kiện đánh giá là sai.

Lưu ý rằng nếu bạn có nhiều điều kiện kết hợp với toán tử &&, quá trình xử lý sẽ dừng ngay khi một điều kiện được đánh giá là sai. Ví dụ: nếu bạn có A && B && C và A là đúng nhưng B là sai, C sẽ không bao giờ được thực thi.

Tương tự như vậy, nếu bạn đang sử dụng || Toán tử, xử lý dừng ngay khi bất cứ điều gì là đúng. Ví dụ: nếu bạn có A || B || C và A là sai và B là đúng, bạn cũng sẽ không bao giờ thực hiện C.

### **+ Kiểm tra điều kiện trong các biểu thức boolean**

Biểu thức Boolean trả về TRUE hoặc FALSE. Chúng ta có thể sử dụng các biểu thức như vậy khi làm việc với nhiều loại dữ liệu, bao gồm cả chuỗi hoặc số, cũng như với các tệp.

Ví dụ: để kiểm tra xem một tệp có tồn tại hay không, hãy sử dụng kiểm tra có điều kiện sau:

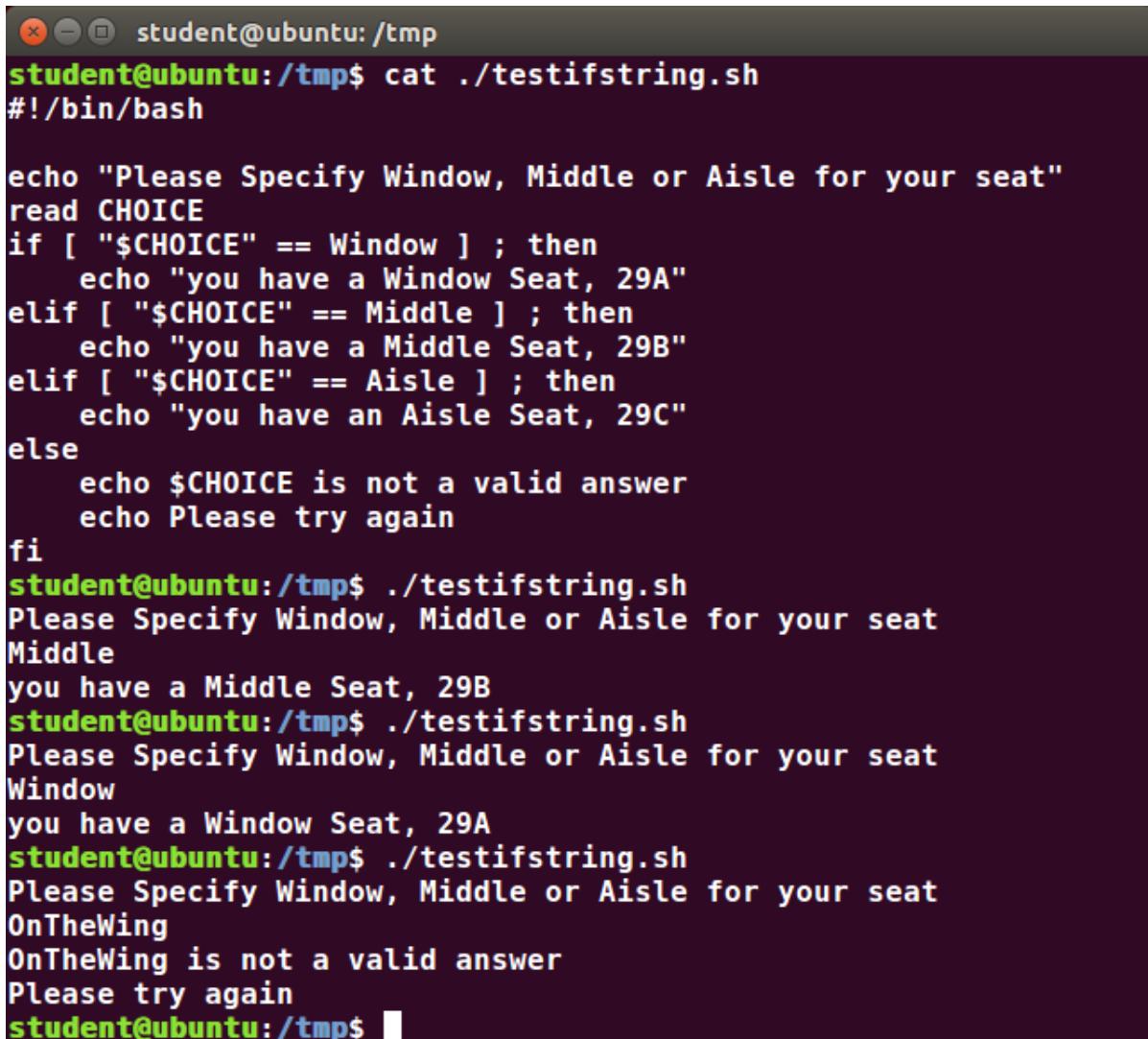
[-e <tên tệp>]

Tương tự, để kiểm tra xem giá trị của số 1 có lớn hơn giá trị của số 2 hay không, hãy sử dụng thử nghiệm có điều kiện sau:

[\$ number1 -gt \$ number2]

Toán tử -gt trả về TRUE nếu số1 lớn hơn số2.

#### + **Thí dụ kiểm tra điều kiện trong chuỗi**



```
student@ubuntu:/tmp$ cat ./testifstring.sh
#!/bin/bash

echo "Please Specify Window, Middle or Aisle for your seat"
read CHOICE
if [ "$CHOICE" == Window ] ; then
    echo "you have a Window Seat, 29A"
elif [ "$CHOICE" == Middle ] ; then
    echo "you have a Middle Seat, 29B"
elif [ "$CHOICE" == Aisle ] ; then
    echo "you have an Aisle Seat, 29C"
else
    echo $CHOICE is not a valid answer
    echo Please try again
fi
student@ubuntu:/tmp$ ./testifstring.sh
Please Specify Window, Middle or Aisle for your seat
Middle
you have a Middle Seat, 29B
student@ubuntu:/tmp$ ./testifstring.sh
Please Specify Window, Middle or Aisle for your seat
Window
you have a Window Seat, 29A
student@ubuntu:/tmp$ ./testifstring.sh
Please Specify Window, Middle or Aisle for your seat
OnTheWing
OnTheWing is not a valid answer
Please try again
student@ubuntu:/tmp$
```

Bạn có thể sử dụng câu lệnh if để so sánh các chuỗi bằng toán tử == (hai dấu bằng). Cú pháp như sau:

```
if [ string1 == string2 ] ; then
```

```
    ACTION
```

```
fi
```

Lưu ý rằng sử dụng dấu = cũng sẽ hoạt động, nhưng một số người cho rằng việc sử dụng không được chấp nhận. Bây giờ hãy xem xét một ví dụ về các chuỗi thử nghiệm.

Trong ví dụ minh họa ở đây, câu lệnh if được sử dụng để so sánh đầu vào do người dùng cung cấp và theo đó hiển thị kết quả.

### 8.14 Kiểm tra các giá trị số

Bạn có thể sử dụng các toán tử được xác định đặc biệt với câu lệnh if để so sánh các số. Các toán tử khác nhau có sẵn được liệt kê trong bảng:

Operator	Meaning
-eq	Equal to
-ne	Not equal to
-gt	Greater than
-lt	Less than
-ge	Greater than or equal to
-le	Less than or equal to

Cú pháp để so sánh các số như sau:

```
exp1 -op exp2
```

Thí dụ: Bây giờ chúng ta hãy xem xét một ví dụ về so sánh các số bằng cách sử dụng các toán tử khác nhau:

```

student@ubuntu: /tmp
#!/bin/bash
AGE=$1
if [[ $AGE -ge 20 ]] && [[ $AGE -lt 30 ]]; then
    echo "You are in your 20s"
elif [[ $AGE -ge 30 ]] && [[ $AGE -lt 40 ]]; then
    echo "You are in your 30s"
elif [[ $AGE -ge 40 ]] && [[ $AGE -lt 50 ]]; then
    echo "You are in your 40s"
else
    echo at AGE=$AGE, you are not in the proper range of 21-50
fi
student@ubuntu:/tmp$ ./checkage.sh 33
You are in your 30s
student@ubuntu:/tmp$ ./checkage.sh 21
You are in your 20s
student@ubuntu:/tmp$ ./checkage.sh 18
at AGE=18, you are not in the proper range of 21-50
student@ubuntu:/tmp$ 

```

## + Biểu thức số học

Biểu thức số học có thể được đánh giá theo ba cách sau (không gian rất quan trọng!):

Sử dụng tiện ích expr

expr là một chương trình tiêu chuẩn nhưng hơi phản đối. Cú pháp như sau:

expr 8 + 8

echo \$(expr 8 + 8)

Sử dụng cú pháp \$ ((...))

Đây là định dạng vỏ tích hợp. Cú pháp như sau:

echo \$ ((x + 1))

```

File Edit View Search Terminal Help
c7:/tmp>echo $(expr 8+8)
8+8
c7:/tmp>let x=( 8 + 8 ) ; echo $x
16
c7:/tmp>echo $((8+8))
16
c7:/tmp>

```

Sử dụng lệnh shell tích hợp cho phép. Cú pháp như sau:

set x = (1 + 2); echo \$ x

Trong các kịch bản shell hiện đại, việc sử dụng expr được thay thế tốt hơn bằng var = \$ ((...)).

## Bài tập LAB : Phép toán số học và hàm

Viết một kịch bản sẽ hoạt động như một máy tính đơn giản để cộng, trừ, nhân và chia.

Mỗi hoạt động nên có chức năng riêng của nó.

Bất kỳ phương pháp nào trong ba phương pháp cho số học bash, (\$ ((..))), let hoặc expr đều có thể được sử dụng.

Người dùng nên đưa ra 3 đối số khi thực thi tập lệnh:

Đầu tiên phải là một trong các chữ cái a, s, m hoặc d để chỉ định phép toán nào.

Đối số thứ hai và thứ ba phải là các số đang được vận hành.

Tập lệnh sẽ phát hiện các giá trị đầu vào xấu hoặc thiếu và hiển thị kết quả khi hoàn tất.

## 8.15 Thao tác với các chuỗi

Hãy để sâu hơn và tìm hiểu làm thế nào để làm việc với các chuỗi trong các shell script.

Một biến chuỗi chứa một chuỗi các ký tự văn bản. Nó có thể bao gồm chữ cái, số, ký hiệu và dấu chấm câu. Một số ví dụ bao gồm: abcde, 123, abcde 123, abcde-123, & acbde =% 123.

Toán tử chuỗi bao gồm những toán tử thực hiện so sánh, sắp xếp và tìm độ dài. Bảng sau đây cho thấy việc sử dụng một số toán tử chuỗi cơ bản:

Ý nghĩa toán tử

[[string1 > string2]] So sánh thứ tự sắp xếp của chuỗi1 và chuỗi2.

[[string1 == string2]] So sánh các ký tự trong chuỗi1 với các ký tự trong chuỗi2.

myLen1 = \${# string1} Lưu độ dài của chuỗi1 trong biến myLen1.

Thí dụ:

Trong ví dụ đầu tiên, chúng tôi so sánh chuỗi đầu tiên với chuỗi thứ hai và hiển thị một thông báo phù hợp bằng cách sử dụng câu lệnh if.

+ Trong ví dụ thứ hai, chúng ta chuyển vào một tên tệp và xem liệu tệp đó có tồn tại trong thư mục hiện tại hay không.

### + Phần của chuỗi

Đôi khi, bạn có thể không cần so sánh hoặc sử dụng toàn bộ chuỗi. Để trích xuất n ký tự đầu tiên của chuỗi, chúng ta có thể chỉ định: \$ {chuỗi: 0: n}. Ở đây, 0 là phần bù trong chuỗi (nghĩa là bắt đầu từ ký tự nào) trong đó phần trích xuất cần bắt đầu và n là số lượng ký tự được trích xuất.

Để trích xuất tất cả các ký tự trong một chuỗi sau dấu chấm (.), Sử dụng biểu thức sau: \$ {chuỗi # \* }.

```

File Edit View Search Terminal Help
c7:/tmp>cat str_demo1.sh
#!/bin/bash
echo Give two strings to compare:
echo ""
read str1 str2
echo ""
if [ "$str1" = "$str2" ] ; then
    echo "The first string: $str1, is the same as the second string: $str2"
else
    echo "The first string: $str1, is not the same as the second string: $str2"
fi
c7:/tmp>./str_demo1.sh
Give two strings to compare:

dog cat

The first string: dog, is not the same as the second string: cat
c7:/tmp>./str_demo1.sh
Give two strings to compare:

parakeet parakeet

The first string: parakeet, is the same as the second string: parakeet
c7:/tmp>

```

```

File Edit View Search Terminal Help
c7:/tmp>NAME=Eddie.Haskel
c7:/tmp>first=${NAME:0:5} ; echo first name = $first
first name = Eddie
c7:/tmp>last=${NAME#*.} ; echo last name = $last
last name = Haskel
c7:/tmp>

```

## Bài tập LAB

+ Viết một kịch bản đọc hai chuỗi làm đối số và sau đó:

Kiểm tra xem chuỗi đầu tiên có độ dài bằng không và nếu chuỗi kia có độ dài khác không, cho người dùng biết cả hai kết quả.

Xác định độ dài của mỗi chuỗi và báo cáo chuỗi nào dài hơn hoặc nếu chúng có độ dài bằng nhau.

So sánh các chuỗi để xem nếu chúng giống nhau và báo cáo kết quả.

## 8.16 Biểu thức case

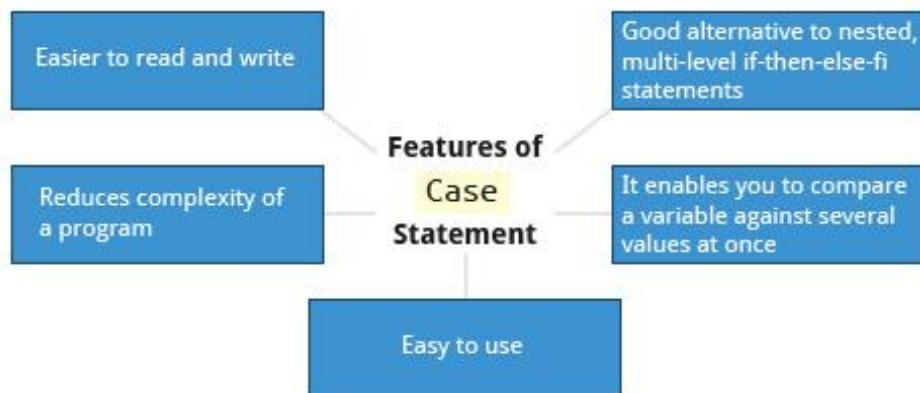
Câu lệnh tình huống được sử dụng trong các tình huống trong đó giá trị thực của một biến có thể dẫn đến các đường dẫn thực thi khác nhau. báo cáo trường hợp thường được sử dụng để xử lý các tùy chọn dòng lệnh. Dưới đây là một số lợi thế của việc sử dụng báo cáo trường hợp:

Nó dễ đọc và viết hơn.

Nó là một thay thế tốt cho các khối mã if-then-other-fi lồng nhau, đa cấp.

Nó cho phép bạn so sánh một biến với nhiều giá trị cùng một lúc.

Nó làm giảm sự phức tạp của một chương trình.



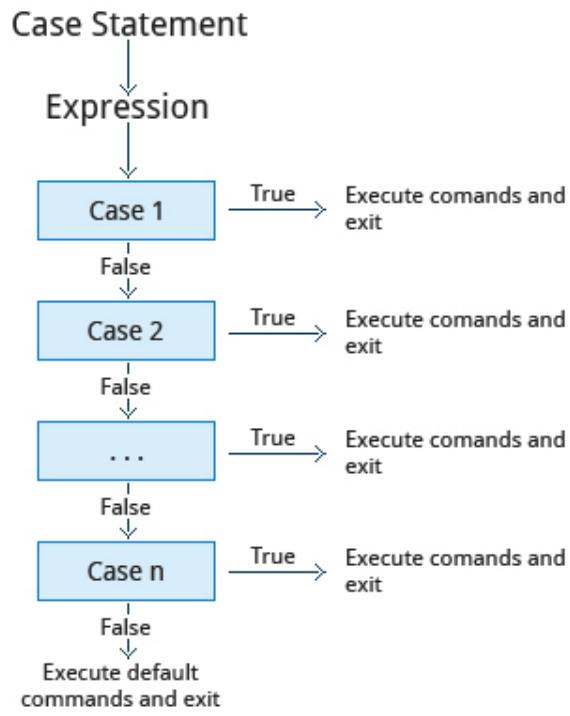
## Cấu trúc của biểu thức case

**case expression in**

```
pattern1) execute commands;;
pattern2) execute commands;;
pattern3) execute commands;;
pattern4) execute commands;;
* )   execute some default commands
or nothing ;;
esac
```

## Thí dụ sử dụng

Dưới đây là một ví dụ về việc sử dụng cấu trúc trường hợp. Lưu ý rằng bạn có thể có nhiều khả năng cho mỗi giá trị trường hợp thực hiện cùng một hành động.



```

File Edit View Search Terminal Help
c7:/tmp>cat testcase.sh
#!/bin/sh

echo "Do you want to destroy your entire file system?"
read response

case "$response" in
    "yes") echo "I hope you know what you are doing!";
              echo "I am supposed to type: rm -rf /";
              echo "But I refuse to let you commit suicide";;
    "no" ) echo "You have some common sense! Aborting...";;
    "y" | "Y" | "YES" ) echo "I hope you know what you are doing!";
              echo "I am supposed to type: rm -rf /";
              echo "But I refuse to let you commit suicide";;
    "n" | "N" | "NO" ) echo "You have some common sense! Aborting...";;
    * ) echo "You have to give an answer!";;
esac
exit 0
c7:/tmp>./testcase.sh
Do you want to destroy your entire file system?
NO
You have some common sense! Aborting...
c7:/tmp>./testcase.sh
Do you want to destroy your entire file system?
y
I hope you know what you are doing!
I am supposed to type: rm -rf /
But I refuse to let you commit suicide
c7:/tmp>■

```

+ Sử dụng case

Viết một tập lệnh lấy tham số một tháng dưới dạng số (nghĩa là từ 1 đến 12) và dịch mã này sang tên tháng và hiển thị kết quả trên tiêu chuẩn (thiết bị đầu cuối).

Nếu không có đối số nào được đưa ra hoặc một số xấu được đưa ra, tập lệnh sẽ báo cáo lỗi và thoát.

### 8.17 Cấu trúc vòng lặp

Bằng cách sử dụng các cấu trúc lặp, bạn có thể thực thi một hoặc nhiều dòng mã lặp đi lặp lại, thường là trên một lựa chọn các giá trị của dữ liệu, chẳng hạn như các tệp riêng lẻ. Thông thường, bạn làm điều này cho đến khi một bài kiểm tra có điều kiện trả về đúng hoặc sai, theo yêu cầu.

Ba loại vòng lặp thường được sử dụng trong hầu hết các ngôn ngữ lập trình:

**for**  
**while**  
**until**

Tất cả các vòng lặp này dễ dàng được sử dụng để lặp lại một tập hợp các câu lệnh cho đến khi điều kiện thoát là đúng.

+ **The for Loop:** vòng lặp for hoạt động trên từng thành phần của danh sách các mục. Cú pháp của vòng lặp for là:

**for variable-name in list**  
**do**  
    **execute one iteration for each item in the list until the list is finished**  
**done**

Trong trường hợp này, tên biến và danh sách được thay thế bởi bạn khi thích hợp (xem ví dụ). Cũng như các cấu trúc lặp khác, các câu lệnh được lặp lại phải được bao quanh bằng cách làm và thực hiện. Ảnh chụp màn hình ở đây hiển thị một ví dụ về vòng lặp for để in tổng các số từ 1 đến 10.

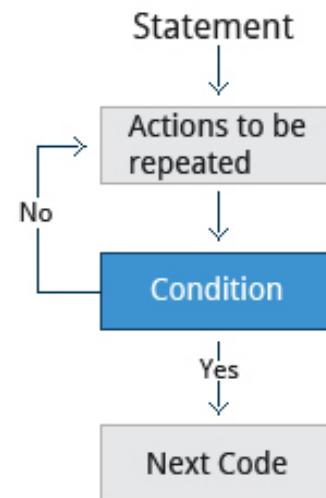
```
File Edit View Search Terminal Help
c7:/tmp>cat testfor.sh
#!/bin/sh

sum=0

for j in 1 2 3 4 5 6 7 8 9 10
do
    sum=$(( $sum + $j ))
done
echo The sum is: $sum
echo The sum of numbers from 1 to n is : 'n*(n+1)/2'
echo Check Value = $(( ($j*($j+1))/2 ))
exit 0

c7:/tmp>./testfor.sh
The sum is: 55
The sum of numbers from 1 to n is : n*(n+1)/2
Check Value = 55
c7:/tmp>
```

+ **The while Loop**



Vòng lặp while lặp lại một tập hợp các câu lệnh miễn là lệnh điều khiển trả về true. Cú pháp là:

```
while condition is true
do
    Commands for execution
    ----
done
```

Tập hợp các lệnh cần được lặp lại phải được đặt giữa làm và thực hiện. Bạn có thể sử dụng bất kỳ lệnh hoặc toán tử làm điều kiện. Thông thường, nó được đặt trong dấu ngoặc vuông ([]). Ảnh chụp màn hình ở đây cho thấy một ví dụ về vòng lặp while tính toán giai thừa của một số. Bạn có biết tại sao tính toán của 21! cho kết quả xâu?

```
File Edit View Search Terminal Help
c7:/tmp>cat testwhile.sh
#!/bin/sh
n=$1
[ "$n" == "" ] && echo please give a number and try again && exit
factorial=1 ; j=1

while [ $j -le $n ]
do
    factorial=$(( $factorial * $j ))
    j=$((j+1))
done
echo The factorial of $n, "$n"! = $factorial
exit 0

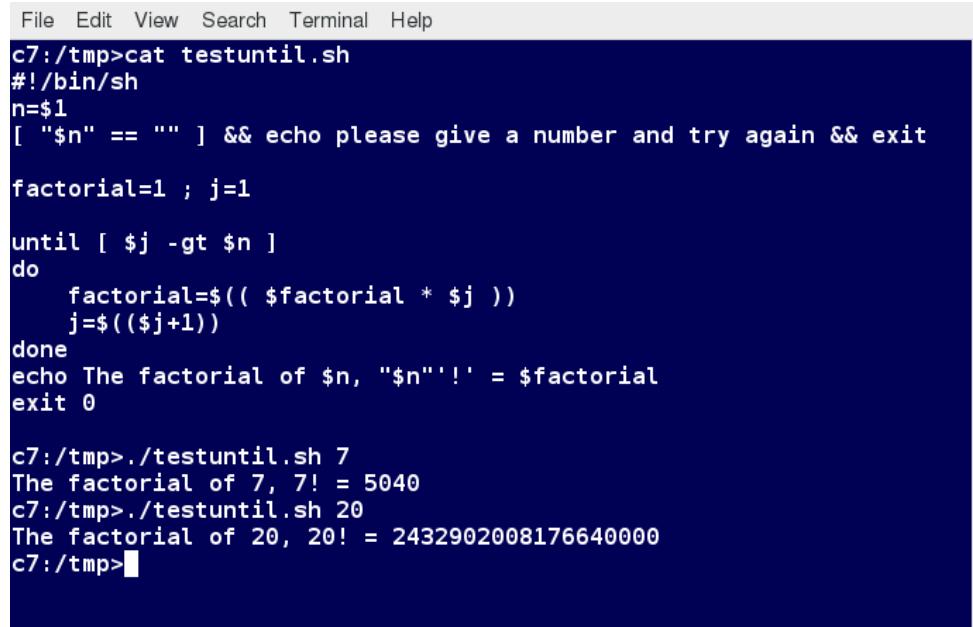
c7:/tmp>./testwhile.sh 6
The factorial of 6, 6! = 720
c7:/tmp>./testwhile.sh
please give a number and try again
c7:/tmp>./testwhile.sh 20
The factorial of 20, 20! = 2432902008176640000
c7:/tmp>./testwhile.sh 21
The factorial of 21, 21! = -4249290049419214848
c7:/tmp>
```

### + The until Loop

Vòng lặp cho đến khi lặp lại một tập hợp các câu lệnh miễn là lệnh điều khiển là sai. Vì vậy, về cơ bản nó trái ngược với vòng lặp while. Cú pháp là:

```
until condition is false
do
    Commands for execution
    ----
done
```

Tương tự như vòng lặp while, tập hợp các lệnh cần lặp lại phải được đặt giữa làm và thực hiện. Bạn có thể sử dụng bất kỳ lệnh hoặc toán tử làm điều kiện. Ảnh chụp màn hình ở đây cho thấy ví dụ về vòng lặp cho đến khi một lần nữa tính toán các yếu tố; nó chỉ khác một chút so với trường hợp thử nghiệm cho vòng lặp while.



```
File Edit View Search Terminal Help
c7:/tmp>cat testuntil.sh
#!/bin/sh
n=$1
[ "$n" == "" ] && echo please give a number and try again && exit
factorial=1 ; j=1
until [ $j -gt $n ]
do
    factorial=$(( $factorial * $j ))
    j=$((j+1))
done
echo The factorial of $n, "$n"! = $factorial
exit 0

c7:/tmp>./testuntil.sh 7
The factorial of 7, 7! = 5040
c7:/tmp>./testuntil.sh 20
The factorial of 20, 20! = 2432902008176640000
c7:/tmp>
```

## 8.18 Gỡ lỗi bash Script

Trong khi làm việc với các tập lệnh và lệnh, bạn có thể gặp lỗi. Đây có thể là do lỗi trong tập lệnh, chẳng hạn như cú pháp không chính xác hoặc các thành phần khác, chẳng hạn như tệp bị thiếu hoặc không đủ quyền để thực hiện thao tác. Những lỗi này có thể được báo cáo với một mã lỗi cụ thể, nhưng thường chỉ mang lại đầu ra không chính xác hoặc gây nhầm lẫn. Vì vậy, làm thế nào để bạn đi về xác định và sửa lỗi?

Gỡ lỗi giúp bạn khắc phục sự cố và giải quyết các lỗi đó và là một trong những nhiệm vụ quan trọng nhất mà quản trị viên hệ thống thực hiện.

### + Chế độ gỡ lỗi

Trước khi sửa lỗi (hoặc lỗi), điều quan trọng là phải biết nguồn của nó.

Bạn có thể chạy tập lệnh bash trong chế độ gỡ lỗi bằng cách thực hiện bash mậtx ./script\_file hoặc đặt dấu ngoặc cho các phần của tập lệnh với set -x và set + x. Chế độ gỡ lỗi giúp xác định lỗi vì:

Nó theo dõi và tiền tố mỗi lệnh với ký tự +.

Nó hiển thị từng lệnh trước khi thực hiện nó.

Nó chỉ có thể gỡ lỗi các phần được chọn của tập lệnh (nếu muốn) với:

```
set -x # bật gõ lỗi
```

```
...
```

```
set +x # tắt gõ lỗi
```

Ảnh chụp màn hình hiển thị ở đây cho thấy một tập lệnh chạy trong chế độ gõ lỗi nếu chạy với bất kỳ đối số nào trên dòng lệnh.

```
File Edit View Search Terminal Help
c7:/tmp>cat shdebug.sh
#!/bin/sh

if [ "$1" != "" ] ; then set -x ; fi

echo "Do you want to destroy your entire file system?"
read response

case "$response" in
    "yes")          echo "I hope you know what you are doing!" ;
                    echo "I am supposed to type: rm -rf /";
                    echo "But I refuse to let you commit suicide";;

    "no" )          echo "You have some common sense! Aborting..." ;;

    "y" | "Y" | "YES" ) echo "I hope you know what you are doing!" ;
                        echo "I am supposed to type: rm -rf /";
                        echo "But I refuse to let you commit suicide";;

    "n" | "N" | "NO" ) echo "You have some common sense! Aborting..." ;;

    * )             echo "You have to give an answer!" ;;

esac
exit 0
c7:/tmp>shdebug.sh debug
+ echo 'Do you want to destroy your entire file system?'
Do you want to destroy your entire file system?
+ read response
no
+ case "$response" in
+ echo 'You have some common sense! Aborting...'
You have some common sense! Aborting...
+ exit 0
c7:/tmp>shdebug.sh
Do you want to destroy your entire file system?
no
You have some common sense! Aborting...
c7:/tmp>
```

### + Chuyển hướng lỗi sang tập tin và màn hình

Trong UNIX / Linux, tất cả các chương trình chạy được cung cấp ba luồng tệp mở khi chúng được khởi động như được liệt kê trong bảng:

## Tập hồ sơ

### Mô tả tập tin mô tả

Theo tiêu chuẩn stdin, theo mặc định, bàn phím / thiết bị đầu cuối cho các chương trình chạy từ dòng lệnh 0

đầu ra tiêu chuẩn, theo mặc định, màn hình cho các chương trình chạy từ dòng lệnh 1

stderr Lỗi tiêu chuẩn, trong đó thông báo lỗi đầu ra được hiển thị hoặc lưu 2

Sử dụng chuyển hướng, chúng ta có thể lưu luồng đầu ra stdout và stderr vào một tệp hoặc hai tệp riêng biệt để phân tích sau khi chương trình hoặc lệnh được thực thi.

Ảnh chụp màn hình hiển thị tập lệnh shell với một lỗi đơn giản, sau đó được chạy và đầu ra lỗi được chuyển hướng sang error.log. Sử dụng cat để hiển thị nội dung của nhật ký lỗi thêm vào gỡ lỗi. Bạn có thấy cách sửa tập lệnh không?

```
student@openSUSE:/tmp
File Edit View Search Terminal Help
student@openSUSE:/tmp>
student@openSUSE:/tmp> cat testbasherror.sh
#!/bin/bash

sum=0
for i in 1 2 3 4
do
    sum=$((sum+$i))
done
echo "the sum of \"$i\" numbers is $sum"
ls error
student@openSUSE:/tmp> ./testbasherror.sh 2> error.log
student@openSUSE:/tmp> cat error.log
./testbasherror.sh: line 6: syntax error near unexpected token `('
./testbasherror.sh: line 6: `    sum=$((sum+$i))'
./testbasherror.sh: line 7: syntax error near unexpected token `done'
./testbasherror.sh: line 7: `done'
student@openSUSE:/tmp>
```

### 8.19 Tạo tập tin tạm thời và thư mục

Hãy xem xét một tình huống mà bạn muốn lấy 100 bản ghi từ một tệp có 10.000 bản ghi. Bạn sẽ cần một nơi để lưu trữ thông tin được trích xuất, có thể trong một tệp tạm thời, trong khi bạn xử lý thêm về nó.

Các tệp tạm thời (và thư mục) có nghĩa là lưu trữ dữ liệu trong một thời gian ngắn. Thông thường, người ta sắp xếp nó để các tệp này biến mất khi chương trình sử dụng chúng kết thúc. Mặc dù bạn cũng có thể sử dụng cảm ứng để tạo một tệp tạm thời, nhưng trong một số trường hợp, điều này có thể giúp tin tức dễ dàng truy cập vào dữ liệu của bạn. Điều này đặc biệt đúng nếu tên và vị trí tệp của tệp tạm thời có thể dự đoán được.

Cách thực hành tốt nhất là tạo tên tệp ngẫu nhiên và không thể đoán trước để lưu trữ tạm thời. Một cách để làm điều này là với tiện ích mktemp, như trong các ví dụ sau.

XXXXXXX được thay thế bởi tiện ích mktemp bằng các ký tự ngẫu nhiên để đảm bảo tên của tệp tạm thời không thể dự đoán dễ dàng và chỉ được biết trong chương trình của bạn.

Command	Usage
<code>TEMP=\$(mktemp /tmp/tempfile.XXXXXXXX)</code>	To create a temporary file
<code>TEMPDIR=\$(mktemp -d /tmp/tempdir.XXXXXXXX)</code>	To create a temporary directory

#### + Ví dụ về việc tạo một tệp và thư mục tạm thời

Sự chậm chạp trong việc tạo các tệp tạm thời có thể dẫn đến thiệt hại thực sự, do vô tình hoặc nếu có một diễn viên độc hại. Ví dụ: nếu ai đó tạo liên kết tượng trưng từ tệp tạm thời đã biết được sử dụng bởi root đến tệp / etc / passwd, như sau:

```
$ ln -s /etc/passwd /tmp/tempfile
```

Có thể có một vấn đề lớn nếu một tập lệnh chạy bằng root có một dòng như thế này:

```
echo $VAR> /tmp/tempfile
```

Tệp mật khẩu sẽ bị ghi đè bởi nội dung tệp tạm thời.

Để ngăn chặn tình huống như vậy, hãy đảm bảo bạn chọn ngẫu nhiên tên tệp tạm thời của mình bằng cách thay thế dòng trên bằng các dòng sau:

```
TEMP = $(mktemp /tmp/tempfile.XXXXXXXX)
```

```
echo $VAR> $TEMP
```

Lưu ý chụp màn hình hiển thị các tệp tạm thời được đặt tên tương tự từ các ngày khác nhau, nhưng với các ký tự được tạo ngẫu nhiên trong đó.

```
student@CentOS7:/tmp
File Edit View Search Terminal Help
[student@CentOS7 tmp]$ /bin/ls -F
hsperfdata_root/
ssh-0n2bLK5mQA48/
ssh-z4dCINvYrZKH/
systemd-private-423202d25edd4d40bf141f42a8864246-colord.service-IEldnV/
systemd-private-423202d25edd4d40bf141f42a8864246-cups.service-4Iuhbe/
systemd-private-423202d25edd4d40bf141f42a8864246-rtkit-daemon.service-KjEQCz/
systemd-private-423202d25edd4d40bf141f42a8864246-vmtoolsd.service-UwKRYL/
systemd-private-ca46f7c53b2d4c70924f3adec942f026-colord.service-l0tpFP/
systemd-private-ca46f7c53b2d4c70924f3adec942f026-cups.service-fo9FLw/
systemd-private-ca46f7c53b2d4c70924f3adec942f026-rtkit-daemon.service-G6mbNV/
systemd-private-ca46f7c53b2d4c70924f3adec942f026-vmtoolsd.service-SgkkJB/
tracker-extract-files.1000/
[student@CentOS7 tmp]$
```

## 8.20 Loại bỏ đầu ra với / dev / null

Một số lệnh nhất định (như find) sẽ tạo ra lượng đầu ra khổng lồ, có thể áp đảo giao diện điều khiển. Để tránh điều này, chúng ta có thể chuyển hướng đầu ra lớn đến một tệp đặc biệt (nút thiết bị) được gọi là / dev / null. Giả hành này cũng được gọi là xô bit hoặc lỗ đen.

Tất cả dữ liệu được ghi vào nó bị loại bỏ và các thao tác ghi không bao giờ trả về một điều kiện thất bại. Sử dụng các toán tử chuyển hướng thích hợp, nó có thể làm cho đầu ra biến mất khỏi các lệnh thường tạo đầu ra thành thiết bị xuất chuẩn và / hoặc stderr:

```
$ ls -lR / tmp> / dev / null
```

Trong lệnh trên, toàn bộ luồng đầu ra tiêu chuẩn bị bỏ qua, nhưng mọi lỗi sẽ vẫn xuất hiện trên bàn điều khiển. Tuy nhiên, nếu có:

```
$ ls -lR / tmp> & / dev / null
```

cả stdout và stderr sẽ được kết xuất vào / dev / null.

## 8.21 Số ngẫu nhiên và dữ liệu

Nó thường hữu ích để tạo số ngẫu nhiên và dữ liệu ngẫu nhiên khác khi thực hiện các tác vụ như:

Thực hiện các nhiệm vụ liên quan đến bảo mật

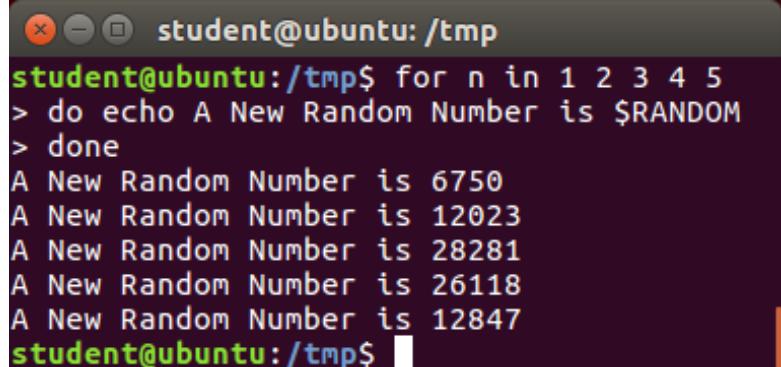
Reinitializing thiết bị lưu trữ

Xóa và / hoặc che khuất dữ liệu hiện có

Tạo dữ liệu vô nghĩa được sử dụng cho các bài kiểm tra.

```
File Edit View Search Terminal Help
c7:/tmp>ls -lR > /dev/null
ls: cannot open directory ./pcp.ibMBphQI3: Permission denied
ls: cannot open directory ./systemd-private-dee8b3ebc6184b2aa5d894c4bd54
  cfd0-colord.service-nLrMij: Permission denied
ls: cannot open directory ./systemd-private-dee8b3ebc6184b2aa5d894c4bd54
  cfd0-cups.service-Vm42ql: Permission denied
ls: cannot open directory ./systemd-private-dee8b3ebc6184b2aa5d894c4bd54
  cfd0-httdp.service-9nSoUU: Permission denied
ls: cannot open directory ./vmware-root: Permission denied
c7:/tmp>
c7:/tmp>
c7:/tmp>ls -lR >& /dev/null
c7:/tmp>
```

Các số ngẫu nhiên như vậy có thể được tạo bằng cách sử dụng biến môi trường \$ RANDOM, được lấy từ trình tạo số ngẫu nhiên tích hợp của nhân Linux hoặc bởi hàm thư viện OpenSSL, sử dụng thuật toán Trin140 (Tiêu chuẩn xử lý thông tin liên bang) để tạo các số ngẫu nhiên để mã hóa



```
student@ubuntu: /tmp
student@ubuntu:/tmp$ for n in 1 2 3 4 5
> do echo A New Random Number is $RANDOM
> done
A New Random Number is 6750
A New Random Number is 12023
A New Random Number is 28281
A New Random Number is 26118
A New Random Number is 12847
student@ubuntu:/tmp$
```

Để tìm hiểu về Trin140, hãy đọc bài viết "Trin 140-2" của Wikipedia.

Ví dụ cho bạn thấy cách dễ dàng sử dụng phương pháp biến môi trường để tạo số ngẫu nhiên.

### + Cách mà nhân Linux tạo số ngẫu nhiên

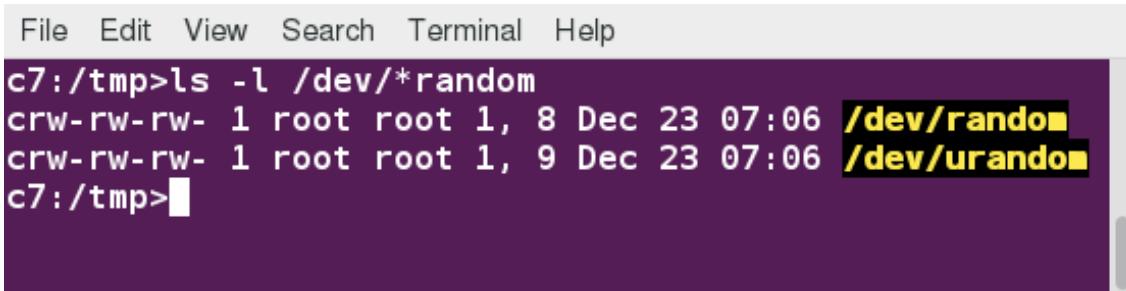
Một số máy chủ có bộ tạo số ngẫu nhiên phần cứng lấy đầu vào các loại tín hiệu nhiều khác nhau, chẳng hạn như nhiệt và hiệu ứng quang điện. Đầu dò chuyển đổi nhiều này thành tín hiệu điện, một lần nữa được chuyển đổi thành số kỹ thuật số bằng bộ chuyển đổi A-D. Con số này được coi là ngẫu nhiên. Tuy nhiên, hầu hết các máy tính phổ biến không chứa phần cứng chuyên dụng như vậy và thay vào đó, dựa vào các sự kiện được tạo trong quá trình khởi động để tạo dữ liệu thô cần thiết.

Bất kể nguồn nào trong hai nguồn này được sử dụng, hệ thống vẫn duy trì cái gọi là nhóm entropy của các số kỹ thuật số / bit ngẫu nhiên này. Số ngẫu nhiên được tạo ra từ nhóm entropy này.

Nhân Linux cung cấp các nút thiết bị / dev / ngẫu nhiên và / dev / urandom, được vẽ trên nhóm entropy để cung cấp các số ngẫu nhiên được rút ra từ số bit tiếng ồn ước tính trong nhóm entropy.

/dev/random được sử dụng khi yêu cầu tính ngẫu nhiên chất lượng rất cao, chẳng hạn như pad một lần hoặc tạo khóa, nhưng việc cung cấp các giá trị tương đối chậm. / dev / urandom nhanh hơn và phù hợp (đủ tốt) cho hầu hết các mục đích mã hóa.

Hơn nữa, khi nhóm entropy trống, / dev / random bị chặn và không tạo ra bất kỳ số nào cho đến khi tiếng ồn môi trường bổ sung (lưu lượng mạng, di chuyển chuột, v.v.) được thu thập, trong khi / dev / urandom sử dụng lại nhóm nội bộ để tạo ra nhiều hơn bit giả ngẫu nhiên.



```
File Edit View Search Terminal Help
c7:/tmp>ls -l /dev/*random
crw-rw-rw- 1 root root 1, 8 Dec 23 07:06 /dev/random
crw-rw-rw- 1 root root 1, 9 Dec 23 07:06 /dev/urandom
c7:/tmp>
```

### Bài tập LAB Số ngẫu nhiên

Viết một kịch bản:

Lấy một từ làm đối số.

Nối một số ngẫu nhiên cho nó.

Hiển thị câu trả lời.

## 8.22 Câu hỏi ôn tập Chương 8

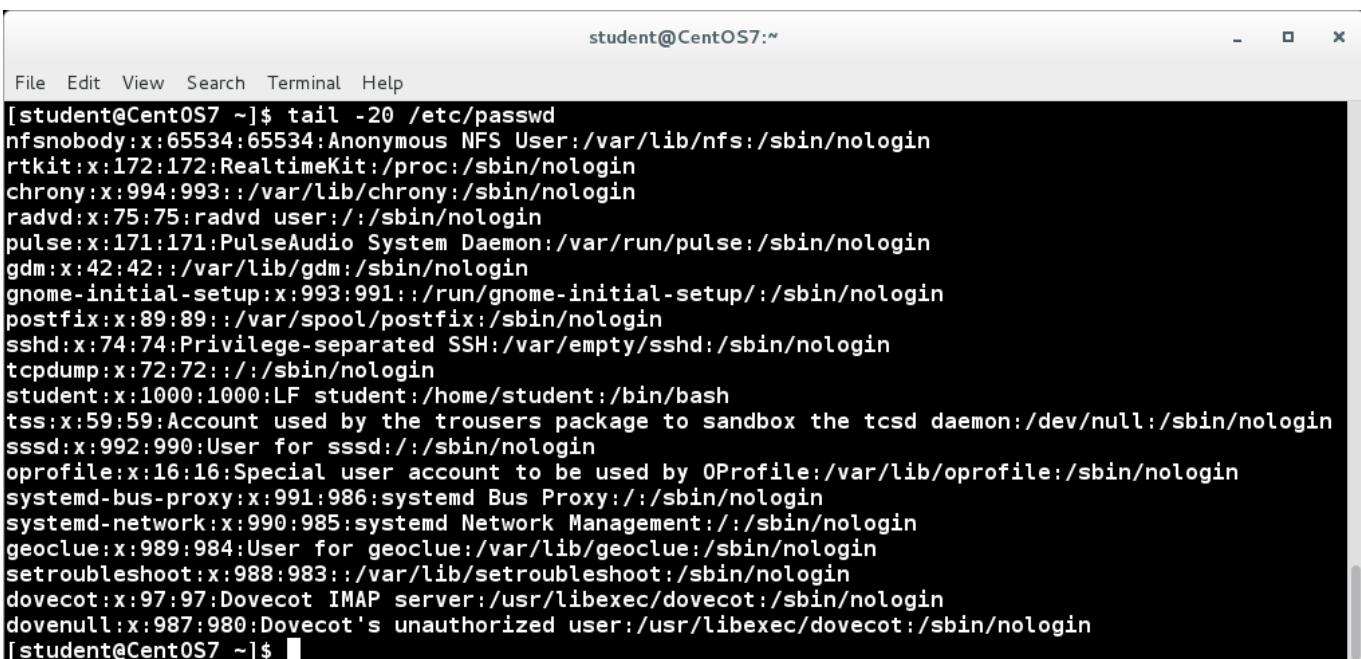
- Giải thích các tính năng và khả năng của bash shell scripting.
- Biết cú pháp cơ bản của các câu lệnh scripting.
- Hãy làm quen với các phương pháp và cấu trúc khác nhau được sử dụng.
- Kiểm tra các thuộc tính và sự tồn tại của các tập tin và các đối tượng khác.
- Sử dụng các câu lệnh có điều kiện, chẳng hạn như các khối if-then-other.
- Thực hiện các phép toán số học bằng ngôn ngữ scripting.
- Thao tác các chuỗi để thực hiện các hành động như so sánh và sắp xếp.

8. Sử dụng các biểu thức Boolean khi làm việc với nhiều loại dữ liệu, bao gồm cả chuỗi hoặc số, cũng như các tệp.
9. Sử dụng các câu lệnh tình huống để xử lý các tùy chọn dòng lệnh.
10. Sử dụng các cấu trúc lặp để thực thi một hoặc nhiều dòng mã lặp đi lặp lại.
11. Các kịch bản gỡ lỗi sử dụng set -x và set + x.
12. Tạo tập tin và thư mục tạm thời.
13. Tạo và sử dụng số ngẫu nhiên.

# Chương 9 Bảo mật Hệ điều hành Linux

## 9.1 Tài khoản người dùng

Nhân Linux cho phép người dùng xác thực đúng cách truy cập các tệp và ứng dụng. Mặc dù mỗi người dùng được xác định bởi một số nguyên duy nhất (id người dùng hoặc UID), một cơ sở dữ liệu riêng biệt liên kết tên người dùng với mỗi UID. Khi tạo tài khoản, thông tin người dùng mới được thêm vào cơ sở dữ liệu người dùng và thư mục chính của người dùng phải được tạo và điền vào một số tệp cần thiết. Các chương trình dòng lệnh như useradd và userdel cũng như các công cụ GUI được sử dụng để tạo và xóa tài khoản.



```
student@CentOS7:~$ tail -20 /etc/passwd
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
chrony:x:994:993:/var/lib/chrony:/sbin/nologin
radvd:x:75:75:radvd user:/sbin/nologin
pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
gdm:x:42:42:/var/lib/gdm:/sbin/nologin
gnome-initial-setup:x:993:991:/run/gnome-initial-setup:/sbin/nologin
postfix:x:89:89:/var/spool/postfix:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
tcpdump:x:72:72::/sbin/nologin
student:x:1000:1000:LF student:/home/student:/bin/bash
tss:x:59:59:Account used by the trousers package to sandbox the tcscd daemon:/dev/null:/sbin/nologin
sssd:x:992:990:User for sssd:/sbin/nologin
oprofile:x:16:16:Special user account to be used by OProfile:/var/lib/oprofile:/sbin/nologin
systemd-bus-proxy:x:991:986:systemd Bus Proxy:/sbin/nologin
systemd-network:x:990:985:systemd Network Management:/sbin/nologin
geoclue:x:989:984:User for geoclue:/var/lib/geoclue:/sbin/nologin
setroubleshoot:x:988:983:/var/lib/setroubleshoot:/sbin/nologin
dovecot:x:97:97:Dovecot IMAP server:/usr/libexec/dovecot:/sbin/nologin
dovenull:x:987:980:Dovecot's unauthorized user:/usr/libexec/dovecot:/sbin/nologin
[student@CentOS7 ~]$
```

Đối với mỗi người dùng, bảy trường sau được duy trì trong tệp / etc / passwd:

Tên trường Chi tiết Nhận xét

Tên người dùng Tên đăng nhập của người dùng phải dài từ 1 đến 32 ký tự

Mật khẩu Mật khẩu người dùng (hoặc ký tự x nếu mật khẩu được lưu trong tệp / etc / bóng) ở định dạng được mã hóa Không bao giờ được hiển thị trong Linux khi nó được nhập; điều này dùng mắt tò mò

ID người dùng (UID) Mỗi người dùng phải có id người dùng (UID)

UID 0 được dành riêng cho người dùng root

Phạm vi từ 1-99 được dành riêng cho các tài khoản được xác định trước khác

Phạm vi từ 100-999 được dành riêng cho các tài khoản và nhóm hệ thống

Người dùng bình thường có UID từ 1000 trở lên

ID nhóm (GID) ID nhóm chính (GID); Số nhận dạng nhóm được lưu trữ trong tệp / etc / nhóm Được trình bày chi tiết trong chương về tiến trình

Thông tin người dùng Trường này là tùy chọn và cho phép chèn thêm thông tin về người dùng, chẳng hạn như tên của họ Ví dụ: Rufus T. Firefly

Trang chủ Thư mục Vị trí đường dẫn tuyệt đối của thư mục chính của người dùng Ví dụ: / home / rtfirefly

Shell Vị trí tuyệt đối của shell mặc định của người dùng Ví dụ: / bin / bash

## 9.2 Các kiểu tài khoản

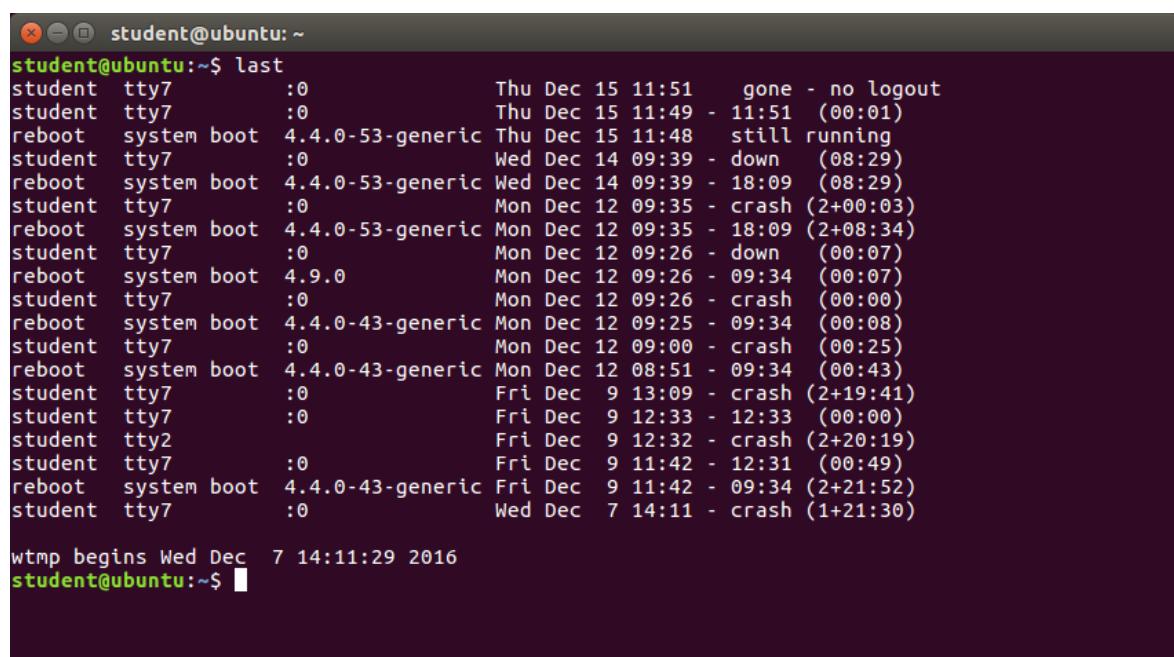
Theo mặc định, Linux phân biệt giữa một số loại tài khoản để cài đặt các tiến trình và khởi động công việc. Linux có bốn loại tài khoản:

**Root**

**System**

**Normal**

**Network**



```
student@ubuntu:~$ last
student  :0           Thu Dec 15 11:51   gone - no logout
student  :0           Thu Dec 15 11:49 - 11:51 (00:01)
reboot  system boot  4.4.0-53-generic Thu Dec 15 11:48   still running
student  :0           Wed Dec 14 09:39 - down (08:29)
reboot  system boot  4.4.0-53-generic Wed Dec 14 09:39 - 18:09 (08:29)
student  :0           Mon Dec 12 09:35 - crash (2+00:03)
reboot  system boot  4.4.0-53-generic Mon Dec 12 09:35 - 18:09 (2+08:34)
student  :0           Mon Dec 12 09:26 - down (00:07)
reboot  system boot  4.9.0          Mon Dec 12 09:26 - 09:34 (00:07)
student  :0           Mon Dec 12 09:26 - crash (00:00)
reboot  system boot  4.4.0-43-generic Mon Dec 12 09:25 - 09:34 (00:08)
student  :0           Mon Dec 12 09:00 - crash (00:25)
reboot  system boot  4.4.0-43-generic Mon Dec 12 08:51 - 09:34 (00:43)
student  :0           Fri Dec 9 13:09 - crash (2+19:41)
student  :0           Fri Dec 9 12:33 - 12:33 (00:00)
student  tty2          Fri Dec 9 12:32 - crash (2+20:19)
student  :0           Fri Dec 9 11:42 - 12:31 (00:49)
reboot  system boot  4.4.0-43-generic Fri Dec 9 11:42 - 09:34 (2+21:52)
student  :0           Wed Dec 7 14:11 - crash (1+21:30)

wtmp begins Wed Dec 7 14:11:29 2016
student@ubuntu:~$
```

Để có một môi trường làm việc an toàn, nên cấp các đặc quyền tối thiểu có thể và cần thiết cho các tài khoản và xóa các tài khoản không hoạt động. Tiện ích cuối cùng, cho thấy lần cuối mỗi người dùng đăng nhập vào hệ thống, có thể được sử dụng để giúp xác định các tài khoản không hoạt động có khả năng là ứng cử viên để xóa hệ thống.

Hãy nhớ rằng các thực tiễn bạn sử dụng trên các hệ thống kinh doanh đa người dùng nghiêm ngặt hơn các thực tiễn bạn có thể sử dụng trên các hệ thống máy tính để bàn cá nhân chỉ ảnh hưởng đến người dùng thông thường. Điều này đặc biệt đúng với bảo mật. Chúng tôi hy vọng cho bạn thấy các thực tiễn áp dụng cho các máy chủ doanh nghiệp mà bạn có thể sử dụng trên tất cả các hệ thống, nhưng hiểu rằng bạn có thể chọn thư giãn các quy tắc này trên hệ thống cá nhân của riêng bạn.

### 9.3 Hiểu biết về tài khoản root

root là tài khoản đặc quyền nhất trên hệ thống Linux / UNIX. Tài khoản này có khả năng thực hiện tất cả các khía cạnh của quản trị hệ thống, bao gồm thêm tài khoản, thay đổi mật khẩu người dùng, kiểm tra tệp nhật ký, cài đặt phần mềm, v.v. Phải hết sức cẩn thận khi sử dụng tài khoản này. Nó không có giới hạn bảo mật áp đặt lên nó.

Khi bạn đã đăng nhập, hoặc đóng vai trò là root, dấu nhắc shell sẽ hiển thị '#' (nếu bạn đang sử dụng bash và bạn đã tùy chỉnh lời nhắc khi chúng ta thảo luận ở nơi khác trong khóa học này). Công ước này nhằm phục vụ như một cảnh báo cho bạn về sức mạnh tuyệt đối của tài khoản này.

#### + Các tác vụ yêu cầu quyền tài khoản root

quyền root được yêu cầu để thực hiện các hoạt động như:

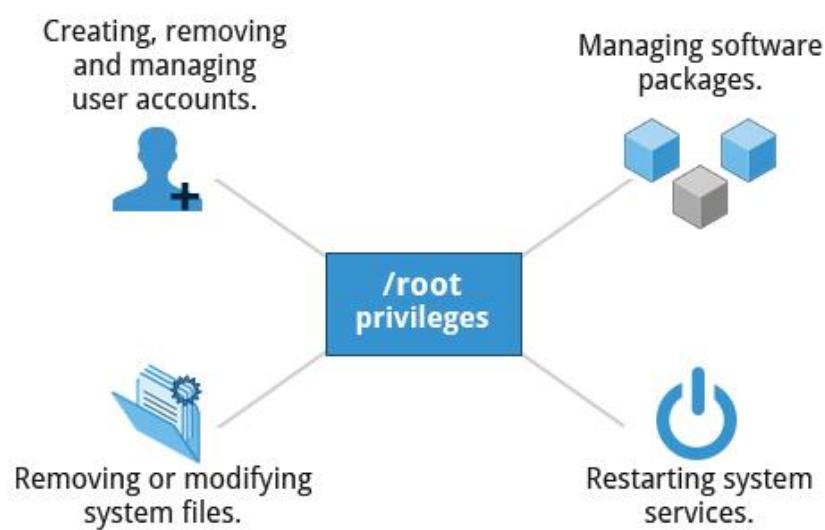
Tạo, xóa và quản lý tài khoản người dùng

Quản lý gói phần mềm

Loại bỏ hoặc sửa đổi các tập tin hệ thống

Khởi động lại dịch vụ hệ thống.

Người dùng tài khoản thông thường của các bản phân phối Linux có thể được phép cài đặt các gói phần mềm, cập nhật một số cài đặt, sử dụng một số thiết bị ngoại vi và áp dụng các loại thay đổi khác nhau cho hệ thống. Tuy nhiên, đặc quyền gốc là cần thiết để thực hiện các tác vụ quản trị như khởi động lại dịch vụ, cài đặt thủ công các gói và quản lý các phần của hệ thống tệp nằm ngoài thư mục người dùng thông thường.



#### + Các tác vụ không yêu cầu tài khoản root

Người dùng tài khoản thông thường có thể thực hiện một số thao tác cần có quyền đặc biệt; tuy nhiên, cấu hình hệ thống phải cho phép các khả năng đó được thực hiện.

SUID (Đặt ID người dùng của chủ sở hữu khi thực thi, tương tự như tính năng "chạy dưới dạng" của Windows) là một loại quyền đặc biệt của tệp được cấp cho một tệp. SUID cung cấp quyền tạm thời cho người dùng để chạy chương trình với quyền của chủ sở hữu tệp (có thể là root) thay vì quyền do người dùng nắm giữ.

Bảng cung cấp các ví dụ về các hoạt động không yêu cầu quyền root:

Các hoạt động không yêu cầu đặc quyền Root Ví dụ về hoạt động này

Chạy máy khách mạng Chia sẻ tệp qua mạng

Sử dụng các thiết bị như máy in In qua mạng

Hoạt động trên các tệp mà người dùng có quyền thích hợp để truy cập Truy cập các tệp mà bạn có quyền truy cập hoặc chia sẻ dữ liệu qua mạng

Chạy các ứng dụng gốc SUID Thực hiện các chương trình như passwd

#### + So sánh lệnh sudo và su

Trong Linux, bạn có thể sử dụng su hoặc sudo để tạm thời cấp quyền truy cập root cho người dùng bình thường; Những phương pháp này thực sự khá khác nhau. Dưới đây là sự khác biệt giữa hai lệnh: **su** và **sudo**

Khi nâng đặc quyền, bạn cần nhập mật khẩu gốc. Việc cung cấp mật khẩu gốc cho người dùng bình thường sẽ không bao giờ được thực hiện. Khi nâng cao đặc quyền, bạn cần nhập mật khẩu người dùng và không phải mật khẩu gốc.

Khi người dùng nâng lên tài khoản root bằng su, người dùng có thể làm bất cứ điều gì mà người dùng root có thể làm miễn là người dùng muốn mà không cần hỏi lại mật khẩu. Cung cấp nhiều tính năng hơn và được coi là an toàn hơn và cấu hình nhiều hơn. Chính xác những gì người dùng được phép làm có thể được kiểm soát và giới hạn chính xác. Theo mặc định, người dùng sẽ luôn phải cung cấp mật khẩu để thực hiện các thao tác tiếp theo với sudo hoặc có thể tránh làm như vậy trong khoảng thời gian có thể định cấu hình.

Lệnh có các tính năng đăng nhập hạn chế. Lệnh có các tính năng đăng nhập chi tiết.

#### 9.4 Các đặc trưng của lệnh sudo

sudo có khả năng theo dõi các nỗ lực không thành công trong việc giành quyền truy cập root. Sự cho phép của người dùng khi sử dụng sudo dựa trên thông tin cấu hình được lưu trữ trong tệp / etc / sudoers và trong thư mục /etc/sudoers.d.

Một thông báo như sau sẽ xuất hiện trong tệp nhật ký hệ thống (thường là /var/log/safe) khi cố gắng thực thi sudo bash mà không xác thực thành công người dùng:

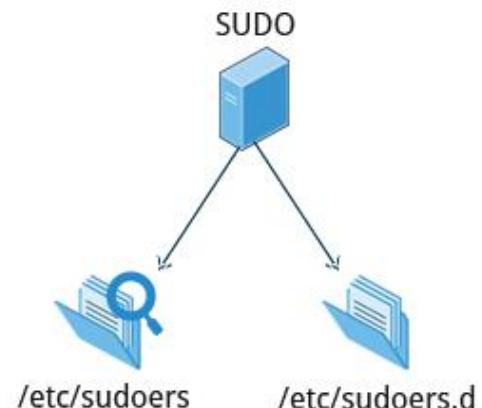
*authentication failure; logname=op uid=0 euid=0 tty=/dev/pts/6 ruser=op rhost=user=op*

*conversation failed*

*auth could not identify password for [op]*

*op : 1 incorrect password attempt ;*

*TTY=pts/6 ; PWD=/var/log ; USER=root ; COMMAND=/bin/bash*



## + Tập tin sudoerrs

Bất cứ khi nào sudo được gọi, một trình kích hoạt sẽ xem xét /etc/sudoers và các tệp trong /etc/sudoers.d để xác định xem người dùng có quyền sử dụng sudo hay không và phạm vi đặc quyền của họ là gì. Yêu cầu người dùng không xác định và yêu cầu thực hiện các thao tác không được phép cho người dùng ngay cả với sudo được báo cáo. Bạn có thể chỉnh sửa tệp sudoers bằng cách sử dụng visudo, điều này đảm bảo rằng chỉ có một người đang chỉnh sửa tệp tại một thời điểm, có quyền thích hợp và từ chối ghi ra tệp và thoát nếu có lỗi trong các thay đổi được thực hiện.

Cấu trúc cơ bản của một mục là:

**who where = (as\_whom) what**

Các tập tin có rất nhiều tài liệu trong đó về cách tùy chỉnh. Hầu hết các bản phân phối Linux hiện nay thích bạn thêm một tệp trong thư mục /etc/sudoers.d với tên giống như người dùng. Tệp này chứa cấu hình sudo của từng người dùng và người dùng không được chạm vào tệp cấu hình chính trừ các thay đổi ảnh hưởng đến tất cả người dùng.

```
student@openSUSE:~/Pictures
File Edit View Search Terminal Help
#includedir /etc/sudoers.d
openSUSE:/etc # cat /etc/sudoers
## sudoers file.
##
## This file MUST be edited with the 'visudo' command as root.
## Failure to use 'visudo' may result in syntax or file permission errors
## that prevent sudo from running.
##
## See the sudoers man page for the details on how to write a sudoers file.
##

##
## Host alias specification
##
## Groups of machines. These may include host names (optionally with wildcards),
## IP addresses, network numbers or netgroups.
# Host_Alias    WEBSERVERS = www1, www2, www3

##
## User alias specification
##
## Groups of users. These may consist of user names, uids, Unix groups,
## or netgroups.
# User_Alias    ADMINS = millert, dowdy, mikef

##
## Cmnd alias specification
##
## Groups of commands. Often used to group related commands together.
# Cmnd_Alias    PROCESSES = /usr/bin/nice, /bin/kill, /usr/bin/renice, \
#                           /usr/bin/pkill, /usr/bin/top

##
## Defaults specification
##
```

## 9.5 Ghi nhật ký lệnh

Theo mặc định, các lệnh sudo và mọi lỗi đều được đăng nhập /var/log/auth.log trong họ phân phối Debian và trong / var / log / message và / hoặc / var / log / safe trên các hệ thống khác. Đây là một biện pháp bảo vệ quan trọng để cho phép theo dõi và chịu trách nhiệm về việc sử dụng sudo. Một mục tiêu biểu của tin nhắn chúa:

Gọi tên người dùng

Thông tin đầu cuối

## Thư mục làm việc

Tài khoản người dùng được gọi

Lệnh với các đối số.

Chạy một lệnh như sudo whoami dẫn đến một mục nhập tệp nhật ký, chẳng hạn như:

Dec 8 14:20:47 server1 sudo: op : TTY=pts/6 PWD=/var/log USER=root  
COMMAND=/usr/bin/whoami

```
File Edit View Search Terminal Help
exit
c7:/home/coop>sudo tail /var/log/secure
Dec 15 15:31:55 c7 sudo: pam_unix(sudo:auth): authentication failure; logname=coop uid=1001 euid=0 tty=/dev/pts/2 ruser=linux1 rhost= user=linux1
Dec 15 15:32:07 c7 sudo:  linux1 : user NOT in sudoers ; TTY=pts/2 ; PWD=/usr/local/coop ; USER=root ; COMMAND=/usr/bin/ls -l /root
Dec 15 15:32:23 c7 su: pam_unix(su:auth): authentication failure; logname=coop uid=1001 euid=0 tty=pts/2 ruser=linux1 rhost= user=coop
Dec 15 15:33:00 c7 sudo: pam_unix(sudo:auth): authentication failure; logname=coop uid=1001 euid=0 tty=/dev/pts/2 ruser=linux1 rhost= user=linux1
Dec 15 15:33:07 c7 sudo:  linux1 : user NOT in sudoers ; TTY=pts/2 ; PWD=/usr/local/coop ; USER=root ; COMMAND=/usr/bin/bash
Dec 15 16:11:46 c7 sudo: pam_unix(sudo:auth): conversation failed
Dec 15 16:11:46 c7 sudo: pam_unix(sudo:auth): auth could not identify password for [linux1]
Dec 15 16:11:46 c7 sudo:  linux1 : user NOT in sudoers ; TTY=pts/2 ; PWD=/usr/local/coop ; USER=root ; COMMAND=/usr/bin/tail /var/log/secure
Dec 15 16:11:47 c7 su: pam_unix(su:session): session closed for user linux1
Dec 15 16:11:53 c7 sudo:  coop : TTY=pts/2 ; PWD=/usr/local/coop ; USER=root ; COMMAND=/usr/bin/tail /var/log/secure
c7:/home/coop>
```

## 9.6 Cách ly các tiến trình

Linux được coi là an toàn hơn so với nhiều hệ điều hành khác vì các tiến trình được cách ly tự nhiên với nhau. Một tiến trình thường không thể truy cập tài nguyên của tiến trình khác, ngay cả khi tiến trình đó đang chạy với cùng đặc quyền người dùng. Do đó, Linux gây khó khăn (mặc dù chắc chắn là không thể) đối với các vi rút và khai thác bảo mật để truy cập và tấn công các tài nguyên ngẫu nhiên trên một hệ thống.

Các cơ chế bảo mật bổ sung gần đây hạn chế rủi ro hơn nữa bao gồm:

Nhóm kiểm soát (nhóm)

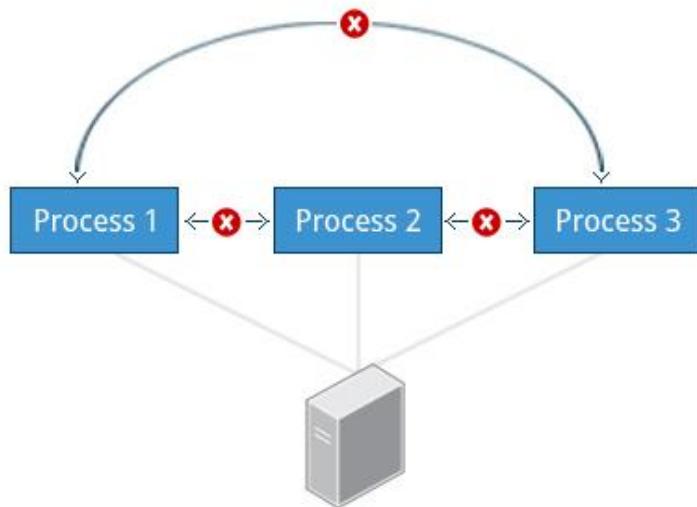
Cho phép quản trị viên hệ thống nhóm các tiến trình và liên kết các tài nguyên hữu hạn cho mỗi nhóm.

Hộp đựng

Làm cho nó có thể chạy nhiều hệ thống Linux (container) bị cô lập trên một hệ thống bằng cách dựa vào các nhóm.

## Ảo hóa

Phần cứng được mô phỏng theo cách không chỉ có thể tách rời các tiến trình mà toàn bộ hệ thống được chạy đồng thời dưới dạng khách cách ly và cách điện (máy ảo) trên một máy chủ vật lý.



## 9.7 Truy cập thiết bị phần cứng

Linux giới hạn người dùng truy cập vào các thiết bị phần cứng không nối mạng theo cách cực kỳ giống với truy cập tệp thông thường. Các ứng dụng tương tác bằng cách tham gia lớp hệ thống tập tin (độc lập với thiết bị hoặc phần cứng thực tế mà tập tin cư trú). Lớp này sau đó sẽ mở một tệp đặc biệt của thiết bị (thường được gọi là nút thiết bị) trong thư mục / dev tương ứng với thiết bị được truy cập. Mỗi tệp đặc biệt của thiết bị có các trường quyền sở hữu tiêu chuẩn, nhóm và thế giới. Bảo mật được thực thi một cách tự nhiên giống như khi các tệp tiêu chuẩn được truy cập.

Ví dụ, đĩa cứng được biểu diễn dưới dạng / dev / sd \*. Trong khi người dùng root có thể đọc và ghi vào đĩa theo cách thô, chẳng hạn, bằng cách thực hiện một cái gì đó như:

```
$ echo Hello world > / dev / sda1
```

Các quyền tiêu chuẩn như trong hình, khiến người dùng thông thường không thể làm như vậy. Viết cho một thiết bị theo kiểu này có thể dễ dàng xóa sạch hệ thống tệp được lưu trữ trên thiết bị theo cách không thể sửa chữa mà không cần nỗ lực lớn, nếu có. Việc đọc và ghi tệp bình thường trên đĩa cứng của các ứng dụng được thực hiện ở mức cao hơn thông qua hệ thống tệp và không bao giờ thông qua truy cập trực tiếp vào nút thiết bị.

```
c7:/etc>cd /dev
c7:/dev>ls -l /dev/sd*
brw-rw---- 1 root disk 8,  0 Dec 21 07:57 /dev/sda
brw-rw---- 1 root disk 8,  1 Dec 21 07:57 /dev/sda1
brw-rw---- 1 root disk 8,  2 Dec 21 07:57 /dev/sda2
brw-rw---- 1 root disk 8,  3 Dec 21 07:57 /dev/sda3
brw-rw---- 1 root disk 8,  5 Dec 21 07:57 /dev/sda5
brw-rw---- 1 root disk 8,  6 Dec 21 07:57 /dev/sda6
brw-rw---- 1 root disk 8, 16 Dec 21 07:57 /dev/sdb
brw-rw---- 1 root disk 8, 17 Dec 21 07:57 /dev/sdb1
brw-rw---- 1 root disk 8, 18 Dec 21 07:57 /dev/sdb2
brw-rw---- 1 root disk 8, 21 Dec 21 07:57 /dev/sdb5
brw-rw---- 1 root disk 8, 22 Dec 21 07:57 /dev/sdb6
brw-rw---- 1 root disk 8, 23 Dec 21 07:57 /dev/sdb7
c7:/dev>
```

## 9.8 Giữ cho Linux luôn cập nhật

Khi các vấn đề bảo mật trong nhân Linux hoặc các ứng dụng và thư viện được phát hiện, các bản phân phối Linux có một hồ sơ tốt về phản ứng nhanh và đẩy ra các bản sửa lỗi cho tất cả các hệ thống bằng cách cập nhật kho phần mềm của chúng và gửi thông báo để cập nhật ngay lập tức. Điều tương tự cũng đúng với các sửa lỗi và cải thiện hiệu suất không liên quan đến bảo mật.

Tuy nhiên, ai cũng biết rằng nhiều hệ thống không được cập nhật đủ thường xuyên và các vấn đề đã được chữa khỏi được phép tồn tại trên máy tính trong một thời gian dài; điều này đặc biệt đúng với các hệ điều hành độc quyền nơi người dùng không hiểu rõ hoặc không tin tưởng vào chính sách và của nhà cung cấp vì đôi khi các bản cập nhật có thể gây ra sự cố mới và phá vỡ các hoạt động hiện có. Nhiều vector tấn công thành công nhất đến từ việc khai thác các lỗ hổng bảo mật mà các bản sửa lỗi đã được biết đến nhưng chưa được triển khai trên toàn cầu.

Vì vậy, cách tốt nhất là tận dụng cơ chế phân phối Linux của bạn để cập nhật tự động và không bao giờ hoãn chúng. Rất hiếm khi một bản cập nhật như vậy sẽ gây ra vấn đề mới.



### Bài tập LAB

#### + Sử dụng lệnh sudo

1. Tạo người dùng mới, sử dụng useradd và cung cấp cho người dùng mật khẩu ban đầu với mật khẩu.
2. Cấu hình người dùng này để có thể sử dụng sudo.
3. Đăng nhập với tư cách hoặc chuyển sang người dùng mới này và đảm bảo bạn có thể thực thi một lệnh yêu cầu quyền root.



Timely System Update

Ví dụ: một lệnh tầm thường yêu cầu quyền root có thể là:

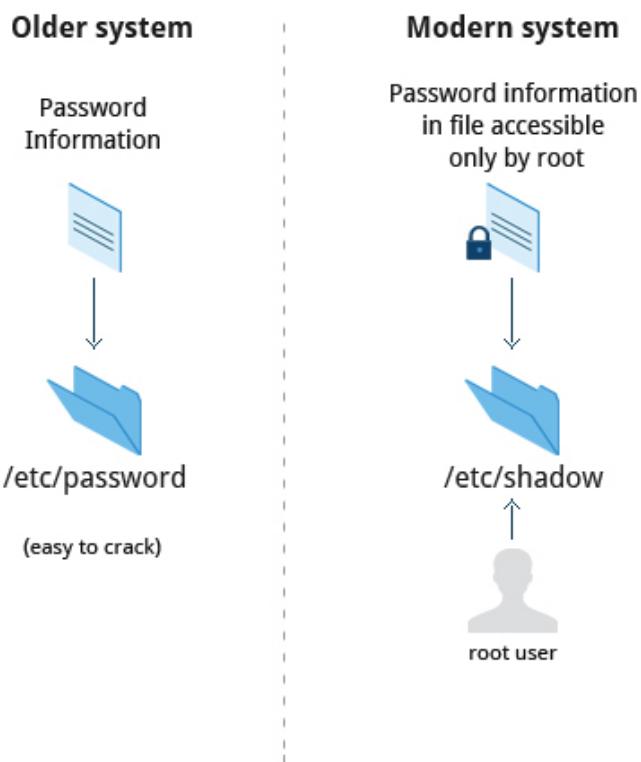
```
$ ls /root
```

### 9.9 Mật khẩu được lưu trữ như thế nào

Hệ thống xác minh tính xác thực và danh tính bằng thông tin đăng nhập của người dùng.

Ban đầu, mật khẩu được mã hóa được lưu trữ trong tệp /etc/passwd, mọi người đều có thể đọc được. Điều này làm cho mật khẩu bị bẻ khóa khá dễ dàng.

Trên các hệ thống hiện đại, mật khẩu thực sự được lưu trữ ở định dạng được mã hóa trong một tệp thứ cấp có tên /etc/bóng. Chỉ những người có quyền truy cập root mới có thể sửa đổi / đọc tệp này.



### 9.10 Thuật toán mật khẩu

Bảo vệ mật khẩu đã trở thành một yếu tố quan trọng của bảo mật. Hầu hết các bản phân phối Linux dựa trên thuật toán mã hóa mật khẩu hiện đại có tên SHA-512 (Thuật toán băm an toàn 512 bit), được phát triển bởi Cơ quan an ninh quốc gia Hoa Kỳ (NSA) để mã hóa mật khẩu.

Thuật toán SHA-512 được sử dụng rộng rãi cho các ứng dụng và giao thức bảo mật. Các ứng dụng và giao thức bảo mật này bao gồm TLS, SSL, PHP, SSH, S / MIME và IPSec. SHA-512 là một trong những thuật toán băm được thử nghiệm nhiều nhất.

Ví dụ: nếu bạn muốn thử nghiệm mã hóa SHA-512, từ "thử nghiệm" có thể được mã hóa bằng chương trình sha512sum để tạo ra mẫu SHA-512 (xem hình):

```
File Edit View Search Terminal Help
c7:/tmp>echo -n test | sha512sum
ee26b0dd4af7e749aa1a8ee3c10ae9923f618980772e473f8819a5d4940e0db27ac
185f8a0e1d5f84f88bc887fd67b143732c304cc5fa9ad8e6f57f50028a8ff -
c7:/tmp>
```

#### + Thực hành mật khẩu tốt

Các chuyên gia CNTT tuân theo một số thực tiễn tốt để bảo mật dữ liệu và mật khẩu của mọi người dùng.

Lão hóa mật khẩu là một phương pháp để đảm bảo rằng người dùng nhận được lời nhắc nhở họ tạo mật khẩu mới sau một khoảng thời gian cụ thể. Điều này có thể đảm bảo rằng mật khẩu, nếu bị bẻ khóa, sẽ chỉ có thể sử dụng được trong một khoảng thời gian giới hạn. Tính năng này được triển khai bằng chage, cấu hình thông tin hết hạn mật khẩu cho người dùng.

Một phương pháp khác là buộc người dùng thiết lập mật khẩu mạnh bằng Mô-đun xác thực có thể cảm (PAM). PAM có thể được cấu hình để tự động xác minh rằng mật khẩu được tạo hoặc sửa đổi bằng tiện ích passwd đủ mạnh. Cấu hình PAM được triển khai bằng thư viện có tên pam\_cracklib.so, cũng có thể được thay thế bằng pam\_passwdqc.so để có thêm tùy chọn.

```
student@openSUSE:~>
File Edit View Search Terminal Help
student@openSUSE:~> chage --list student
Password:
Last password change : Dec 14, 2016
Password expires : never
Password inactive : never
Account expires : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
student@openSUSE:~>
```

Người ta cũng có thể cài đặt các chương trình bẻ khóa mật khẩu, chẳng hạn như John The Ripper, để bảo mật tệp mật khẩu và phát hiện các mục nhập mật khẩu yếu. Bạn nên có

được sự cho phép bằng văn bản trước khi cài đặt các công cụ như vậy trên bất kỳ hệ thống nào mà bạn không sở hữu.

## Bài tập LAB: Xử lý vấn đề lão hóa mật khẩu

Với người dùng mới được tạo từ bài tập trước, hãy xem lão hóa mật khẩu cho người dùng.

Sửa đổi ngày hết hạn cho người dùng, đặt nó thành một cái gì đó đã qua và kiểm tra xem những gì đã thay đổi.

Khi bạn kết thúc và muốn xóa tài khoản vừa tạo, hãy sử dụng userdel, như trong:

```
$ sudo userdel người dùng mới
```

### 9.11 Yêu cầu đặt mật khẩu cho bộ tải khởi động

Bạn có thể bảo mật tiến trình khởi động bằng mật khẩu an toàn để ngăn người khác bỏ qua bước xác thực người dùng. Đối với các hệ thống sử dụng bộ tải khởi động GRUB cũ hơn, phiên bản 1, bạn có thể gọi grub-md5-crypt sẽ nhắc bạn nhập mật khẩu và sau đó mã hóa.

Sau đó, bạn phải chỉnh sửa /boot/grub/grub.conf bằng cách thêm dòng sau bên dưới mục thời gian chờ:

```
passwd --md5 $1 $74r8m1 $ NmkE69eAjXre.oF1k0cyk /
```

Bạn cũng có thể buộc mật khẩu chỉ cho một số lựa chọn khởi động chứ không phải tất cả.

Tuy nhiên, đối với phiên bản GRUB 2 (đã tiếp quản gần như hoàn toàn bây giờ) thì mọi thứ phức tạp hơn. Mặc dù bạn có sự linh hoạt hơn, bạn có thể tận dụng các tính năng nâng cao hơn, chẳng hạn như mật khẩu dành riêng cho người dùng (có thể là mật khẩu đăng nhập bình thường của họ). Ngoài ra, bạn không bao giờ chỉnh sửa tệp cấu hình, /boot/grub/grub.cfg, thay vào đó, bạn chỉnh sửa các tệp cấu hình hệ thống trong /etc/grub.d và sau đó chạy update-grub hoặc tiện ích tương đương trên bản phân phối Linux của bạn.

### 9.12 Các lỗ hổng trong Linux

#### Lỗ hổng phần cứng

Khi phần cứng có thể truy cập vật lý, bảo mật có thể bị xâm phạm bởi:

Ghi nhật ký chính

Ghi lại hoạt động thời gian thực của người dùng máy tính bao gồm các phím họ nhấn. Dữ liệu thu được có thể được lưu trữ cục bộ hoặc truyền đến các máy từ xa.

Mạng đánh hơi

Chụp và xem dữ liệu cấp gói mạng trên mạng của bạn.

Khởi động với đĩa trực tiếp hoặc cứu hộ

Kẻ lại và sửa đổi nội dung đĩa.

Chính sách bảo mật CNTT của bạn nên bắt đầu với các yêu cầu về cách bảo mật truy cập vật lý đúng cách vào máy chủ và máy trạm. Truy cập vật lý vào một hệ thống giúp kẻ tấn công có thể dễ dàng tận dụng một số vectơ tấn công, theo cách làm cho tất cả các khuyến nghị ở cấp hệ điều hành không liên quan.

Các hướng dẫn của bảo mật là:

Khóa máy trạm và máy chủ.

Bảo vệ các liên kết mạng của bạn sao cho những người bạn không tin tưởng không thể truy cập được.

Bảo vệ bàn phím của bạn nơi nhập mật khẩu để đảm bảo bàn phím không bị giả mạo.

Đảm bảo mật khẩu bảo vệ BIOS theo cách mà hệ thống không thể được khởi động bằng khóa DVD hoặc USB trực tiếp hoặc cứu hộ.

Đối với máy tính người dùng đơn và những người trong môi trường gia đình, một số tính năng trên (như ngăn chặn khả năng khởi động từ phương tiện lưu động) có thể quá mức và bạn có thể tránh thực hiện chúng. Tuy nhiên, nếu thông tin nhạy cảm có trên hệ thống của bạn cần được bảo vệ cẩn thận, thì nó không nên ở đó hoặc cần được bảo vệ tốt hơn bằng cách làm theo các hướng dẫn trên.



## Lỗ hổng phần mềm

Giống như tất cả các phần mềm, tin tặc thỉnh thoảng tìm thấy điểm yếu trong hệ sinh thái Linux. Sức mạnh của Linux (và cộng đồng nguồn mở nói chung) là tốc độ mà các lỗ hổng như vậy được phơi bày và khắc phục. Mặc dù thảo luận cụ thể về các lỗ hổng nằm ngoài phạm vi của khóa học này, chúng tôi đã chọn một vài chủ đề có liên quan cho Ban thảo luận (theo chủ đề "An ninh địa phương") để khuyến khích thảo luận thêm.

### **9.13 Câu hỏi ôn tập Chương 9**

1. Nắm bắt tốt các thực tiễn và công cụ tốt nhất để làm cho các hệ thống Linux an toàn nhất có thể.
2. Hiểu được sức mạnh và sự nguy hiểm của việc sử dụng tài khoản root (superuser).
3. Sử dụng lệnh **sudo** để thực hiện các hoạt động đặc quyền trong khi hạn chế các quyền hạn được tăng cường càng nhiều càng khả thi.
4. Giải thích tầm quan trọng của quá trình cách ly và truy cập phần cứng.
5. Làm việc với mật khẩu, bao gồm cách đặt và thay đổi chúng.
6. Mô tả cách bảo mật tiến trình khởi động và tài nguyên phần cứng