



# Samsung Innovation Campus

| Vạn vật kết nối – IoT

Together for Tomorrow!  
**Enabling People**

Education for Future Generations

Chương 5.

# Linux 101

**Vạn vật kết nối – IoT**

# Giới thiệu chương

---

## ◆ Mục tiêu chung

- ✓ Học về hệ điều hành Linux được sử dụng rộng rãi trong các hệ thống, thiết bị IoT.
- ✓ Đồng thời, học về các tính năng của Raspbian, một hệ điều hành dành riêng cho raspberry pi.
- ✓ Tìm hiểu công cụ dành cho nhà phát triển, Công cụ chỉnh sửa có thể được sử dụng trên Raspberry Pi OS.

## ◆ Nội dung của chương

- ✓ Bài 1. Tìm hiểu về Linux
- ✓ Bài 2. Giới thiệu về một số tập lệnh sử dụng trong hệ điều hành Raspberry Pi
- ✓ Bài 3. Lập trình Shell
- ✓ Bài 4. Công cụ chỉnh sửa và IDE (môi trường phát triển phần mềm)

Bài 1.

# Tìm hiểu về Linux

- | 1.1. Tổng quan về Linux
- | 1.2. Tính năng của Linux
- | 1.3. Hệ điều hành Raspberry Pi

### HĐH (Hệ điều hành) là gì?

- | là chương trình quản lý quan trọng nhất, và là một nguyên tắc giao tiếp cơ bản của phần cứng máy tính.
- | Mục tiêu của HĐH là quản lý phần cứng hiệu quả nhất.
- | HĐH hỗ trợ người dùng làm việc và giúp các chương trình khác hoạt động.

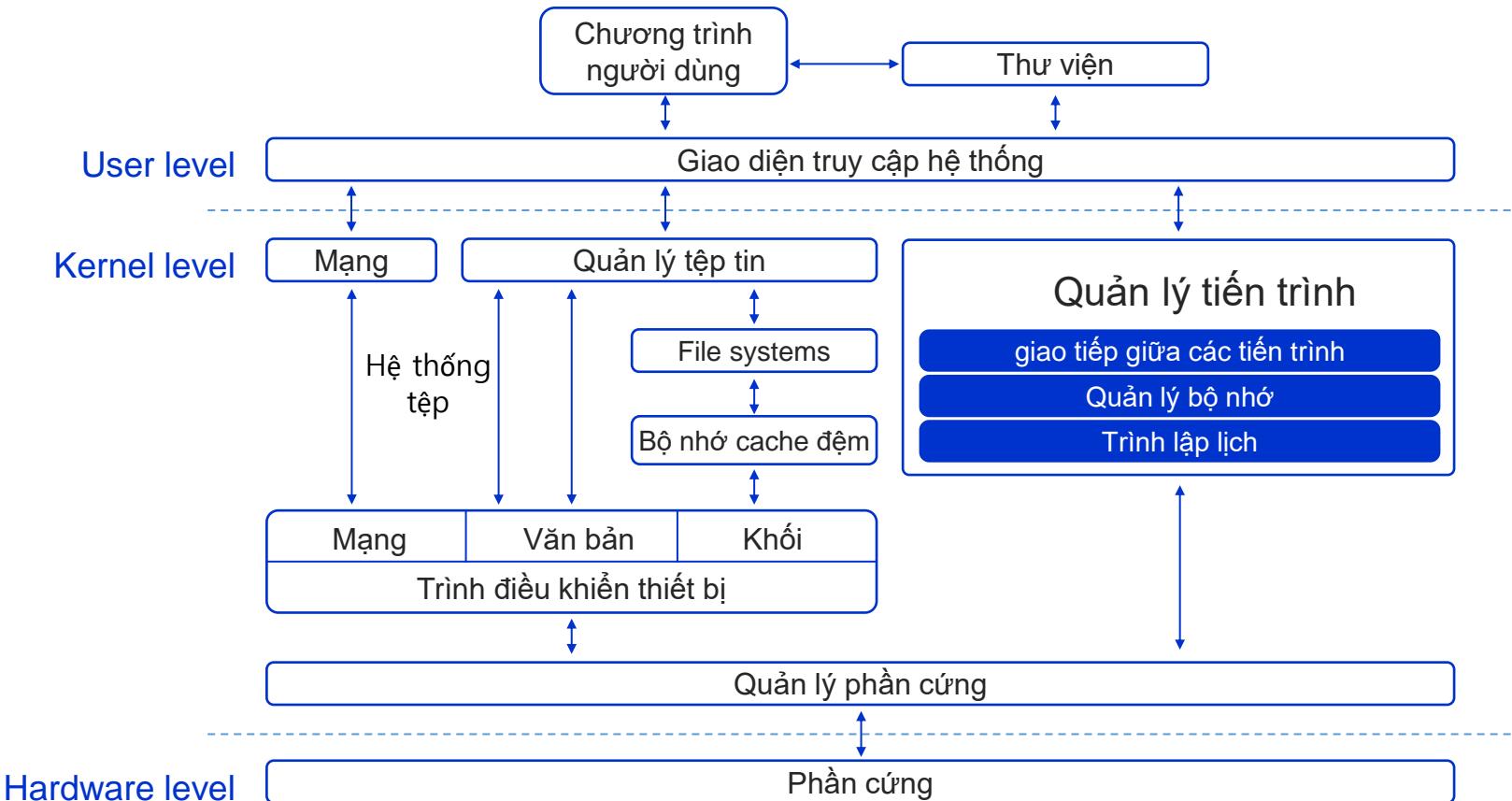
Người dùng phổ thông không cần hiểu rõ về hệ điều hành. Tuy nhiên, lập trình viên hệ thống phải hiểu rõ về hệ điều hành để viết chương trình hiệu quả.

### HĐH (Hệ điều hành) là gì?

- Về lý thuyết, HĐH là một phần mềm phức tạp, được chia thành nhiều loại với các sản phẩm thương mại hay sản phẩm thử nghiệm khác nhau.



## Cấu trúc của Hệ điều hành



Cấu trúc hệ điều hành

# Lịch sử của Linux (UNIX)

## | Các giới hạn của HĐH UNIX, lịch sử phát triển của LINUX

- HĐH UNIX, tiền thân của LINUX, được phát triển vào năm 1969 bởi Kenneth Lane Thompson tại Bell Labs ở Mỹ, người đã tạo ra một hệ điều hành có thể thực thi hiệu quả đa chức năng người dùng và đa nhiệm.
- UNIX là một tập hợp ngôn ngữ được thiết kế cho mô hình PDP-7. Điểm bất lợi là hệ điều hành phải được viết lại theo tập hợp ngữ (ngôn ngữ lập trình bậc thấp) của từng kiểu máy tính khác nhau.
- Năm 1971, Bell Labs đã giải quyết vấn đề bằng cách thực hiện biên dịch chéo để xử lý mã nguồn và tạo mã có thể chạy trên các nền tảng phần cứng máy tính khác nhau.



# Lịch sử của Linux (UNIX)

## Những cải thiện của UNIX với ngôn ngữ lập trình C

- Với công việc của Bell Labs, cùng một hệ điều hành có thể được sử dụng trong các nền tảng khác nhau chỉ bằng cách biên dịch lại mã với từng máy tính.
- UNIX được phát triển với ngôn ngữ lập trình C, không chỉ hỗ trợ việc điều khiển phần cứng trực tiếp như hợp ngữ mà còn có thể tạo mã mà không phụ thuộc vào loại phần cứng. (Được phát triển bởi Dennis Ritchie)
- Do đó, mã nguồn được viết bằng ngôn ngữ C có thể được sử dụng trong các môi trường phần cứng khác nhau thông qua quá trình biên dịch.



Dennis Ritchie



C Language Logo

# Lịch sử của Linux

## | Linus Torvalds, người tạo ra Linux

- MINIX, ban đầu được mô phỏng theo UNIX, là một hệ điều hành phù hợp với đa nhiệm cho PC. Nó được phát miễn phí cùng với mã nguồn cho sinh viên.
- Linus Torvalds từ Đại học Helsinki đã cải thiện hệ thống bằng cách xây dựng lớp nhân (kernel) UNIX chạy trong PC.
- Kernel là thành phần cốt lõi của hệ điều hành. Nó là giao diện chính hoạt động giữa phần cứng và phần mềm. Đây là thành phần quan trọng nhất của HĐH, hỗ trợ các chức năng cơ bản của hệ điều hành như xử lý bắt giữ, quản lý tiến trình, quản lý bộ nhớ, quản lý file hệ thống và cung cấp các giao diện lập trình. Việc tăng cường, mở rộng các chức năng của hệ điều hành thường diễn ra trong các lớp nhân.



Linus Torvalds



Tux (Linux Mascot)

## Lịch sử của Linux

### | Các đặc điểm của Mã nguồn mở đã tăng cường đáng kể sự phát triển của Linux

Linux có mã nguồn mở cho phép nhiều nhà phát triển tham gia phát triển Linux. Điều này có thể đã làm cho Linux phát triển bùng nổ.

- Khi Linux Phiên bản 0.01 được Linus Torvalds phát triển lần đầu tiên vào năm 1991, nó chỉ bao gồm một trình điều khiển đĩa cứng và hệ thống tệp (file) và các chức năng này có hiệu quả kém.
- Khi nó được giới thiệu trên Internet, nó đã thu hút được sự quan tâm của nhiều nhà phát triển. Kể từ khi Richard Stallman, chủ tịch của Tổ chức Phần mềm Tự do phân phối Linux miễn phí cho công chúng, rất nhiều nhà phát triển đã đóng góp vào việc cải tiến Linux.
- Ông đã cống hiến hết mình để thúc đẩy quyền sử dụng các chương trình máy tính cho người dùng máy tính. Khi mới vận động cho việc phân phối miễn phí phần mềm, ông đã bị coi thường và đồ lỗi nhưng giờ đây Mã nguồn mở đang trở thành động lực phát triển công nghệ.
- Đây là lý do tại sao những người dùng Linux trên toàn thế giới gọi Linus Torvalds là cha đẻ của Linux và Richard Stallman là vị thánh của Linux.

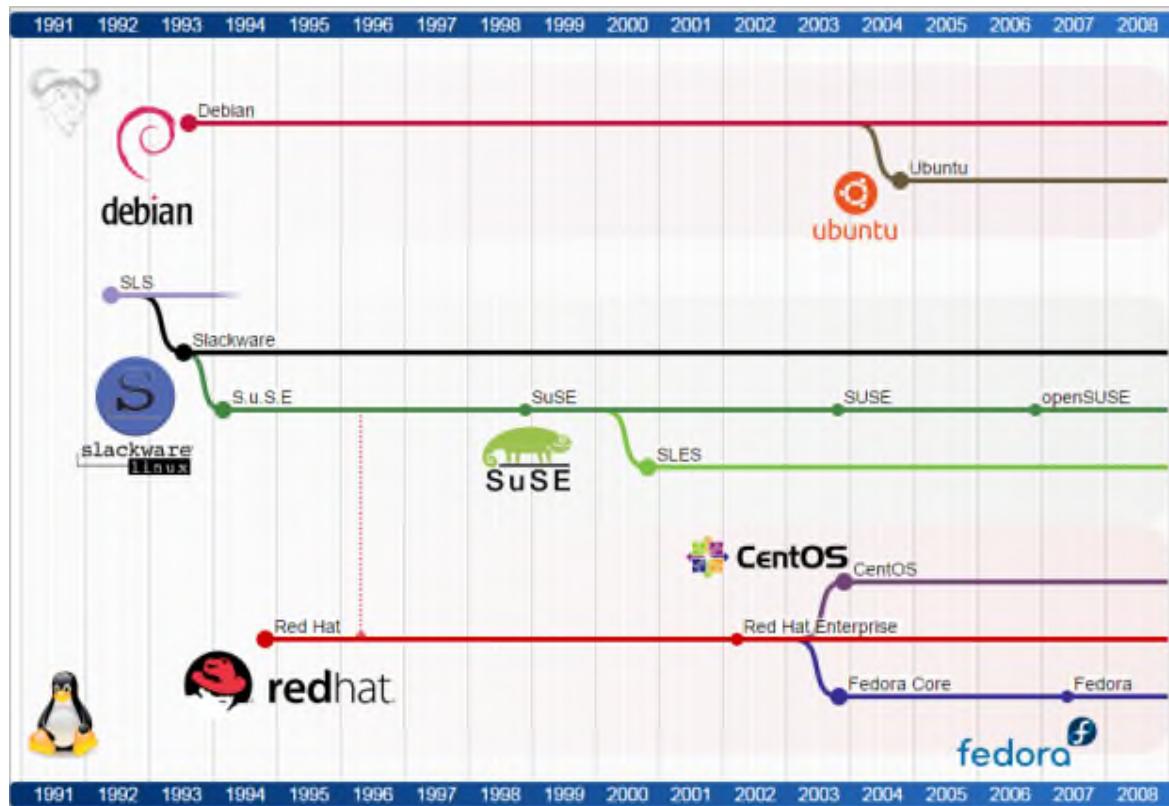


Richard Stallman

## Linux (Linus + UNIX)

- █ Cũng như UNIX, nền tảng của Linux được thiết kế từ đầu để trở thành một mạng truyền thông, được tối ưu hóa để chạy như một máy chủ.
- █ Môi trường làm việc của Linux hỗ trợ đa nhiệm và đa người dùng dựa trên các chức năng của hệ thống máy, với hệ thống X Windows, như một môi trường làm việc GUI (Giao diện người dùng đồ họa). Do đó, có thể đồng thời thiết lập môi trường máy chủ bằng cách cài đặt Apache HTTP Server, v.v.
- █ Máy chủ Linux sau này được tách khỏi hệ thống X Windows, chỉ bao gồm các chương trình cần thiết để tối ưu hóa chức năng của hệ thống máy chủ.
- █ Theo nhà sáng lập Linux, 90% phiên bản HĐH sử dụng cho hệ thống điện toán đám mây, 82% cho điện thoại thông minh trên toàn thế giới, 62% sử dụng cho thiết bị nhúng và 99% siêu máy tính đều chạy trên Linux.
- █ Các hệ điều hành mang tính đột phá như Fedora và raspberry pi đã xuất hiện và hệ điều hành di động 'Android' do Google phân phối cũng là một hệ điều hành mã nguồn mở dựa trên Linux.

### Lược đồ các biến thể chính của Linux



- Hệ điều hành Raspberry Pi là một hệ điều hành phát triển dựa trên HĐH Debian.

Bài 1.

# Tìm hiểu về Linux

| 1.1. Tổng quan về Linux

| 1.2. Tính năng của Linux

| 1.3. HĐH Raspberry Pi

# Tính năng của Hệ điều hành Linux

## Phát hiện và thiết lập trình điều khiển phần cứng

- When starting the Linux operating system (when the computer is turned on), the OS will check the hardware and load the necessary software drivers to control the hard drive (commonly known as a driver).

## Quản lý tiến trình

- The Linux operating system must track multiple processes running simultaneously to determine which process has access to the CPU, and manage the state of each process.

## Quản lý bộ nhớ

- Managing RAM and swap space requires memory allocation for programs. The Linux operating system defines how memory is managed based on user requirements.

## Tính năng của Hệ điều hành Linux

### I Cung cấp giao diện người dùng

- Hệ điều hành sẽ cung cấp các cơ chế truy cập vào hệ thống. Ở các phiên bản đầu, Linux chỉ hỗ trợ truy cập thông qua một công cụ biên dịch theo dòng lệnh, có tên là Shell. Hiện nay, Linux đã hỗ trợ giao diện dạng đồ họa.

### I Quản lý hệ thống tệp tin

- Hệ điều hành Linux quản lý quyền sở hữu và quyền truy cập vào các tệp tin (file) và thư mục trong hệ thống tệp tin.

### I Cung cấp phương pháp xác thực và quản lý truy cập của người dùng

- Tạo tài khoản người dùng và thiết lập ranh giới giữa các người dùng là những tính năng cơ bản nhất của Linux. Tách tài khoản giữa người dùng và nhóm là nền tảng để người dùng có thể quản lý các tệp tin và tiến trình của mình.

## Tính năng của Hệ điều hành Linux

### I Cung cấp tiện ích để quản lý

- Linux cung cấp nhiều lệnh và màn hình giao tiếp đồ họa để thực hiện các tác vụ như thêm người dùng hoặc quản lý ổ đĩa, giám sát mạng, cài đặt phần mềm, duy trì và quản lý bảo mật máy tính.

### I Khởi tạo và chạy dịch vụ

- Các tiến trình chạy ở xử lý nền, được hiểu như một dịch vụ, cung cấp khả năng truy nhập đến các thành phần khác nhau của hệ thống, các kết nối mạng, mỗi tiến trình được HĐH quản lý và thực thi theo lượt. Nhiều loại dịch vụ khác nhau chạy trên Linux sẽ có cách khởi động và dừng khác nhau. Linux gồm một web blinder để hiển thị các trang web, và bản thân Linux là một máy chủ để chuyển tiếp các trang web đến các hệ thống đang kết nối khác.

### I Cung cấp các công cụ lập trình

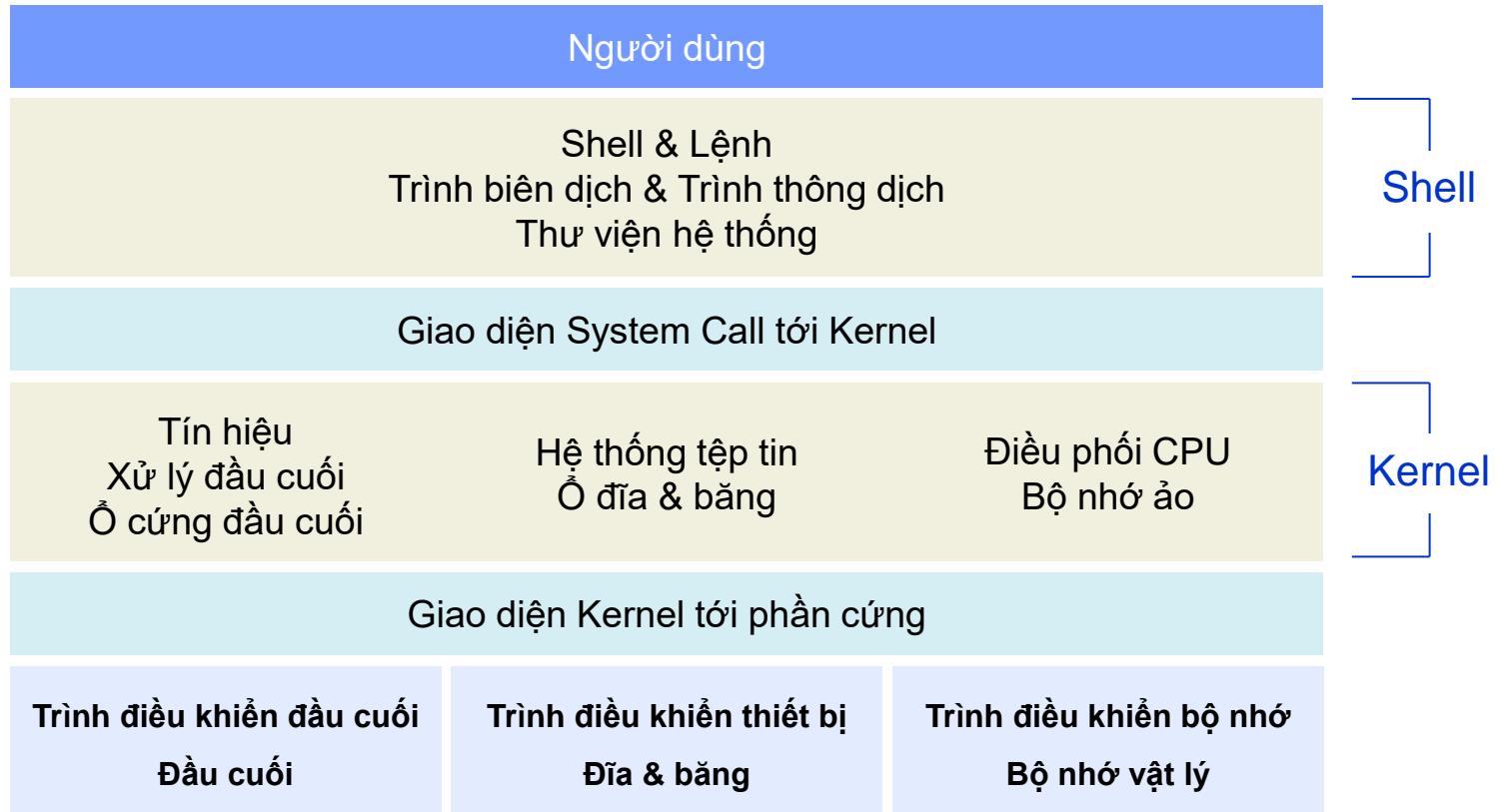
- Linux cung cấp một loạt các tiện ích lập trình để tạo ứng dụng và thư viện nhằm triển khai các giao diện nâng cao.

# Định nghĩa Mã nguồn mở của Tổ chức Sáng kiến Nguồn mở (OSI)

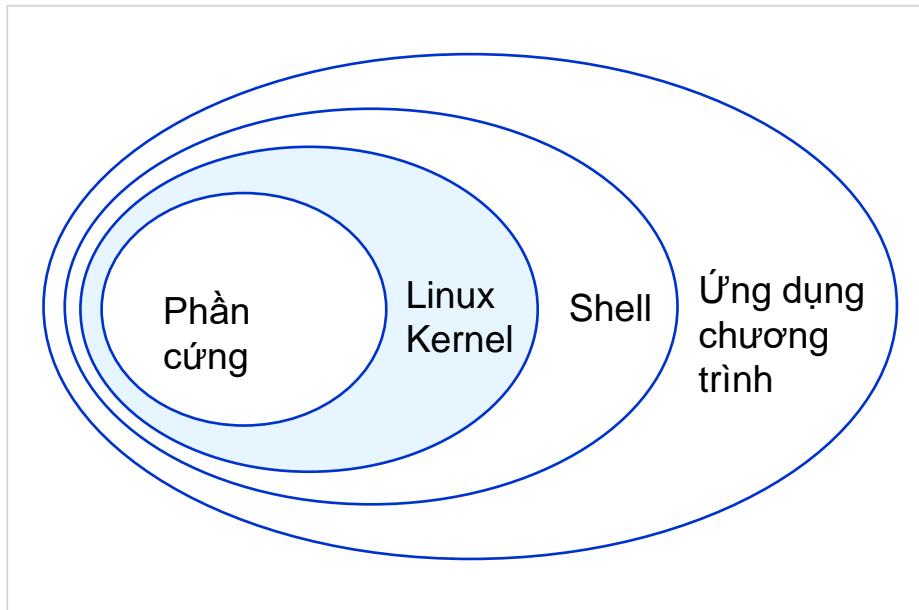
Mục đích chính của mã nguồn mở là cùng sử dụng mã nguồn, nhưng tổ chức OSI cũng xác định các mục tiêu khác trong định nghĩa về mã nguồn mở. Các quy tắc được liệt kê dưới đây nhằm bảo vệ sự tự do và toàn vẹn của mã nguồn mở và tuân thủ các giấy phép mã nguồn mở.

- 1) Tái phân phối miễn phí
- 2) Mã nguồn
- 3) Công việc có nguồn gốc rõ ràng
- 4) Mã nguồn của tác giả có tính toàn vẹn
- 5) Không phân biệt các cá nhân hoặc nhóm người
- 6) Không phân biệt các nỗ lực
- 7) Cấp phát giấy phép
- 8) Giấy phép không được quy định cho một sản phẩm
- 9) Giấy phép không hạn chế phần mềm khác
- 10) Giấy phép phải trung lập về công nghệ

## Cấu trúc hệ thống Linux



## Linux System Architecture



### Kernel: Lõi của HĐH Linux

- Quản lý Tiến trình/Bộ nhớ/Hệ thống tệp tin/Thiết bị
- Khả năng khởi tạo và quản lý toàn bộ tài nguyên trên máy tính

### Shell: Giao diện người dùng

- Thông dịch lệnh, các hàm hỗ trợ lập trình.

### Chương trình ứng dụng

- Các công cụ phát triển chương trình.
- Công cụ soạn thảo văn bản, công cụ liên quan đến mạng, v.v.

Bài 1.

# Understanding Linux

- | 1.1. Tổng quan về Linux
- | 1.2. Tính năng của Linux
- | **1.3. HĐH Raspberry Pi**

## HĐH Raspberry Pi là gì?

### I HĐH Raspberry Pi là gì?

- ▶ Raspbian là một hệ điều hành do Raspberry Pi Foundation phát triển dành riêng cho Raspberry Pi.
- ▶ Được Raspberry Pi Foundation chính thức phát triển làm hệ điều hành chính cho dòng máy tính bảng đơn (SBC) Raspberry Pi kể từ năm 2015.
- ▶ Hệ điều hành Raspberry Pi là một dự án độc lập được Mike Thompson và Peter Green phát triển.
- ▶ Phiên bản đầu tiên được hoàn thiện vào tháng 6 năm 2012 và vẫn tiếp tục được phát triển.
- ▶ Bên cạnh Raspbian, Pidora, OpenELEC, RaspbianMC, và Risc là những hệ điều hành phân phối cho Raspberry Pi.



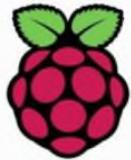
**Raspberry Pi OS**



**Mike Thompson**

## HĐH Raspberry Pi là gì?

### I HĐH Raspberry Pi là gì?



**Raspberry Pi OS**

- ▶ Hệ điều hành Raspberry Pi được tối ưu hóa cao cho dòng CPU ARM hiệu suất thấp Raspberry Pi.
- ▶ Nó là một bản phân phối được tạo dựa trên Debian Linux và HĐH này là tùy chọn cơ bản của NOOBS, phần mềm cài đặt HĐH của Raspberry Pi và nó bao gồm tệp img.
- ▶ Hệ điều hành Raspberry Pi đang được nhiều người dùng Raspberry Pi sử dụng và đây là một trong những hệ điều hành tiện lợi có thể sử dụng cho những người lần đầu sử dụng Raspberry Pi.

Bài 2.

# Hướng dẫn lệnh hệ điều hành Raspberry Pi

## | 2.1. Kết nối với hệ điều hành Raspberry Pi từ máy tính cá nhân

- | 2.2. Lệnh cơ bản
- | 2.3. Các lệnh về Hệ thống tệp
- | 2.4. Các lệnh về điều khiển hệ thống
- | 2.5. Các lệnh về Network
- | 2.6. Các lệnh về Người dùng & Quyền
- | 2.7. Công cụ lập trình

### Thiết lập máy ảo Raspberry Pi trong VirtualBox

Tải xuống hệ điều hành Raspberry Pi chạy trên hệ điều hành máy tính

- ▶ <https://www.raspberrypi.com/software/raspberry-pi-desktop/>

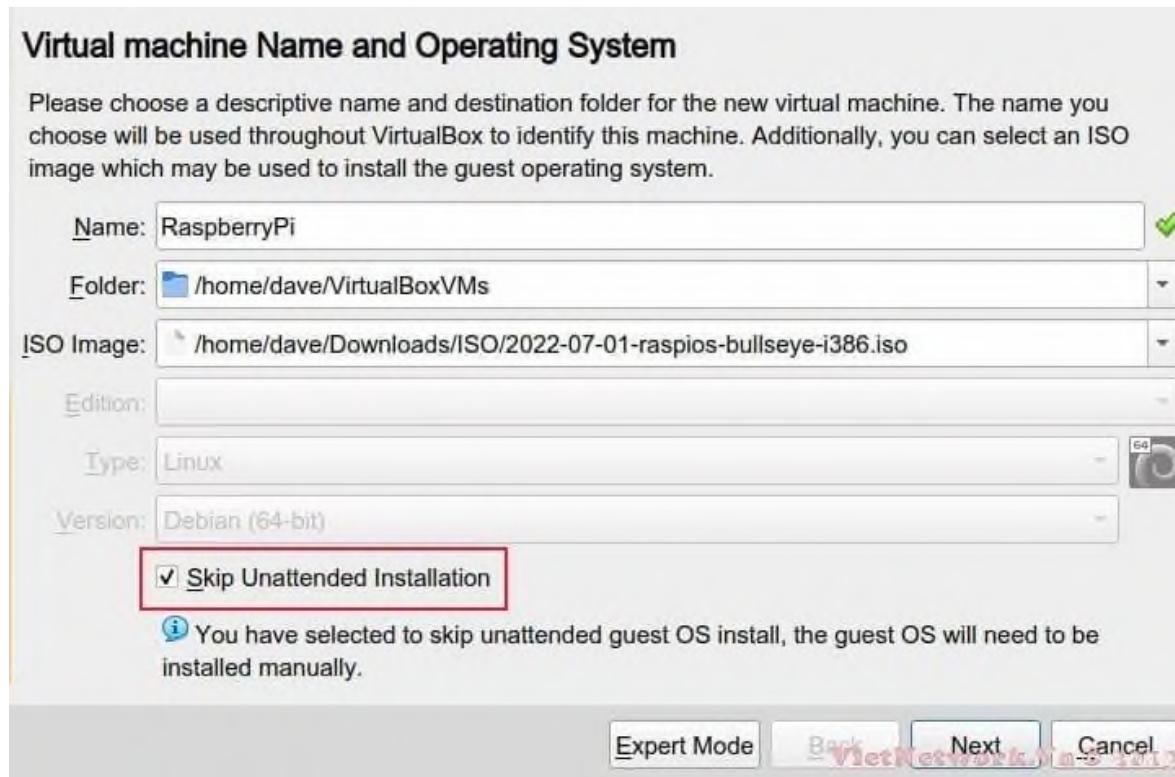
Cài đặt chương trình tạo máy ảo VirtualBox

- ▶ <https://www.virtualbox.org/wiki/Downloads>



## Thiết lập máy ảo Raspberry Pi trong VirtualBox

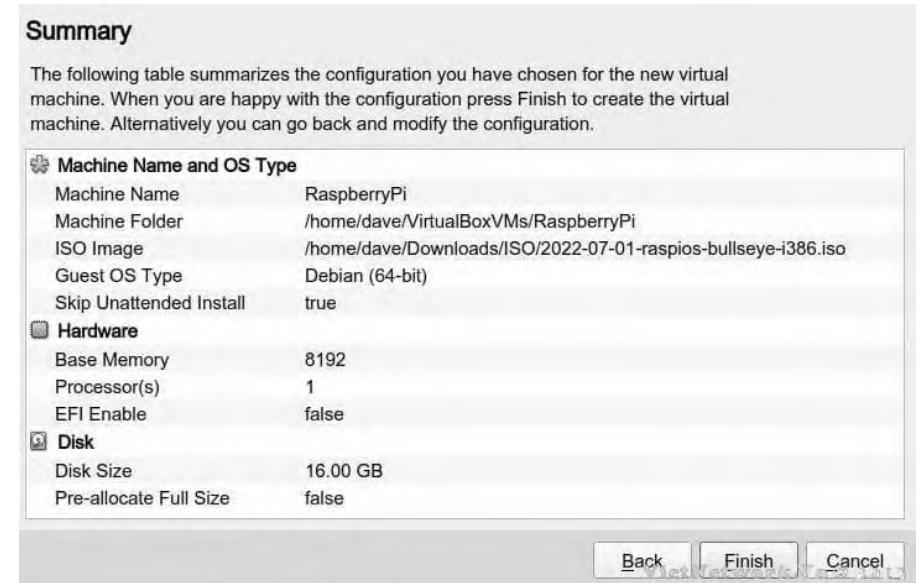
Thiết lập máy ảo để cài đặt hệ điều hành Raspbian OS



## Thiết lập máy ảo Raspberry Pi trong VirtualBox

| Cấu hình máy ảo để cài đặt hệ điều hành Raspbian OS

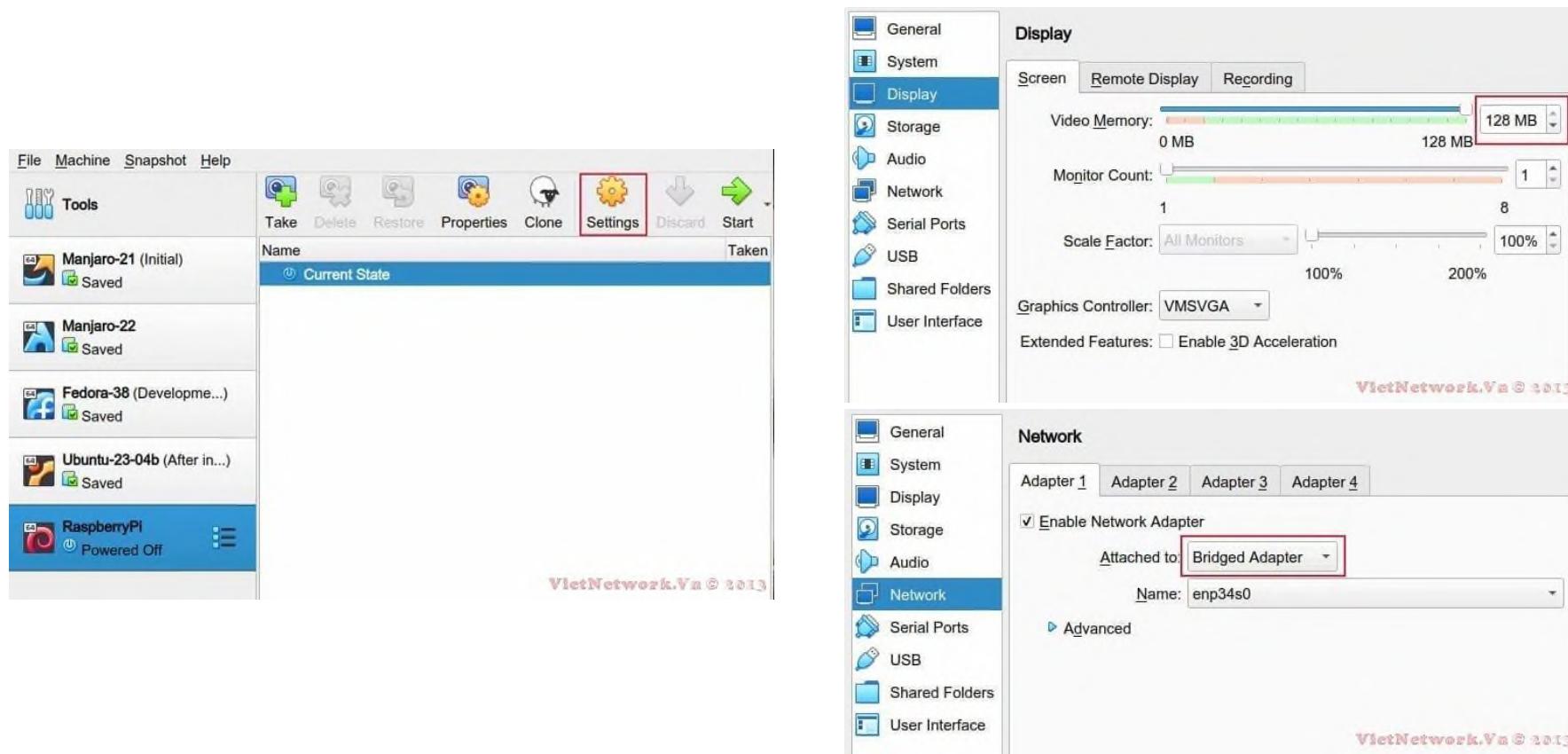
*Tham khảo link: <https://vietnetwork.vn/virtualization/cach-chay-raspberry-pi-trong-virtualbox/>*



## Thiết lập máy ảo Raspberry Pi trong VirtualBox

| Cấu hình máy ảo để cài đặt hệ điều hành Raspbian OS

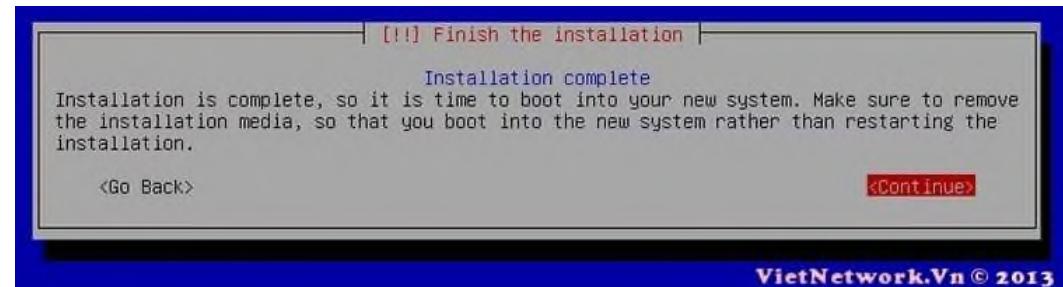
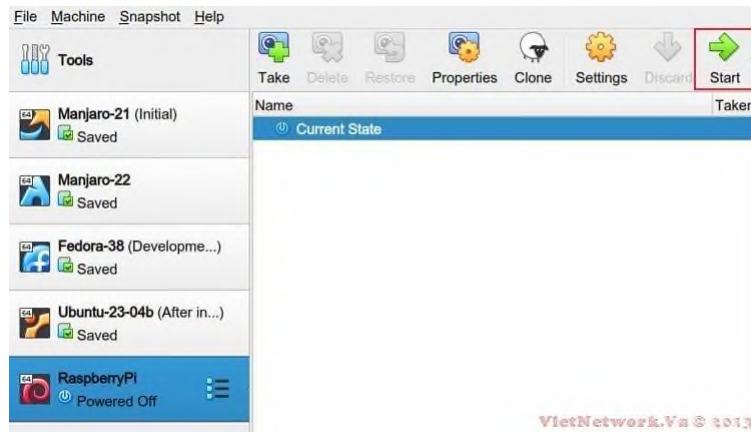
*Tham khảo link: <https://vietnetwork.vn/virtualization/cach-chay-raspberry-pi-trong-virtualbox/>*



## Thiết lập máy ảo Raspberry Pi trong VirtualBox

### Cài đặt hệ điều hành Raspbian OS trên máy ảo

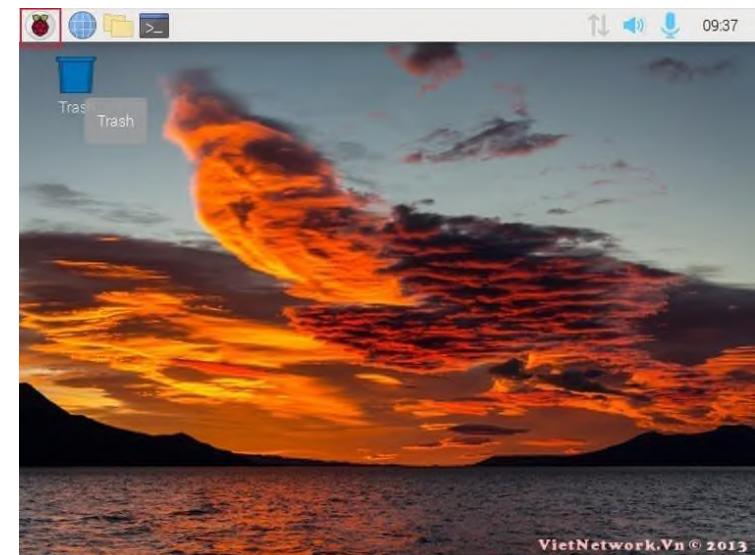
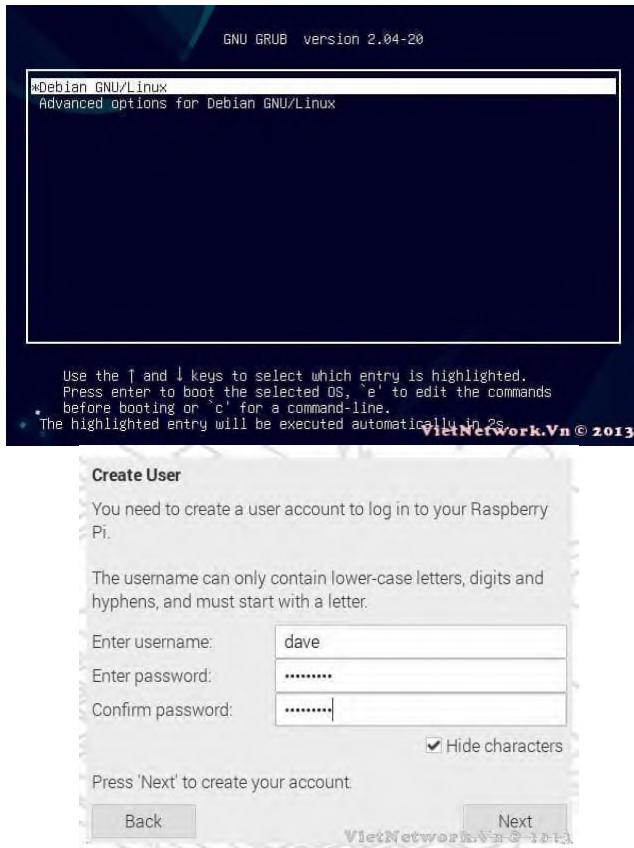
Tham khảo link: <https://vietnetwork.vn/virtualization/cach-chay-raspberry-pi-trong-virtualbox/>



## Thiết lập máy ảo Raspberry Pi trong VirtualBox

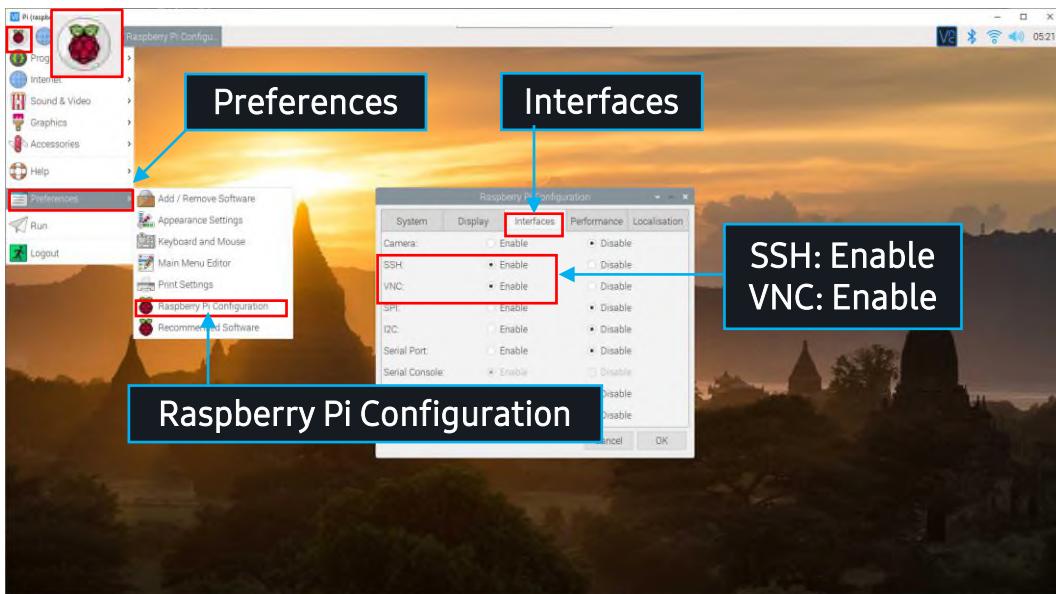
Chạy hệ điều hành Raspbian OS trên máy ảo

*Tham khảo link: <https://vietnetwork.vn/virtualization/cach-chay-raspberry-pi-trong-virtualbox/>*



## Sẵn sàng kết nối với Raspberry Pi OS

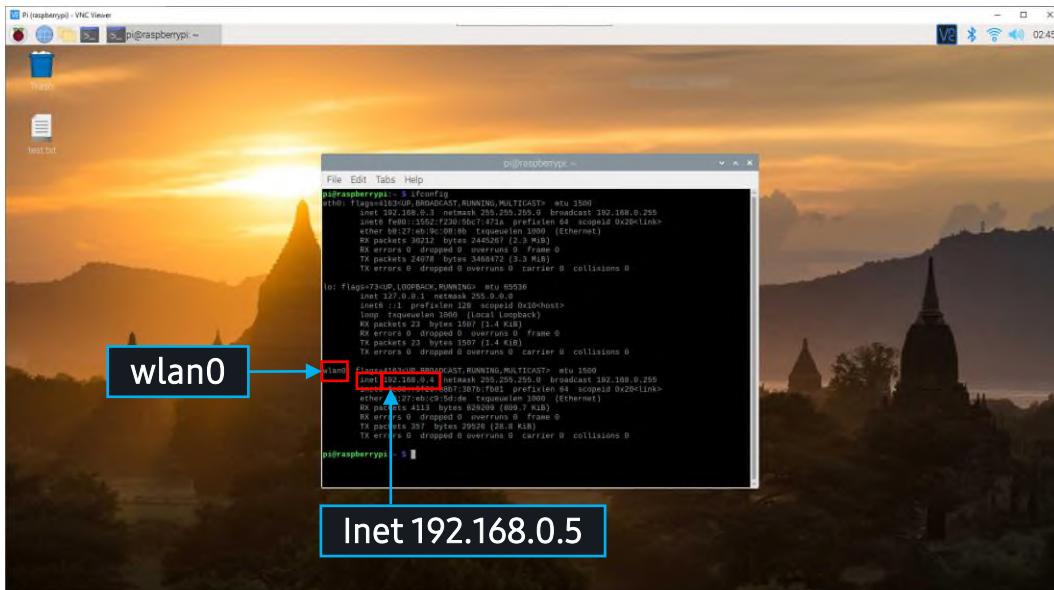
### Thiết lập hệ điều hành Raspberry Pi với Raspberry Pi



- ▶ Thiết lập Raspberry Pi bằng cách kết nối bàn phím và chuột.
- ▶ Tìm hiểu một cách dễ dàng hơn để sử dụng nó từ PC của bạn thay vì điều khiển nó bằng Giao diện I/O (ví dụ: bàn phím, chuột...) được kết nối trực tiếp với Raspberry Pi.
- ▶ Trong số một số cách để kết nối Raspberry Pi từ xa, chúng ta sẽ sử dụng SSH và VNC.
- ▶ Để kích hoạt SSH và VNC,
- ▶ Nhấp vào menu Raspberry Pi:
- ▶ → Chọn Tùy chọn
- ▶ → Chọn Giao diện cấu hình Raspberry Pi
- ▶ → Tick vào Enable trong SSH và VNC

## Sẵn sàng kết nối với Raspberry Pi OS

## Thiết lập hệ điều hành Raspberry Pi với Raspberry Pi



- ▶ Để kết nối Raspberry Pi với thiết bị khác, cần biết địa chỉ IP của Raspberry Pi.
  - ▶ Để tìm địa chỉ IP của Raspberry Pi, hãy mở cửa sổ đầu cuối và nhập ifconfig.

```
$ ifconfig
```

GHI CHÚ

Ghi lại địa chỉ IP inet dưới wlan0 để sử dụng sau này.

Trong trường hợp thiết bị của bạn bị ngắt kết nối và bạn kết nối lại, địa chỉ IP có thể thay đổi. Trong trường hợp này, hãy chạy **ifconfig** để tìm địa chỉ IP.

## Điều khiển Raspberry Pi bằng PC Windows - Sử dụng PuTTY

### Kết nối Raspberry Pi với PC Windows bằng PuTTY

- ▶ Tìm hiểu cách điều khiển Raspberry Pi từ xa từ máy tính Windows và dễ dàng truyền tệp mà không cần kết nối trực tiếp bàn phím, chuột và màn hình với Raspberry Pi.
- ▶ Có thể truy cập màn hình Raspberry Pi hoặc dòng lệnh của bạn từ bất kỳ đâu trong cùng một mạng bằng giao thức Secure Shell (SSH). Putty là cách tốt nhất để truy cập vào dòng lệnh. Nó được sử dụng rộng rãi như phần mềm SSH Client trong Windows.
- ▶ Truy cập trang web chính thức của PuTTY <https://www.putty.org/> trong PC Windows của bạn. Tìm "Tải xuống PuTTY" bên dưới và nhấp vào 'đây'.

 **Tải về PuTTY**

PuTTY là một ứng dụng khách SSH và telnet, ban đầu được phát triển bởi Simon Tatham cho nền tảng Windows.

PuTTY là phần mềm nguồn mở có sẵn mã nguồn và được phát triển và hỗ trợ bởi một nhóm tình nguyện viên.

Bạn có thể tải về PuTTY [here](#).

## Điều khiển Raspberry Pi bằng PC Windows - Sử dụng PuTTY

### Kết nối Raspberry Pi với PC Windows bằng PuTTY

- Chọn phiên bản PuTTY theo Windows của bạn. (Tìm một trong MSI ('Windows Installer'))
- Nếu bạn không chắc chắn về hệ điều hành máy tính của mình, hãy chọn '64-bit x86' hoặc '32-bit x86' vì Windows chạy bộ xử lý x86 phổ biến. Nhấp vào tệp .msi theo phiên bản hệ điều hành máy tính của bạn.

#### Gói tập tin

Bạn có thể muốn một trong số này. Chúng bao gồm các phiên bản của tất cả các tiện ích PuTTY. You can want to a in this number. Chúng bao gồm các phiên bản của tất cả các tiện ích PuTTY. (Không chắc bạn muốn phiên bản 32-bit hay 64-bit? Hãy đọc mục [FAQ entry](#).)

#### MSI ('Windows Installer')

64-bit x86:	<a href="#">putty-64bit-0.76-installer.msi</a>	(or by FTP)	(signature)
64-bit Arm:	<a href="#">putty-64bit-0.76-installer.msi</a>	(or by FTP)	(signature)
32-bit x86:	<a href="#">putty-0.76-installer.msi</a>	(or by FTP)	(signature)

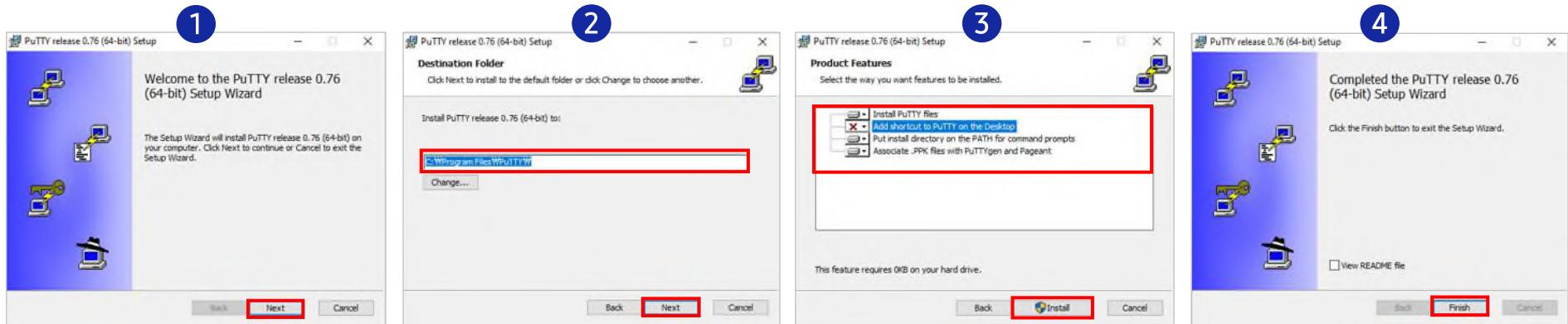
#### Unix source archive

.tar.gz:	<a href="#">putty-0.76.tar.gz</a>	(or by FTP)	(signature)
----------	-----------------------------------	-------------	-------------

## Điều khiển Raspberry Pi bằng PC Windows - Sử dụng PuTTY

### Kết nối Raspberry Pi với PC Windows bằng PuTTY

- ▶ Khi bạn thực thi tệp đã tải xuống, màn hình cài đặt PuTTY sau sẽ xuất hiện.
- ▶ Để cài đặt PuTTY,
  - ① Ấn Next
  - ② Đặt vị trí thư mục (đường dẫn tệp) để cài đặt nó
  - ③ Cài đặt sai khi kiểm tra tùy chọn
  - ④ Ấn Next để kết thúc cài đặt

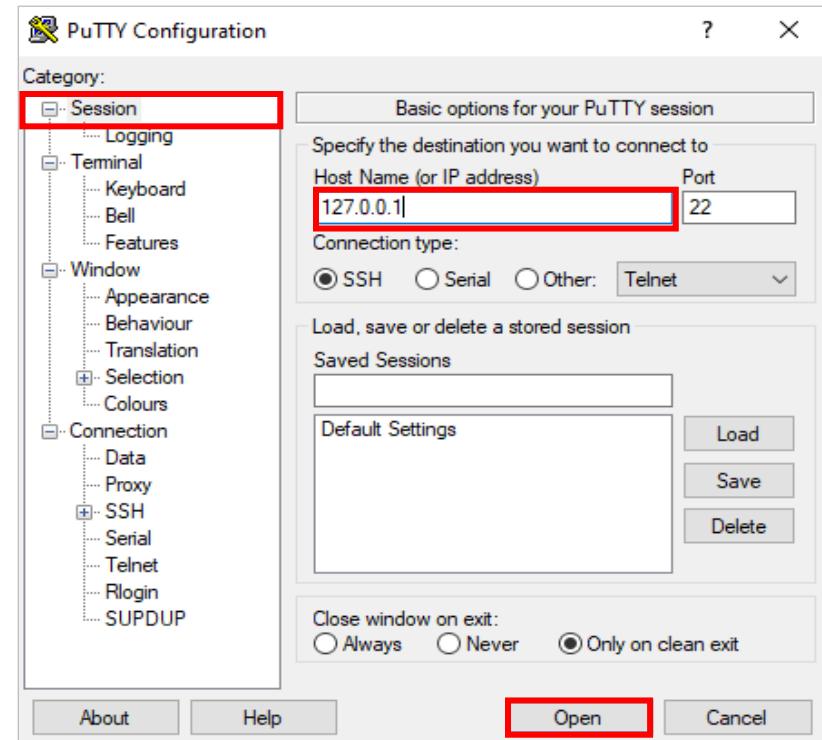
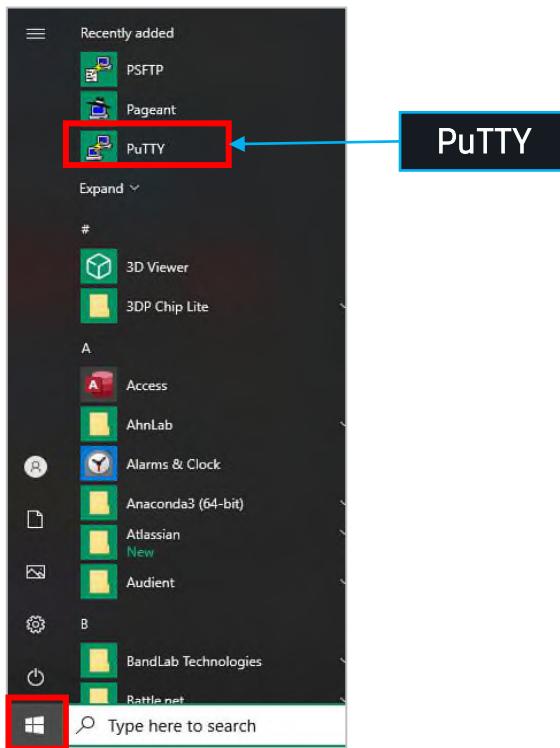


## Điều khiển Raspberry Pi bằng PC Windows - Sử dụng PuTTY

### Kết nối Raspberry Pi với PC Windows bằng PuTTY

- Khi PuTTY được cài đặt, bạn có thể tìm thấy PuTTY trong menu Bắt đầu. Nhấn vào nó để chạy.

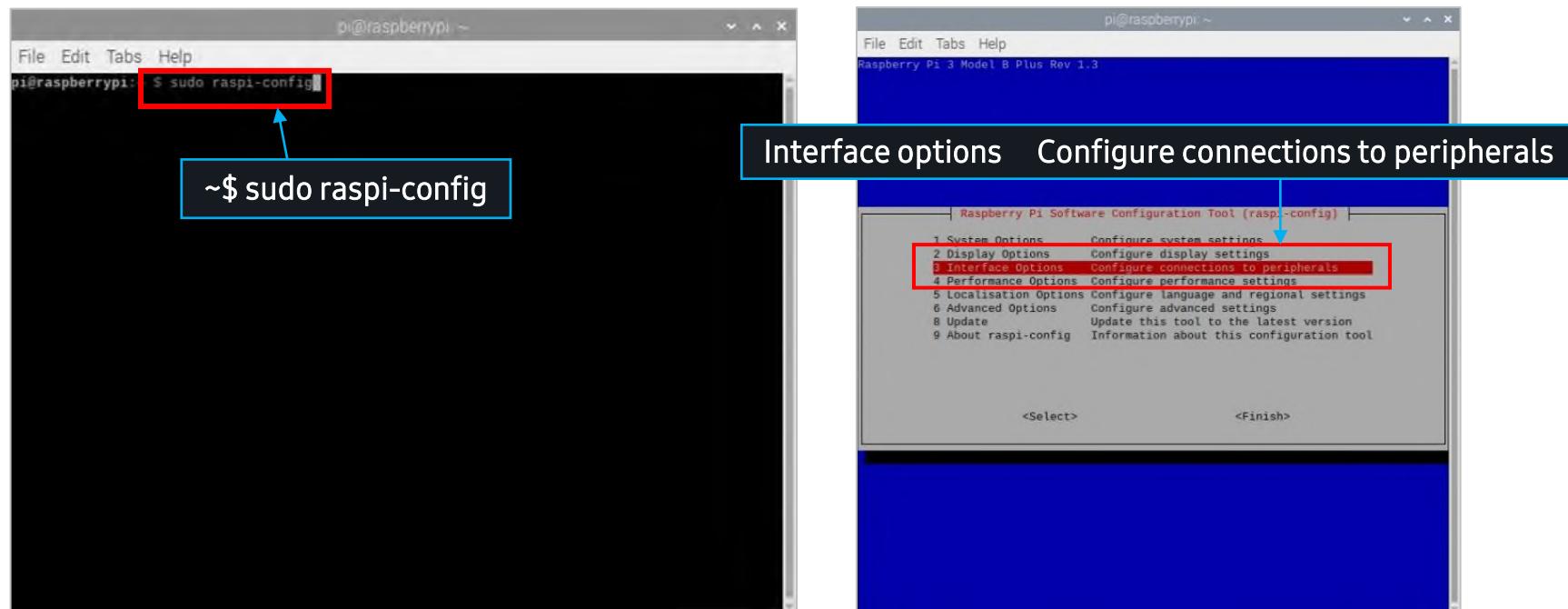
- Nhập địa chỉ Raspberry Pi vào Tên máy chủ và nhấp vào Mở. (\* Nhập địa chỉ inet của wlan0 mà bạn đã ghi chú. Tham khảo Trang 27.)



## Điều khiển Raspberry Pi bằng PC Windows - Sử dụng PuTTY

### Kết nối Raspberry Pi với PC Windows bằng PuTTY

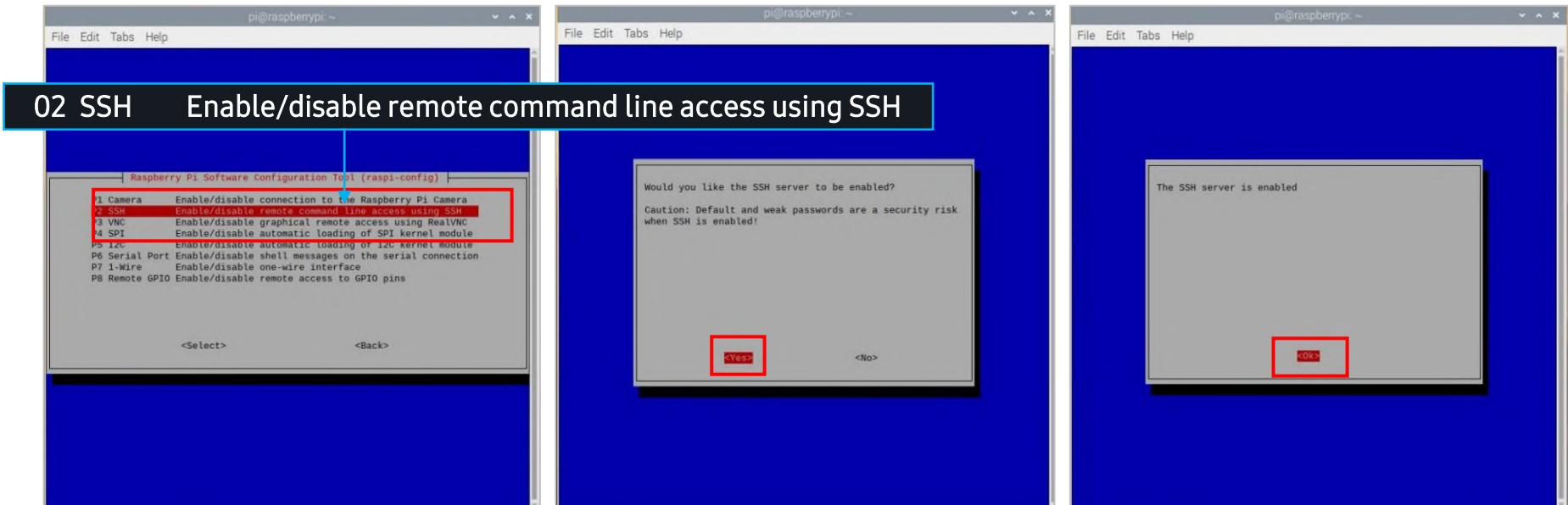
- ▶ Nếu thông báo Lỗi kết nối bị từ chối xuất hiện khi kết nối, hãy nhập sudo raspi-config trong thiết bị đầu cuối Raspberry Pi.
- ▶ Chọn 3 Tùy chọn Giao diện.



## Điều khiển Raspberry Pi bằng PC Windows - Sử dụng PuTTY

### Kết nối Raspberry Pi với PC Windows bằng PuTTY

- Chọn SSH → Yes → Ok như sau



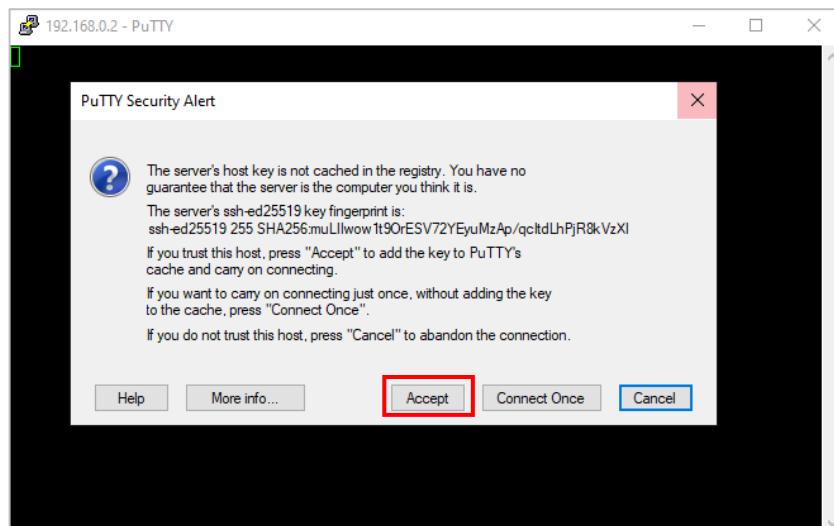
## Điều khiển Raspberry Pi bằng PC Windows - Sử dụng PuTTY

### Kết nối Raspberry Pi với PC Windows bằng PuTTY

- Nhấn Accept

※ Thông báo “Khóa máy chủ của máy chủ không được lưu trong bộ nhớ cache trong kho lưu trữ...” là không cần thiết để hiểu. Bạn có thể bỏ qua nó và nhấp vào Accept.

- Nhập pi để đăng nhập dưới dạng.
- Đối với mật khẩu, hãy nhập mật khẩu bạn đã sử dụng khi cài đặt hệ điều hành Raspberry Pi. (Mặc dù bạn nhập mật khẩu nhưng nó không hiển thị.)
- Nếu màn hình xuất hiện như bên dưới, Raspberry Pi đã được kết nối thành công với PC Windows của bạn.



## Điều khiển Raspberry Pi bằng PC Windows - Sử dụng PuTTY

### Kết nối Raspberry Pi với PC Windows bằng PuTTY



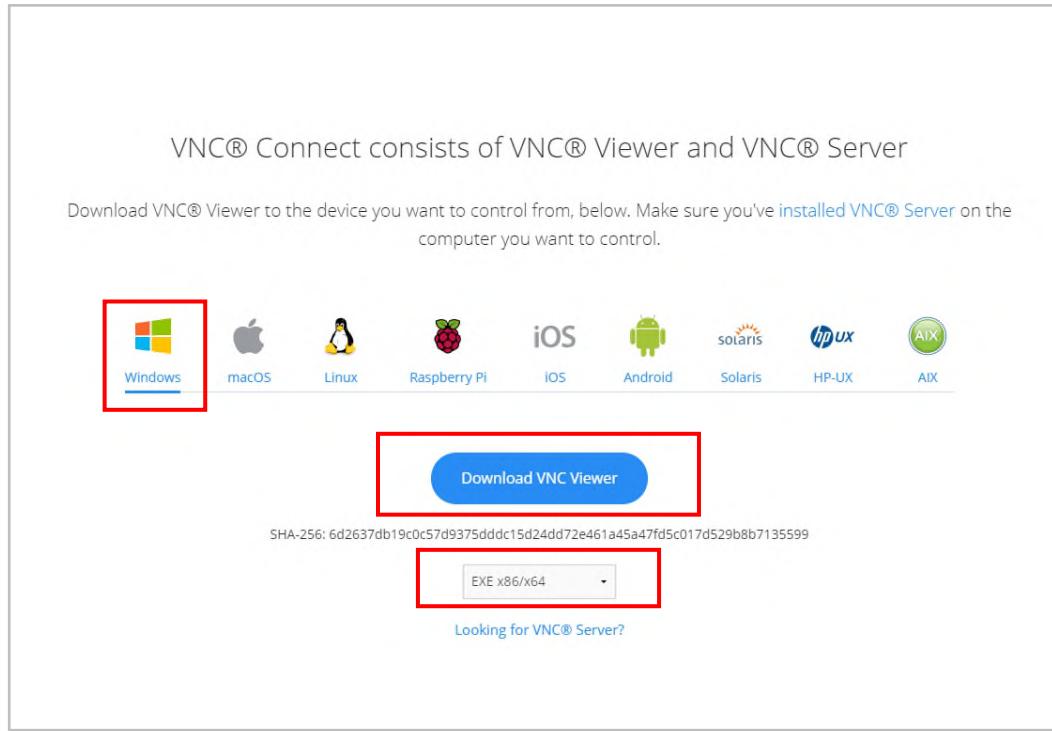
```
pi@raspberrypi: ~ $ cd ..
pi@raspberrypi: /home $ ls
pi
pi@raspberrypi: /home $
```

- ▶ Bạn có thể xác nhận Raspberry được kết nối thành công với thông báo xuất hiện trên màn hình.
- ▶ Tên thư mục được gọi là pi tồn tại trong thư mục máy tính để bàn của Raspberry Pi, kết nối được thực hiện thành công.

```
$ cd ..
$ ls
pi
```

## Điều khiển Raspberry Pi bằng PC Windows – Sử dụng VNC

### Kết nối Raspberry Pi với PC Windows bằng VNC

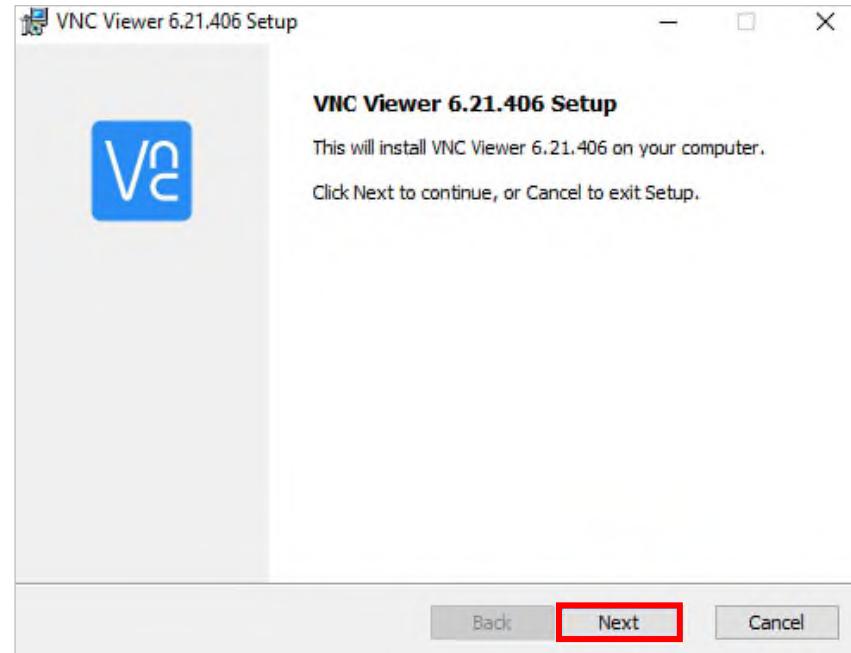
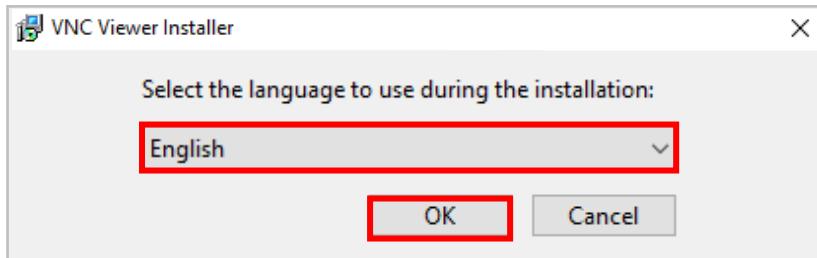


- ▶ Đối với những người mới sử dụng Linux, Giao diện người dùng đồ họa (GUI) sẽ hữu ích.
- ▶ (※ GUI: bất kỳ cách trực quan nào để truy cập máy tính hoặc chương trình)
- ▶ Virtual Network Computing (VNC) Viewer là một phương pháp dễ sử dụng để có quyền truy cập đầy đủ vào GUI trên bất kỳ máy nào bạn muốn kết nối.
- ▶ Chúng tôi sẽ sử dụng RealVNC trong số nhiều Trình VNC khác nhau.
- ▶ Để tải RealVNC Viewer, truy cập <https://www.realvnc.com/en/connect/download/viewer>. Sau khi chọn HĐH máy tính (Windows) muốn kết nối, nhấn Download VNC Viewer.

# Điều khiển Raspberry Pi bằng PC Windows – Sử dụng VNC

## Kết nối Raspberry Pi với PC Windows bằng VNC

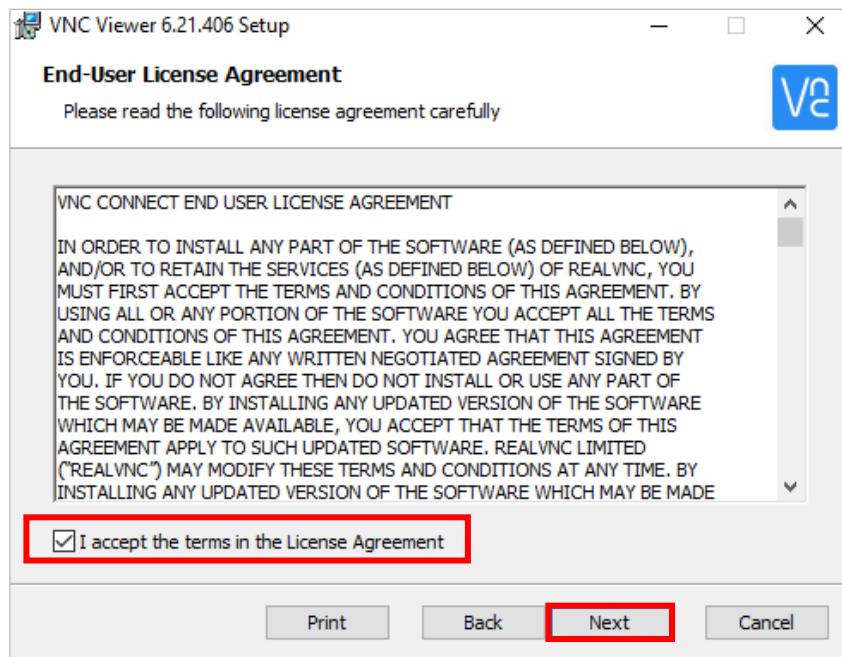
- Chọn ngôn ngữ và bấm OK.
- Nhấn Next để thiết lập VNC Viewer.



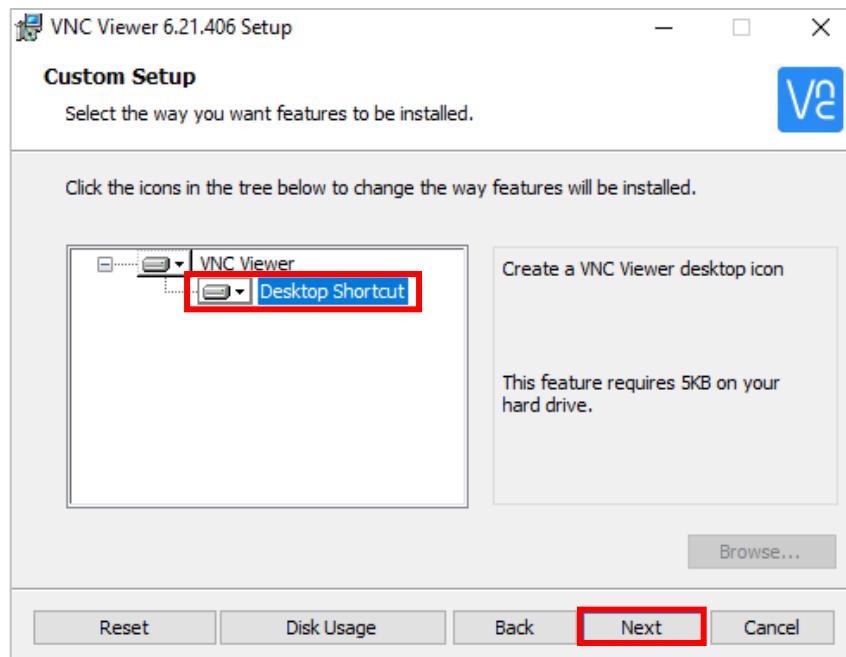
## Điều khiển Raspberry Pi bằng PC Windows – Sử dụng VNC

### Kết nối Raspberry Pi với PC Windows bằng VNC

- Đánh dấu vào License Agreement và nhấp vào Next.

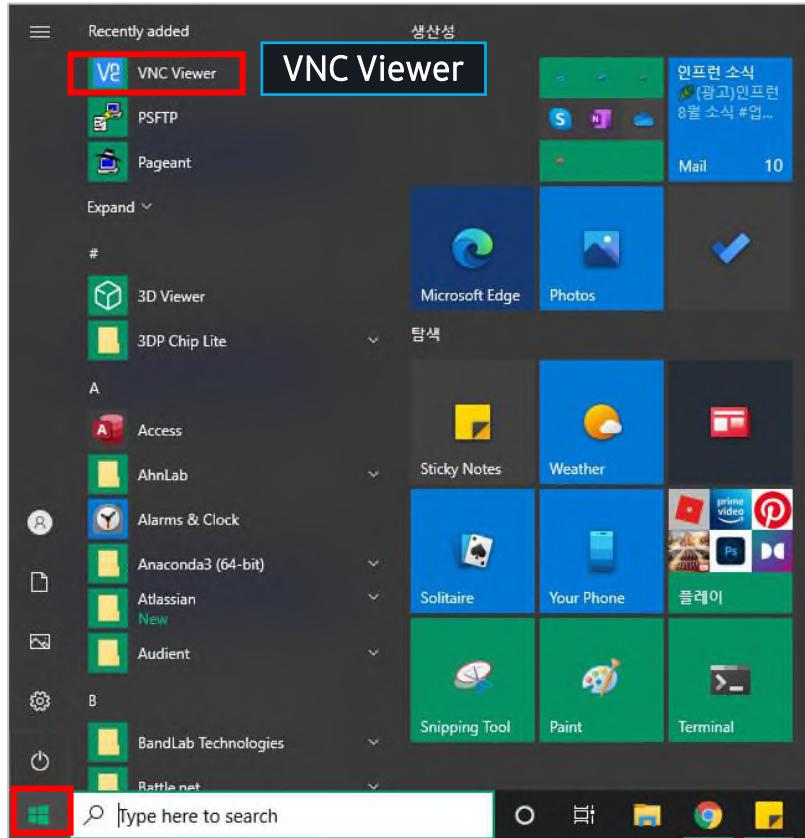


- Để dễ dàng sử dụng, hãy tạo Desktop Shortcut. Bấm Next



## Điều khiển Raspberry Pi bằng PC Windows – Sử dụng VNC

### Kết nối Raspberry Pi với PC Windows bằng VNC



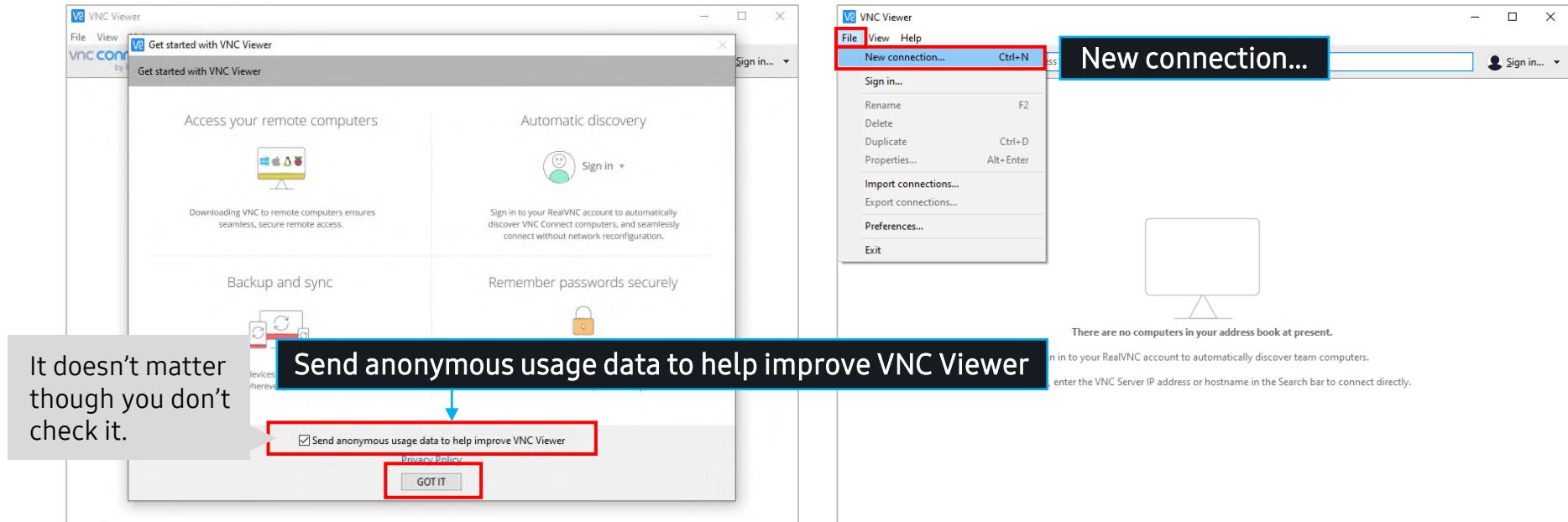
- Để thực thi từ lối tắt, hãy nhấp vào biểu tượng cửa sổ ở góc dưới cùng bên trái của Màn hình nền và nhấp vào VNC Viewer.



## Điều khiển Raspberry Pi bằng PC Windows – Sử dụng VNC

### Kết nối Raspberry Pi với PC Windows bằng VNC

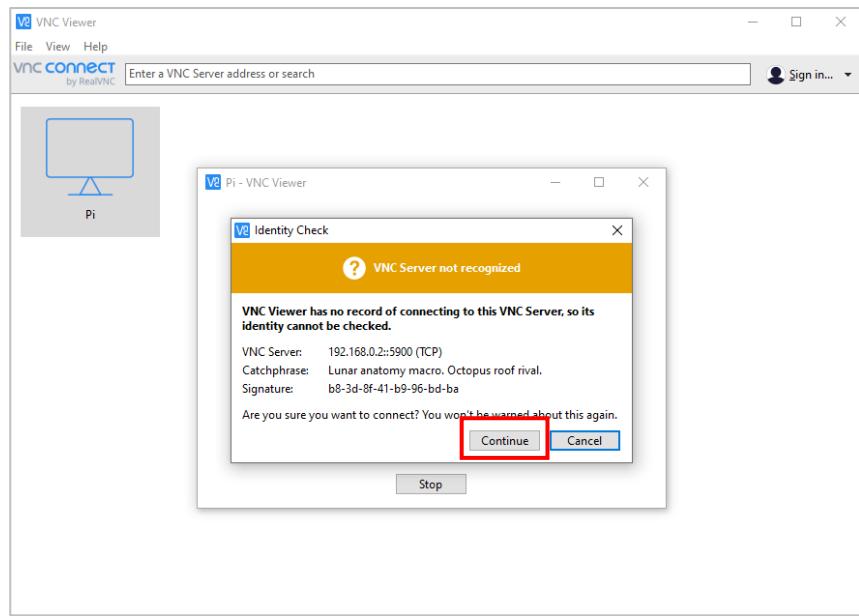
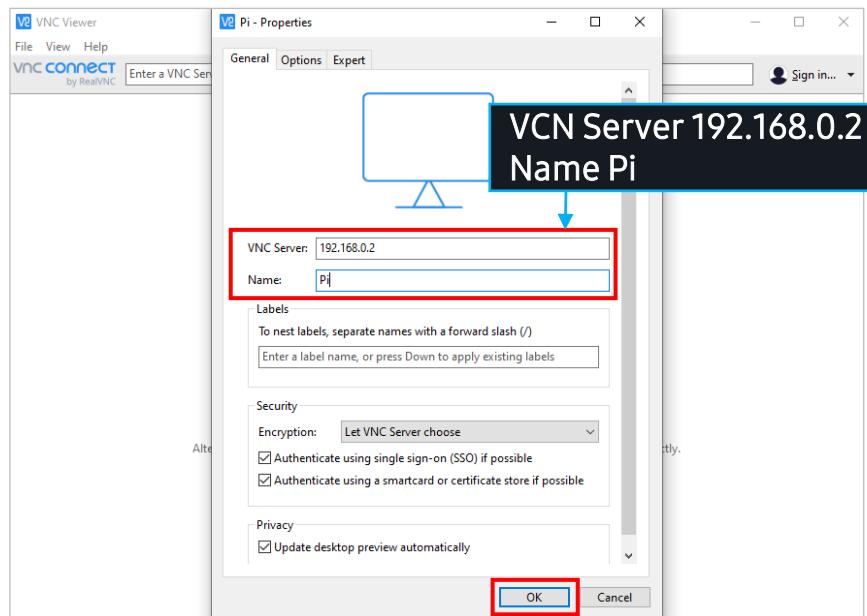
- Đánh dấu vào ô Gửi dữ liệu sử dụng ẩn danh .... và nhấp vào GOT IT.
- Cuối cùng bạn đã truy cập được VNC Viewer. Vì chưa có máy tính nào được kết nối, hãy kết nối nó với PC. Mở menu Tệp và nhấp vào Kết nối mới.



## Điều khiển Raspberry Pi bằng PC Windows – Sử dụng VNC

### Kết nối Raspberry Pi với PC Windows bằng VNC

- ▶ VNC Server: Nhập địa chỉ IP của wlan0 được sử dụng để thiết lập lại Raspberry Pi. (※Tham khảo trang 27)
- ▶ Đối với Tên, bạn có thể nhập bất cứ thứ gì bạn muốn, nhưng để thuận tiện, hãy đặt nó là Pi và nhấn vào OK.
- ▶ Thông báo 'Máy chủ VNC không được nhận dạng' bật lên. Điều này đề cập rằng không có hồ sơ kết nối.
- ▶ Nhấn vào Tiếp tục.

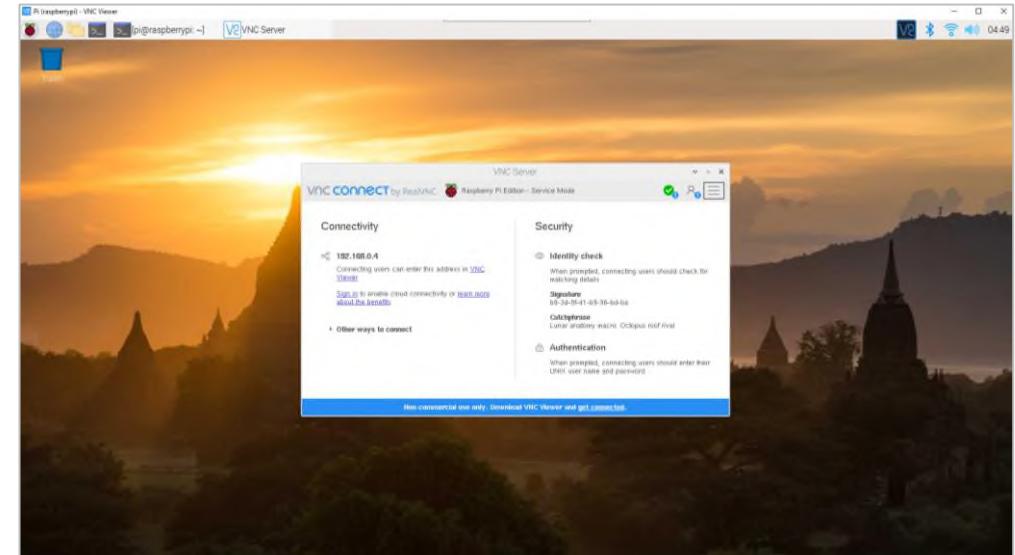
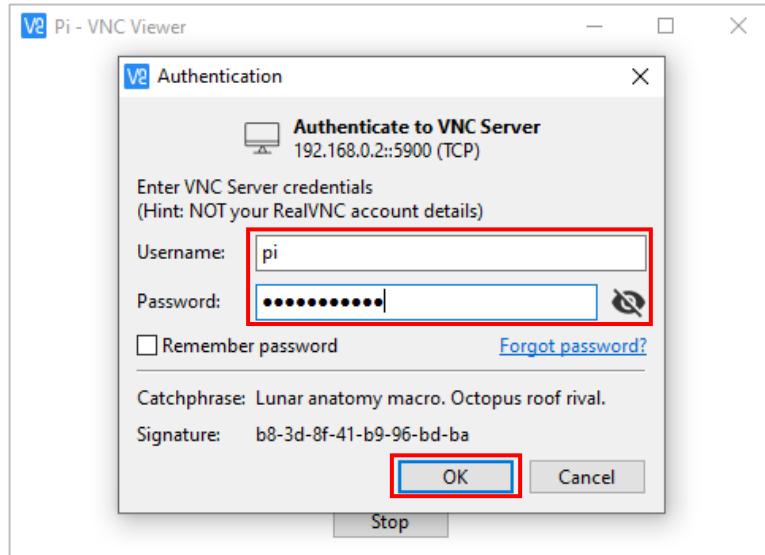


## Điều khiển Raspberry Pi bằng PC Windows – Sử dụng VNC

### Kết nối Raspberry Pi với PC Windows bằng VNC

- Tên người dùng: pi
- Mật khẩu: Nhập mật khẩu bạn đã sử dụng khi thiết lập Raspberry Pi.
- Nhấn OK

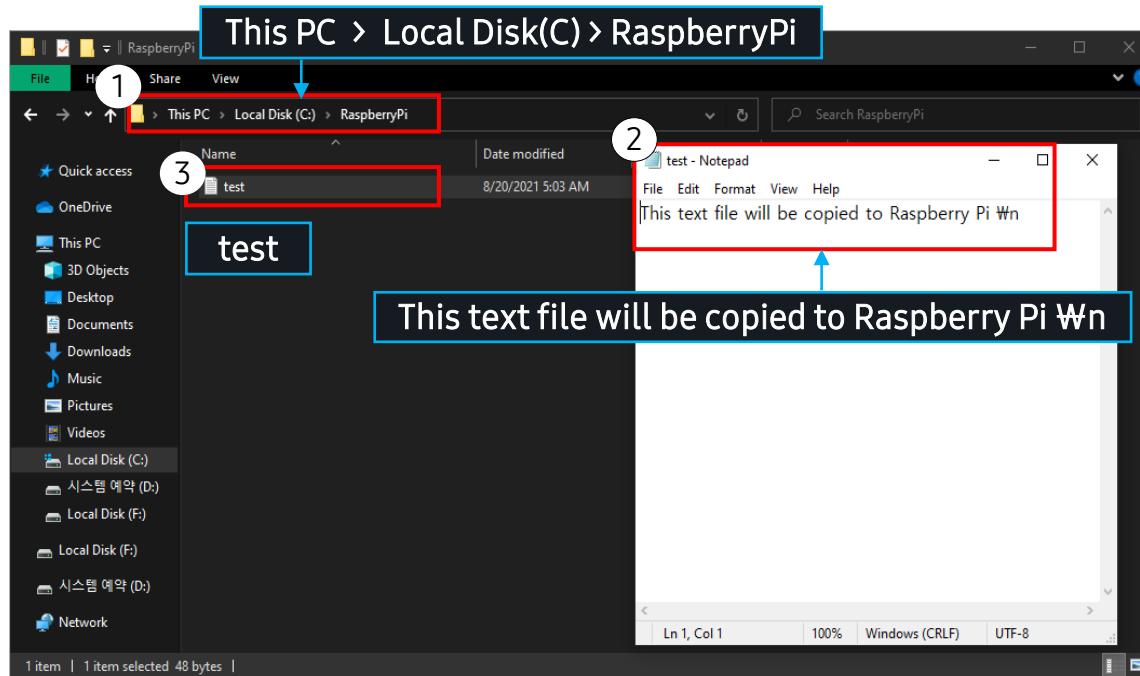
- Nếu Raspberry Pi được kết nối thành công với PC, màn hình PC của bạn sẽ hiển thị như bên dưới.
- Đây là hệ điều hành Raspberry Pi giống với hệ điều hành bạn đã kiểm tra trên màn hình của Raspberry Pi mà bạn đã kết nối trước đó.



## Điều khiển Raspberry Pi bằng PC Windows – Sử dụng VNC

### Gửi tệp từ Windows sang Raspberry Pi bằng VNC

- ▶ Tìm hiểu cách gửi tệp tới Raspberry Pi từ PC Windows bằng VNC.

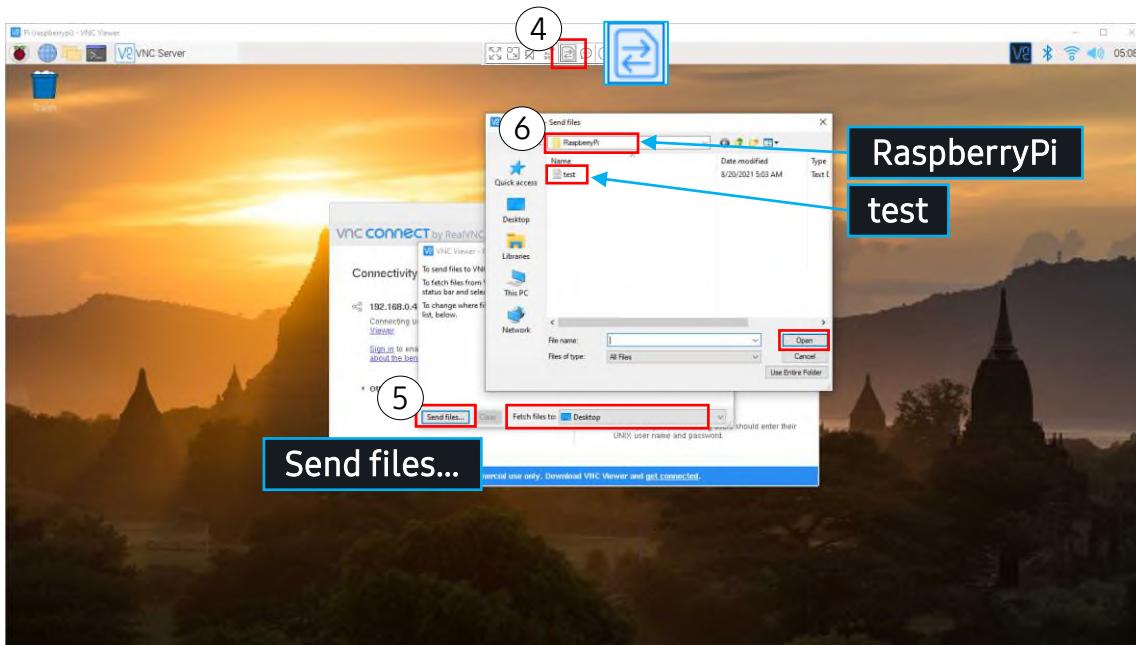


- ① Tạo thư mục 'RaspberryPi' ở vị trí mong muốn trong PC Windows của bạn (This PC) để quản lý tệp dễ dàng.
- ② Tạo tệp txt dưới dạng 'kiểm tra'
- ③ Lưu nó vào đúng vị trí.

## Điều khiển Raspberry Pi bằng PC Windows – Sử dụng VNC

### Gửi tệp từ Windows sang Raspberry Pi bằng VNC

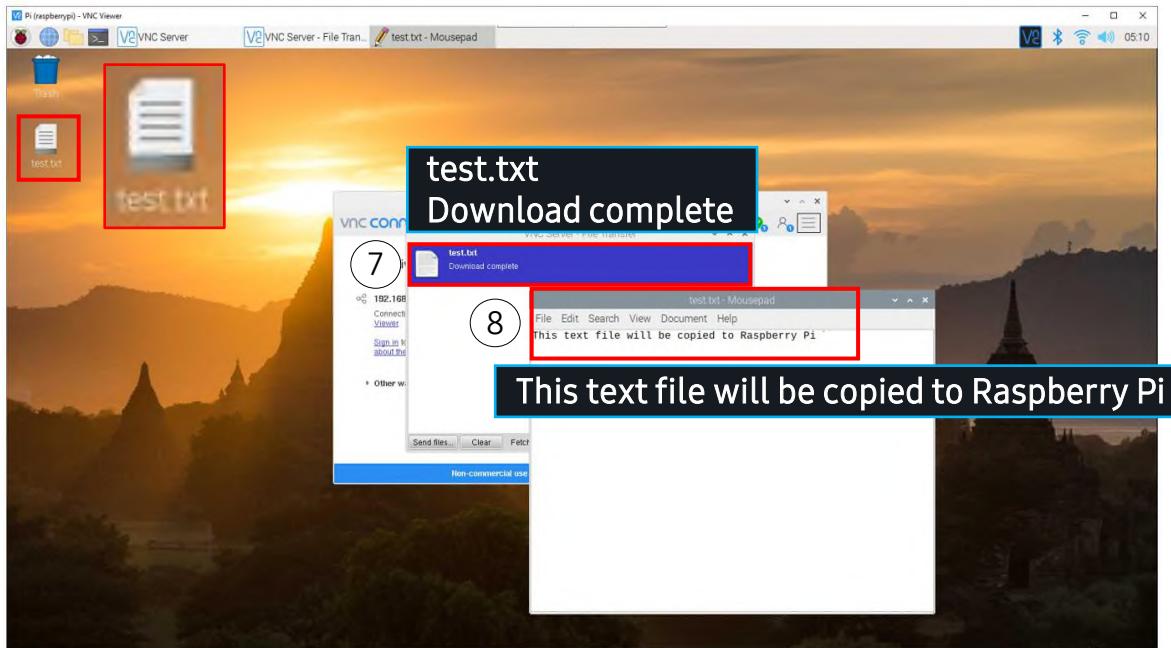
- Tìm hiểu cách gửi tệp tới Raspberry Pi từ PC Windows bằng VNC.



- Ở đầu màn hình Desktop của Raspberry Pi, menu xuất hiện như bên dưới. Di chuột qua nó để menu thả xuống. Nhấp vào biểu tượng Truyền tệp.
- Chỉ định Tùy chọn nạp tệp vào Desktop để bàn và nhấp vào Gửi tệp.
- Chọn tệp thử nghiệm bạn muốn gửi và nhấp vào Mở.

# Sử dụng Raspberry Pi với PC Windows

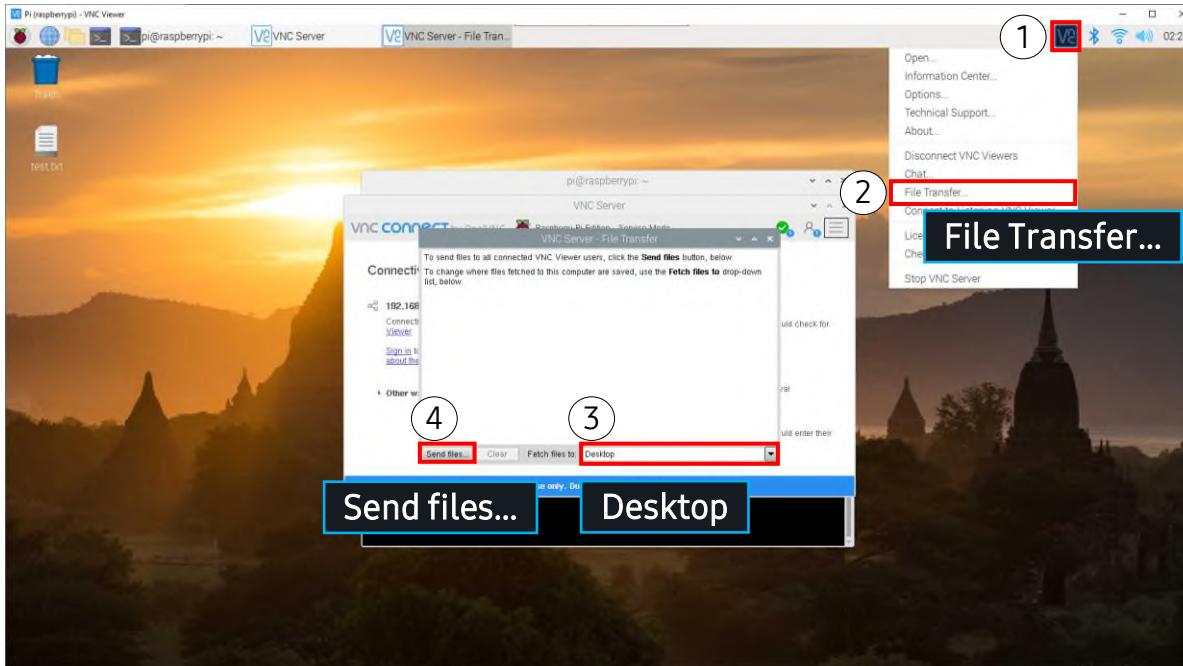
## I Gửi tệp từ Windows sang Raspberry Pi bằng VNC



- ⑦ Bằng cách nhấp vào Mở, VNC chạy Truyền tệp và bạn có thể thấy tệp 'kiểm tra' hiện có trong màn hình máy tính để bàn của Raspberry Pi.
- ⑧ Nhấp đúp vào tệp để mở.

## Sử dụng Raspberry Pi với PC Windows

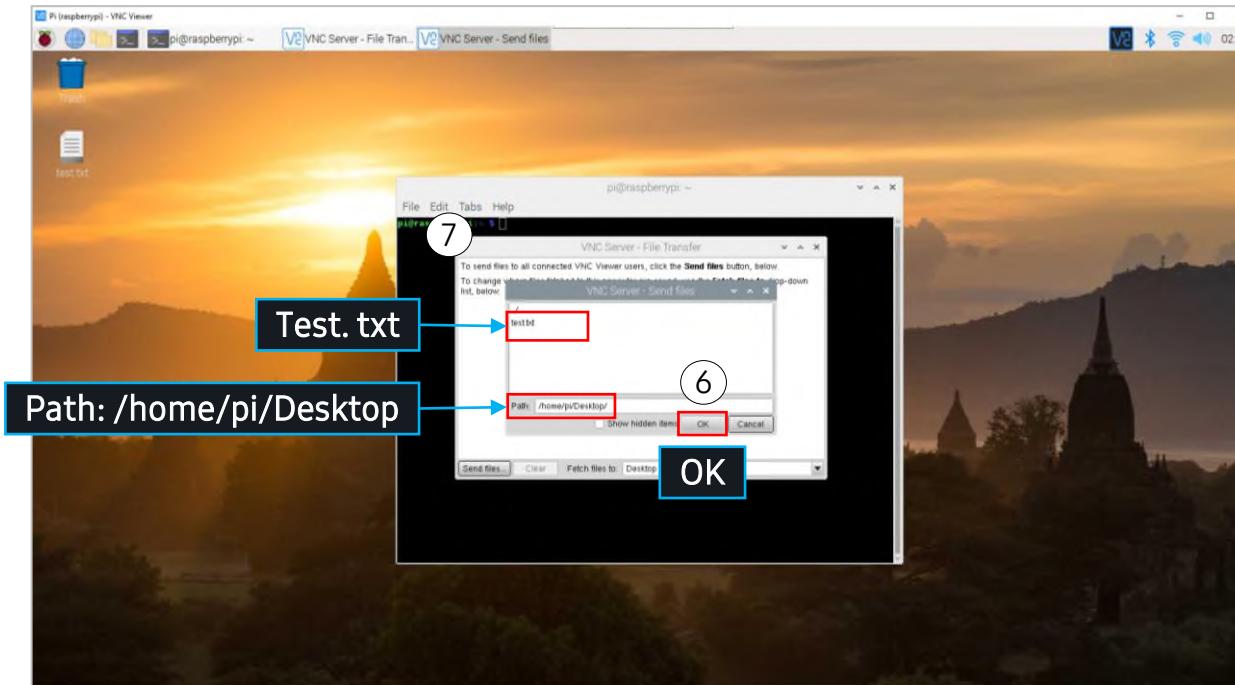
### Gửi tệp từ Windows sang Raspberry Pi bằng VNC



- ① Lần này, hãy gửi ngược lại tệp từ Raspberry Pi sang Windows.
- ② Nhấp chuột phải vào biểu tượng VNC ở trên cùng bên phải và menu thả xuống xuất hiện. Nhấp vào Truyền tệp.
- ③ Đặt Tìm nạp tệp thành Desktop.
- ④ Nhấn Gửi tệp.

## Sử dụng Raspberry Pi với PC Windows

### Gửi tệp từ Windows sang Raspberry Pi bằng VNC



- ⑥ Bạn có thể thấy tệp test.txt được gửi từ /home/pi/Desktop. Chọn tệp và nhấn OK.
- ⑦ Vào Windows Desktop của PC, bạn có thể thấy tệp test.txt tồn tại.

Bài 2.

# Hướng dẫn lệnh Raspberry Pi OS

- | 2.1. Kết nối với hệ điều hành Raspberry Pi từ máy tính  
của bạn
- | **2.2. Lệnh cơ bản**
- | 2.3. Các lệnh về Hệ thống tệp
- | 2.4. Các lệnh về điều khiển hệ thống
- | 2.5. Các lệnh về Network
- | 2.6. Các lệnh về Người dùng & Quyền
- | 2.7. Công cụ lập trình

## Lệnh cơ bản

Mục đích của bài này như sau:

- ▶ Tìm hiểu bash shell và cấu trúc dòng lệnh
- ▶ Tìm hiểu đường dẫn thường được sử dụng cho lệnh
- ▶ Tìm hiểu các lệnh được sử dụng để cài đặt và quản lý các gói
- ▶ Tìm hiểu các thư mục và các lệnh Linux cơ bản được sử dụng hữu ích
- ▶ Tìm hiểu bash shell và kiểm tra các chức năng hữu ích của bash shell
- ▶ Tìm hiểu và sử dụng các lệnh tìm kiếm, đồng thời kiểm tra các biểu thức chính quy của Linux để tìm kiếm
- ▶ Tìm hiểu và sử dụng các lệnh nén

## Bash Shell

### I Bash Shell là gì?

- ▶ Bash shell là shell cơ bản của hệ điều hành Raspberry Pi.
- ▶ Shell là một trình thông dịch dòng lệnh, nghĩa là nó là một chương trình nhận đầu vào văn bản và thực thi các lệnh trên hệ thống (nhân).
- ▶ Có rất nhiều shell được sử dụng phổ biến như Bourne Shell, C Shell, Korn Shell. Chúng ta sẽ tìm hiểu về các Lệnh Linux với Bash Shell.
- ▶ Bash Shell (Bourne Again Shell) là shell được sửa đổi từ Bourne Shell và Bash Shell, được sử dụng rộng rãi cho Linux.
- ▶ Lợi thế của Bash Shell:
  - Chức năng tự động hoàn thành lệnh bằng Tab
  - Chức năng lịch sử dòng lệnh
  - Chức năng bí danh
  - Phím tắt chỉnh sửa dòng

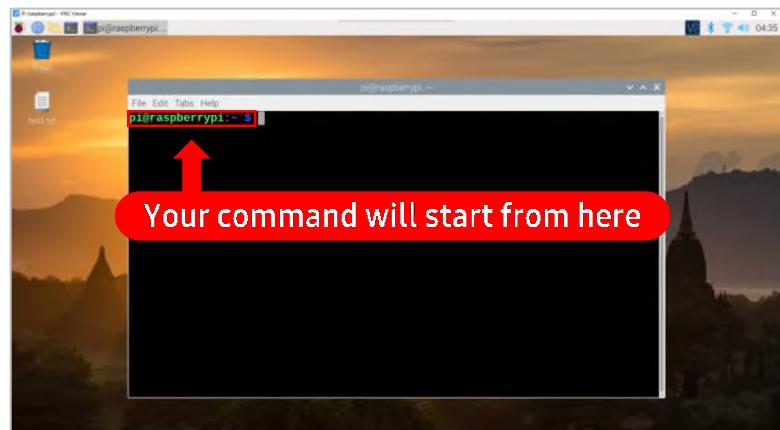
## Cấu trúc của dòng lệnh

### Lời nhắc

- ▶ Dấu hiệu chờ người dùng nhập lệnh.
- ▶ Bash shell: \$
- ▶ Quản trị hệ thống : #

### Thư mục cơ bản

- ▶ Thư mục mặc định của cửa sổ đầu cuối là thư mục nhà (Đăng nhập) của người dùng.
- ▶ pi@raspberrypi:~ \$
- ▶ pi : Tên tài khoản người dùng
- ▶ raspberrypi : Tên máy chủ
- ▶ ~ : Thư mục chính của người dùng
- ▶ \$ : dấu nhắc bash shell



## Lệnh Linux

### Cách chỉnh sửa dòng lệnh

- ▶ Nhập lệnh sau dấu nhắc.
- ▶ Xóa ký tự: Sử dụng nút Backspace hoặc Delete.
- ▶ Xóa từ: Ctrl + w, các từ cách nhau bởi khoảng trắng.
- ▶ Xóa câu: Ctrl + u

### Cấu trúc của một lệnh

Cấu trúc lệnh Linux

Mẫu	Lệnh [Option] [Arguments ...]
Các lệnh	Có hàng trăm lệnh khác nhau mà người dùng nhập để sử dụng Linux, bao gồm data, man, ls, cp, mv, v.v.
Tùy chọn	Bạn có thể sử dụng các tùy chọn để chọn chi tiết của lệnh. Các tùy chọn là một tính năng quan trọng để làm phong phú thêm chức năng của Linux và chúng có thể tồn tại hoặc không tồn tại tùy thuộc vào lệnh. Các tùy chọn bao gồm chữ thường hoặc chữ hoa bắt đầu bằng dấu -. Bạn nên tham khảo hướng dẫn của lệnh để xem nó có những tùy chọn nào và chức năng của nó là gì.
Tranh luận	Đối số là các giá trị được truyền cho lệnh, thường sử dụng tên tệp hoặc thư mục. Đối số có thể không tồn tại tùy thuộc vào lệnh và các đối số cần thiết cho mỗi lệnh là khác nhau, vì vậy bạn nên tham khảo hướng dẫn.

## Đường dẫn

- | Vị trí của một tệp hoặc thư mục cụ thể trong hệ thống phân cấp thư mục trong hệ thống tệp.
- | Sử dụng dấu gạch chéo (/) để phân tách từng đường dẫn trong tên đường dẫn.
- | Dấu gạch chéo (/) đầu tiên trong "/usr/bin/ls" đại diện cho thư mục gốc.
- | Dấu gạch chéo (/) ở giữa tên thư mục và tên tệp riêng biệt.
- | Có hai loại đường dẫn: đường dẫn tuyệt đối và đường dẫn tương đối.
- | Trong Windows, nếu bạn nhập thuộc tính thư mục, bạn có thể kiểm tra đường dẫn như trong cửa sổ bên phải.

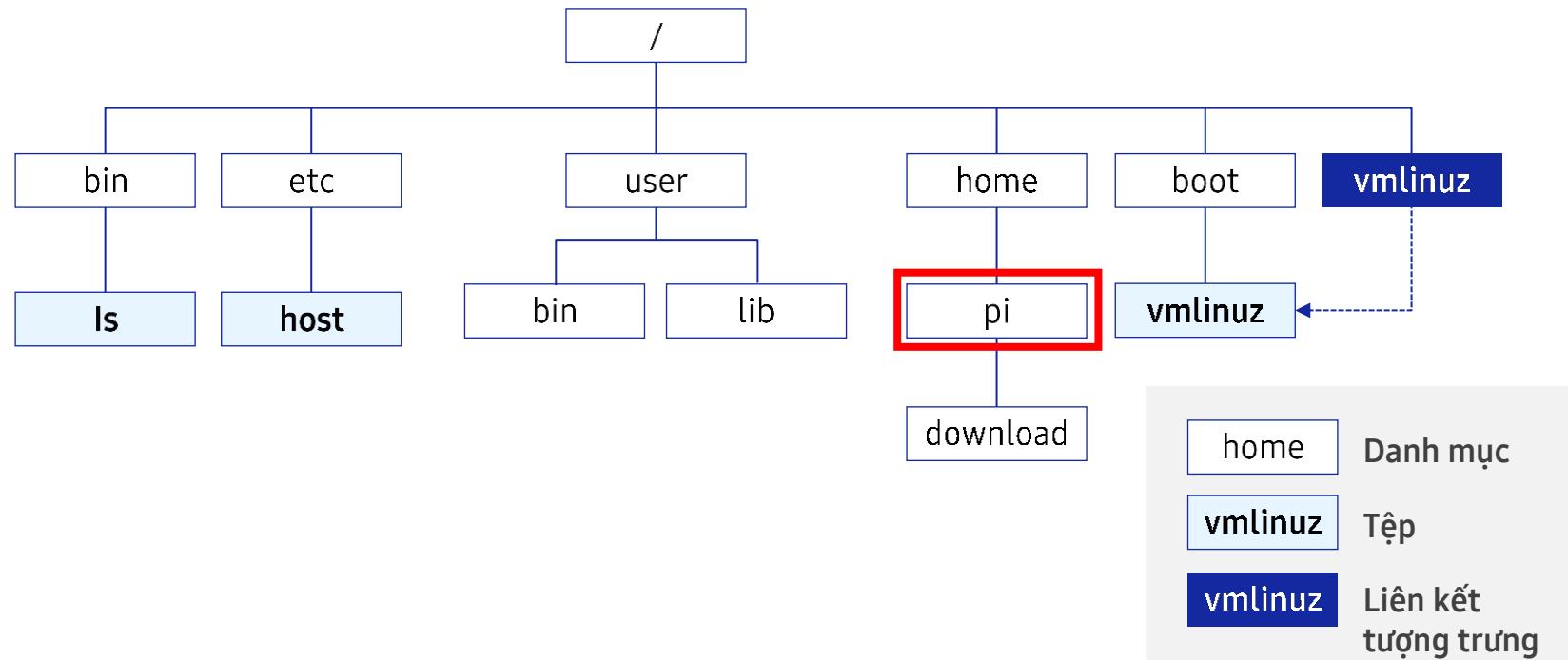
## Tên đường dẫn tuyệt đối

- | Luôn bắt đầu tại thư mục gốc (/).
- | Hiển thị tên của tất cả các thư mục trung gian di chuyển đến vị trí của tệp hoặc thư mục cụ thể.
- | Các tên đường dẫn tuyệt đối trở đến một vị trí cụ thể luôn giống nhau.

## Tên đường dẫn tương đối

- | Bắt đầu dựa trên thư mục hiện tại.
- | Bắt đầu bằng ký tự khác /.
- | Nếu bạn chuyển xuống thư mục con dựa trên thư mục hiện tại, chỉ cần thêm tên của thư mục con.
- | Thêm .. để chuyển đến thư mục mẹ dựa trên thư mục hiện tại.
- | Tên đường dẫn tương đối phụ thuộc vào vị trí của thư mục hiện tại.

## Làm quen với tên đường dẫn



## Khi thư mục làm việc hiện tại là “pi” (theo tên tài khoản người sử dụng)

- | Tên đường dẫn tuyệt đối của pi-> /home/pi
- | Tên đường dẫn tuyệt đối của 'tải xuống' trong pi-> /home/pi/download
- | Tên đường dẫn tương đối cho "download"->download hoặc ./download
- | Tên đường dẫn tương đối của tệp máy chủ-> ../../etc/hosts

## Quy ước đặt tên

- | Dấu gạch chéo (/) không được phép trong tên tệp và thư mục-> vì (/) được sử dụng làm dấu phân cách trong tên đường dẫn.
- | Chỉ sử dụng các ký tự chữ và số, dấu gạch dưới (-), dấu gạch dưới (\_) và dấu chấm(.) cho tên tệp và thư mục.
- | Không bao gồm các ký tự khoảng trắng, \*, |, "", @, #, \$, %, ^, &, v.v. trong tên tệp và thư mục.  
Phân biệt giữa chữ hoa và chữ thường.  
Nếu tên của tệp và thư mục bắt đầu bằng '.', chúng được coi là tệp ẩn.

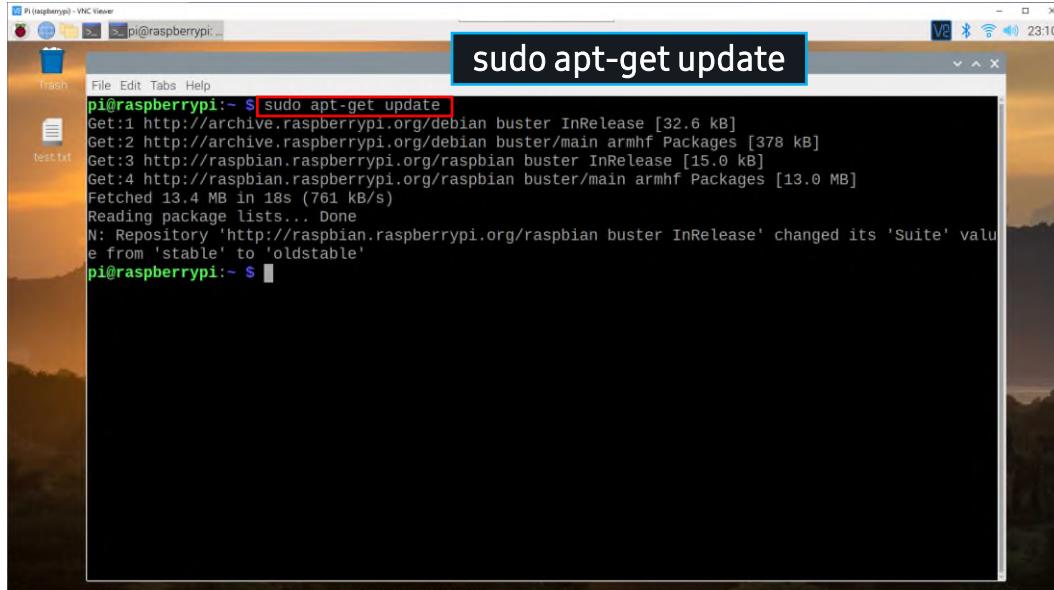
## Cài đặt và quản lý gói

### I apt-get(Công cụ đóng gói nâng cao)

- ▶ Khi cài đặt một chương trình mới trong Windows, bạn có thể dễ dàng tải xuống tệp cài đặt từ Internet để cài đặt và nếu không cần thiết, bạn có thể xóa tệp đó thông qua trình quản lý cài đặt trong bảng điều khiển.
- ▶ Công việc tương tự có thể được thực hiện ngay cả trong Raspberry Pi OS thông qua lệnh apt-get.
- ▶ apt-get là một công cụ lệnh quản lý gói được sử dụng trong Linux dựa trên Debian.
- ▶ Vì Raspberry Pi OS dựa trên Debian Linux, lệnh apt-get có thể được sử dụng như nhau.
- ▶ apt-get yêu cầu quyền root.

## Cài đặt và quản lý gói

### I Cập nhật gói thông tin



A screenshot of a VNC session showing a terminal window. The title bar says "Pi (raspberrypi) - VNC Viewer". The terminal window has a blue header bar with the text "sudo apt-get update". The main area of the terminal shows the output of the command:

```
pi@raspberrypi:~$ sudo apt-get update
Get:1 http://archive.raspberrypi.org/debian buster InRelease [32.6 kB]
Get:2 http://archive.raspberrypi.org/debian buster/main armhf Packages [378 kB]
Get:3 http://archive.raspberrypi.org/debian buster/main Sources [15.0 kB]
Get:4 http://raspbian.raspberrypi.org/raspbian buster InRelease [15.0 kB]
Fetched 13.4 MB in 18s (761 kB/s)
Reading package lists... Done
N: Repository 'http://raspbian.raspberrypi.org/raspbian buster InRelease' changed its 'suite' value from 'stable' to 'oldstable'
pi@raspberrypi:~$
```

- ▶ Lấy thông tin của gói cần cập nhật ở vị trí lưu trữ trong tệp /etc/apt/sources.list.
- ▶ Chỉ cập nhật thông tin của gói cần nâng cấp.
- ▶ Các gói đã cài đặt không được nâng cấp ngay lập tức.

## Cài đặt và quản lý gói

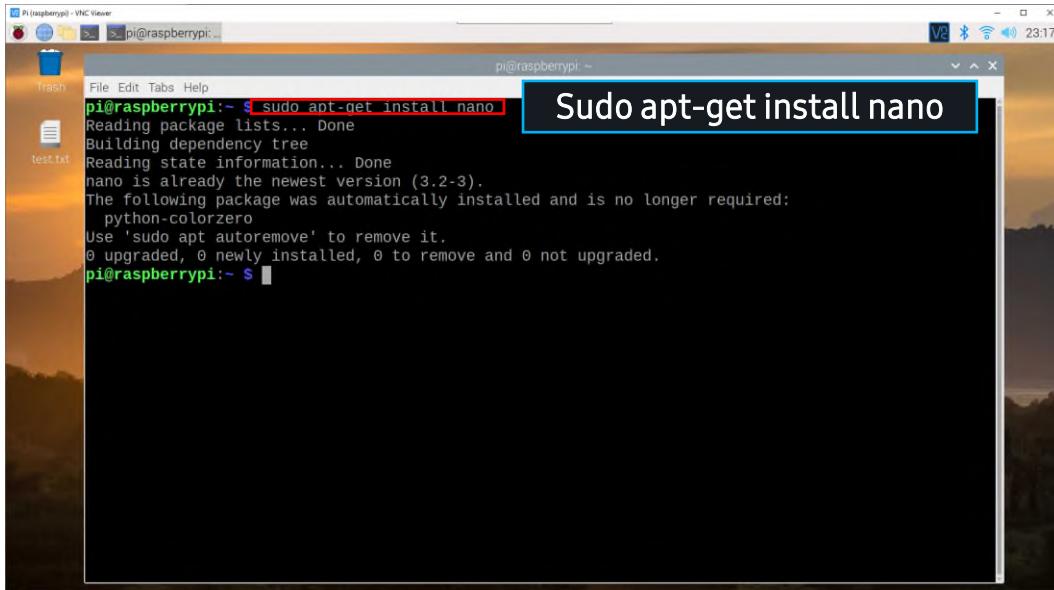
### Cài đặt toàn bộ Gói

```
pi@raspberrypi: ~$ sudo apt-get dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  python-colorzero
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi: ~$
```

- ▶ Cài đặt các gói thu được bằng cách sử dụng lệnh cập nhật apt-get trong HĐH Raspberry Pi.
- ▶ Mặc dù lệnh nâng cấp apt-get cũng là lệnh cài đặt các gói nhưng nó không kiểm tra sự phụ thuộc giữa các gói.
- ▶ Do đó, nếu có thể, hãy cài đặt gói bằng lệnh:  
*sudo apt-get dist-upgrade*

## Cài đặt và quản lý gói

### Cài đặt Gói



```
pi@raspberrypi: ~$ sudo apt-get install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
nano is already the newest version (3.2-3).
The following package was automatically installed and is no longer required:
  python-colorzero
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi: ~$
```

- ▶ Sử dụng cùng với sudo apt-get install <tên gói> và cài đặt gói được chỉ định trong tên gói.
- ▶ cài đặt apt-get chỉ cài đặt gói thuộc về tên gói được chỉ định.
- ▶ Ví dụ: hãy thử cài đặt trình soạn thảo văn bản có tên là nano. Nhưng vì nó đã được cài đặt sẵn nên bạn có thể tích vào cụm từ 'already the latest'.

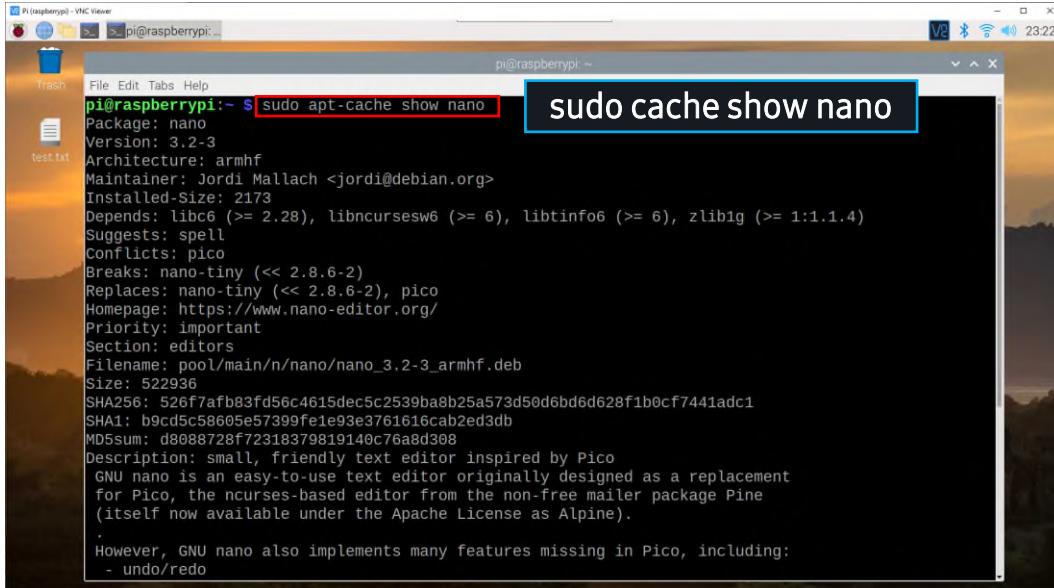
## Cài đặt và quản lý gói

### Xóa Gói

- ▶ Sử dụng cùng với sudo apt-get remove <tên gói> và xóa gói được chỉ định trong tên gói.
- ▶ Để xóa tất cả, ngay cả tệp cài đặt của gói, bạn có thể xóa tất cả thông tin cài đặt bằng cách sử dụng sudo apt-get –purge remove <tên gói>.
- ▶ Nó được sử dụng trong trường hợp không gian lưu trữ cực kỳ nhỏ hoặc trong trường hợp cần cài đặt lại gói do vấn đề phụ thuộc gây ra bởi sự khác biệt trong các phiên bản.

## Cài đặt và quản lý gói

### Xem thông tin Gói



```
pi@raspberrypi: ~$ sudo apt-cache show nano
Package: nano
Version: 3.2-3
Architecture: armhf
Maintainer: Jordi Mallach <jordi@debian.org>
Installed-Size: 2173
Depends: libc6 (>= 2.28), libncursesw6 (>= 6), libtinfo6 (>= 6), zlib1g (>= 1:1.1.4)
Suggests: spell
Conflicts: pico
Breaks: nano-tiny (<< 2.8.6-2)
Replaces: nano-tiny (<< 2.8.6-2), pico
Homepage: https://www.nano-editor.org/
Priority: important
Section: editors
Filename: pool/main/n/nano/nano_3.2-3_armhf.deb
Size: 522936
SHA256: 526f7af83fd56c4615dec5c2539ba8b25a573d50d6bd6d628f1b0cf7441adc1
SHA1: b9cd5c58605e57399fe1e93e3761616cab2ed3db
MD5sum: d8088728f72318379819140c76a8d308
Description: small, friendly text editor inspired by Pico
GNU nano is an easy-to-use text editor originally designed as a replacement
for Pico, the ncurses-based editor from the non-free mailer package Pine
(itself now available under the Apache License as Alpine).

However, GNU nano also implements many features missing in Pico, including:
- undo/redo
```

- Sử dụng cùng với sudo apt-cache show <tên gói> và có thể kiểm tra thông tin gói được chỉ định trong <tên gói>.

## Lệnh hệ thống tệp

| ls

```
pi@raspberrypi: ~ ls
Bookshelf Documents Music Public Videos
Desktop Downloads Pictures Templates
pi@raspberrypi: ~ ls -l
total 36
drwxr-xr-x 2 pi pi 4096 May  7 23:52 Bookshelf
drwxr-xr-x 2 pi pi 4096 Aug 20 05:09 Desktop
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Documents
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Downloads
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Music
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Pictures
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Public
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Templates
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Videos
pi@raspberrypi: ~ ls -a
. .cache .gnupg .profile .Xauthority
.. .config .local Public .xsession-errors
.bash_history .cups Music Templates .xsession-errors.old
.bash_logout Desktop Pictures .thumbnails
.bashrc Documents .pki Videos
Bookshelf Downloads .pp_backup .vnc
```

- ▶ Lệnh hiển thị danh sách tệp và thư mục.
- ▶ Bằng cách thêm tùy chọn phía sau lệnh, các chức năng được phân đoạn có thể được sử dụng nhưng đổi với tùy chọn, nó được sử dụng ở dạng –option.
- ▶ Có các tùy chọn khác nhau của lệnh ls như:
- ▶ -l: Đề cập đến dài. Liệt kê tất cả các Quyền, số lượng tệp được bao gồm, chủ sở hữu, nhóm, kích thước tệp, ngày chỉnh sửa, tên tệp và những thứ khác.
- ▶ -a: Tham khảo tất cả. Hiển thị tất cả các tệp hoặc thư mục ẩn.

```
$ ls
$ ls -l
$ ls -a
```

## Lệnh hữu ích

### clear

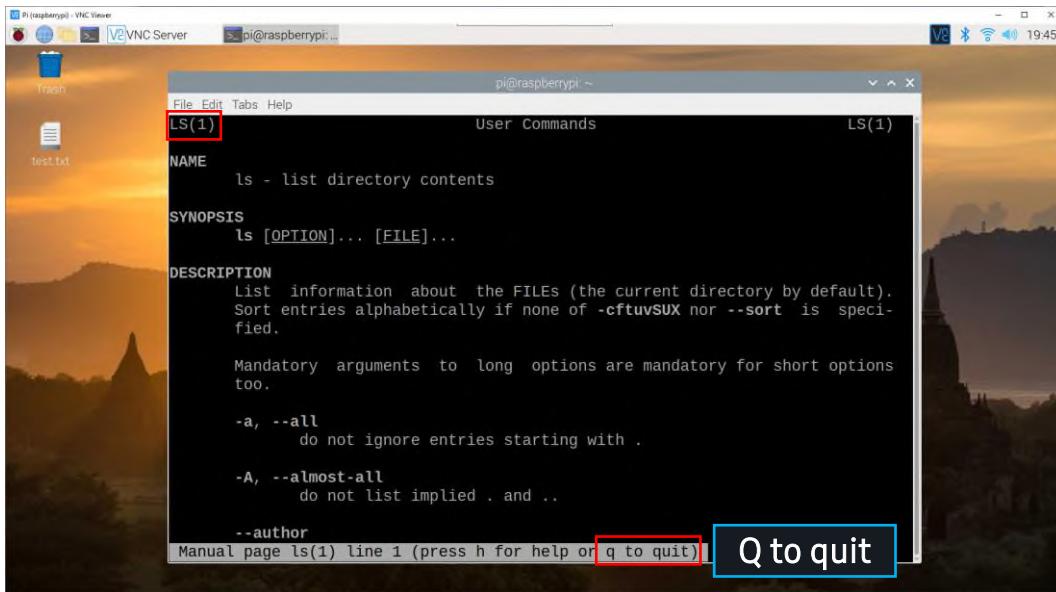
- ▶ Xóa màn hình đầu cuối.
- ▶ Để thực hiện lệnh này, chỉ cần nhập clear trong thiết bị đầu cuối.

### Lịch sử dòng lệnh

- ▶ Linux duy trì lịch sử các lệnh bạn đã nhập.
- ▶ Trong màn hình đã xóa, bằng cách nhấn các phím mũi tên lên và xuống, bạn có thể gọi lại các lệnh đã nhập trước đó trong thiết bị đầu cuối. Phím lên xuất hiện lịch sử của các lệnh cuối cùng từ lệnh mới nhất đến lệnh trước đó. Phím xuống hiển thị ngược lại.
- ▶ Điều này thường xuyên và hữu ích nhất trong Linux.
- ▶ Nếu bạn đã thực hiện đúng quy trình, nhấn nút lên một lần sẽ chuyển sang lệnh xóa. Nếu bạn nhấn nó hai lần, ls –a sẽ được hiển thị trong thiết bị đầu cuối.

## Lệnh hệ thống tệp

| man < tên lệnh >



▶ Xuất hướng dẫn về cách sử dụng các lệnh khác nhau. Ví dụ dưới đây cho thấy cách sử dụng lệnh rõ ràng.

- Nhập man và lệnh bạn muốn biết trong ngoặc đơn và chạy.  
`man <lệnh mà bạn muốn biết>`
- Có thể kiểm tra trang Thủ công bằng cách sử dụng các mũi tên Lên, Xuống, Lên trang, Xuống trang và các mũi tên khác trên bàn phím.
- Sau khi xem Hướng dẫn, hãy nhập q nếu bạn muốn dừng.
- Số bên cạnh lệnh ở trên cùng bên trái và bên phải là số phần người đàn ông.

```
$ man ls
```

## Lệnh hữu ích

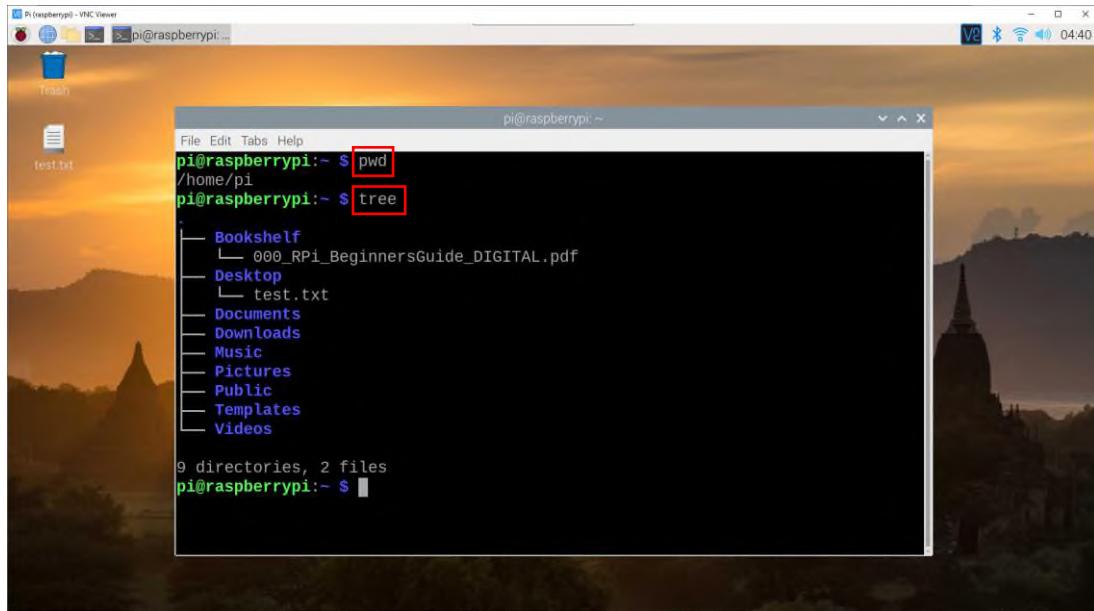
| man <tên lệnh>

Khu vực	Chi tiết chính
<b>man1</b>	Các chương trình thực thi hoặc lệnh shell (ls, cp)
<b>man2</b>	Hệ thống Calls
<b>man3</b>	Thư viện Calls (ex : stdio.h)
<b>man4</b>	Tệp đặc biệt (Quy định liên quan đến thiết bị liên quan, trình điều khiển thiết bị, FIFO, ổ cắm, định dạng trong thư mục/dev)
<b>man5</b>	Định dạng và quy ước tệp (Quy định liên quan đến định dạng tệp như /etc/passwd, /etc/shadow, xinetd.conf)
<b>man6</b>	Games
<b>man7</b>	Miscellanea (Khu vực chứa thông tin liên quan đến tiêu chuẩn liên quan đến tệp hệ thống Linux, quy tắc, giao thức, bộ ký tự và danh sách tín hiệu)
<b>man8</b>	Lệnh quản trị hệ thống (Lệnh liên quan đến quản lý hệ thống được sử dụng bởi root)
<b>man9</b>	Quy trình hạt nhân Linux

- ▶ Trang Thủ công bao gồm 9 phần như trong bảng bên phải.
- ▶ Có thể có các mục có cùng tên nhưng hướng dẫn sử dụng khác nhau trong sách hướng dẫn.
- ▶ Ví dụ: lệnh in có tên printf cũng có thể tồn tại dưới dạng chương trình nhị phân và nếu bạn muốn xem hướng dẫn sử dụng hàm printf là một thư viện, nó thuộc về man3, vì vậy lệnh man phải được sử dụng bằng cách sử dụng nó cùng với man 3 inf.

```
$ man printf
$ man 3 printf
```

## Lệnh hệ thống tệp



### pwd

- ▶ Kiểm tra thư mục hiện đang làm việc.

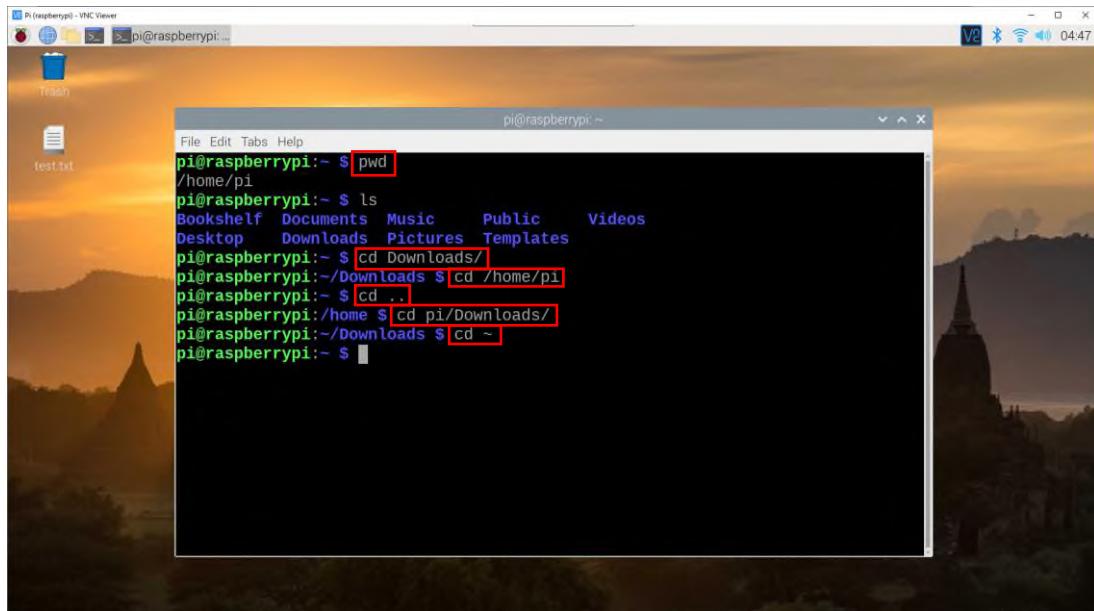
```
$ pwd
```

### tree

- ▶ In và hiển thị cấu trúc của thư mục ở dạng cây.

```
$ tree
```

cd



- ▶ Là viết tắt của thay đổi thư mục, cd là lệnh thay đổi thư mục.
  - cd /dirname : Di chuyển đến “dirname” dựa trên thư mục gốc.
  - cd dirname : Di chuyển đến “dirname” dựa trên thư mục hiện tại.
  - cd ~ : Điều hướng đến thư mục chính.
  - cd ... : Điều hướng lên một cấp độ thư mục.
  - Sử dụng phím TAB để tự động hoàn thành các lệnh, tên tệp hoặc tên thư mục.

```
$ pwd
$ ls
$ cd Downloads
$ cd /home/pi
$ cd ..
$ cd pi/Downloads/
$ cd ~
```

## grep : Tìm kiếm mẫu trong nội dung tệp

grep	
Chức năng	Tìm kiếm các dòng khớp với một mẫu đã chỉ định
Cú pháp	grep [tùy chọn] pattern [file1, file2 ...]
Tùy chọn	<ul style="list-style-type: none"><li>-i      Tìm kiếm không phân biệt chữ hoa và chữ thường</li><li>-l      hiển thị tên của tệp chứa mẫu đã chỉ định</li><li>-n      In số dòng</li></ul>
Ví dụ	grep root /etc/passwd grep -n unix ~/"".txt grep -l hello ".c"

## grep : Tìm kiếm mẫu trong nội dung tệp

|-n option :

The screenshot shows a terminal window titled "Pi (raspberrypi) - VNC Viewer". The terminal content is as follows:

```
pi@raspberrypi:~ $ cp /etc/services data
pi@raspberrypi:~ $ ls
a.txt.cp a.txt.sl b.txt Desktop Downlaods Templates Videos
a.txt.ln Bookshelf data Documents Music
pi@raspberrypi:~ $ grep SSL data
https 443/tcp
nntp 563/tcp ssmtp
ldaps 636/tcp
ftps-data 989/tcp
telnets 992/tcp
imaps 993/tcp
pop3s 995/tcp
suucp 4031/tcp
amqps 5671/tcp
ircs-u 6697/tcp
pi@raspberrypi:~ $ grep -n SSL data
124 https 443/tcp
124 nntp 563/tcp ssmtp
124 ldaps 636/tcp
109 ftps-data 989/tcp
701 telnets 992/tcp
702 imaps 993/tcp
103 pop3s 995/tcp
111 suucp 4031/tcp
361 amqps 5671/tcp
104 ircs-u 6697/tcp
pi@raspberrypi:~ $
```

Red boxes highlight the command "cp /etc/services data", the output of "ls", the command "grep SSL data", and the command "grep -n SSL data". Blue boxes highlight the search term "SSL" in each of the two grep commands.

- ▶ Nhập thông tin chi tiết của /etc/services vào dữ liệu.
- ▶ Trong tệp có tên là dữ liệu, grep tìm thấy mẫu SSH.
- ▶ Tùy chọn -n in tất cả các dòng khớp với một mẫu cụ thể.
- ▶ Các mẫu cụ thể với số dòng khớp với từng mẫu được chỉ định.

## Biểu hiện thông thường

| Biểu thức chính quy là một biểu thức đặc biệt giúp khớp các chuỗi với các mẫu đặc biệt cực kỳ mạnh mẽ.

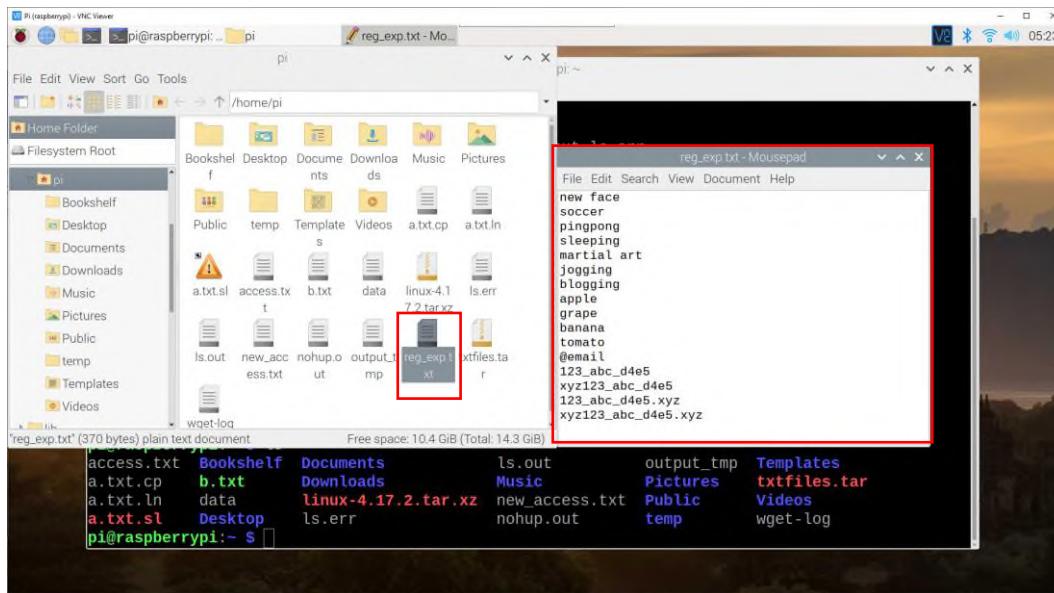
- ▶ Biểu thức chính quy là một nhóm các ký tự đặc biệt giúp tìm kiếm dữ liệu và khớp mẫu phức tạp. Cụm từ thông dụng còn được gọi ngắn gọn là 'regexp' hoặc 'regex'.
- ▶ Biểu thức chính quy có thể được sử dụng không chỉ trong lệnh grep của Linux mà còn trong UNIX, SQL, R, Python và các ngôn ngữ khác.
- ▶ Nếu được sử dụng cùng với greg trong Linux, thì có thể khớp và tìm kiếm mẫu chuỗi thực sự mạnh mẽ.

## Biểu hiện thông thường

Mục	Biểu hiện	Nghĩa
Biểu thức chính quy cơ bản	.	Thay thế bất kỳ ký tự nào bằng số lượng dấu chấm
	^	So khớp vị trí ở đầu chuỗi
	\$	So khớp với vị trí ở cuối chuỗi
	*	Khớp với ký tự phía trước * và khớp khi ký tự liên quan xuất hiện hơn 0 lần
	\	So khớp với ký tự đặc biệt
	()	Nhóm biểu thức chính quy
	?	Khớp với ký tự phía trước? và khớp khi ký tự liên quan xuất hiện chính xác một lần
Biểu thức chính quy ngắt quãng	{n}	So khớp ký tự đứng trước xuất hiện n lần
	{n, m}	So khớp với ký tự đứng trước xuất hiện 'n' lần nhưng không quá m
	{n, }	Chỉ khớp với ký tự trước đó khi nó xuất hiện 'n' lần trở lên
Biểu thức chính quy mở rộng	\+	Khớp với chuỗi trong đó ký tự phía trước \+ xuất hiện nhiều lần
	\?	Khớp với chuỗi có ký tự phía trước \? xuất hiện ít hơn một lần

## Biểu hiện thông thường

### Tìm hiểu các biểu thức thông thường với các ví dụ



- ▶ Tạo một tệp văn bản để kiểm tra các biểu thức chính quy.
- ▶ Tạo một tệp văn bản để kiểm tra các biểu thức chính quy.
- ▶ Tạo một tệp văn bản có tên reg\_exp.txt trong /home/pi và tạo.
- ▶ Chi tiết file đính kèm ở trang sau.

## Ví dụ về biểu thức chính quy

### Chi tiết về tệp ví dụ (Sao chép và Dán)

new face

soccer

pingpong

sleeping

martial art

jogging

blogging

apple

grape

banana

tomato

@email

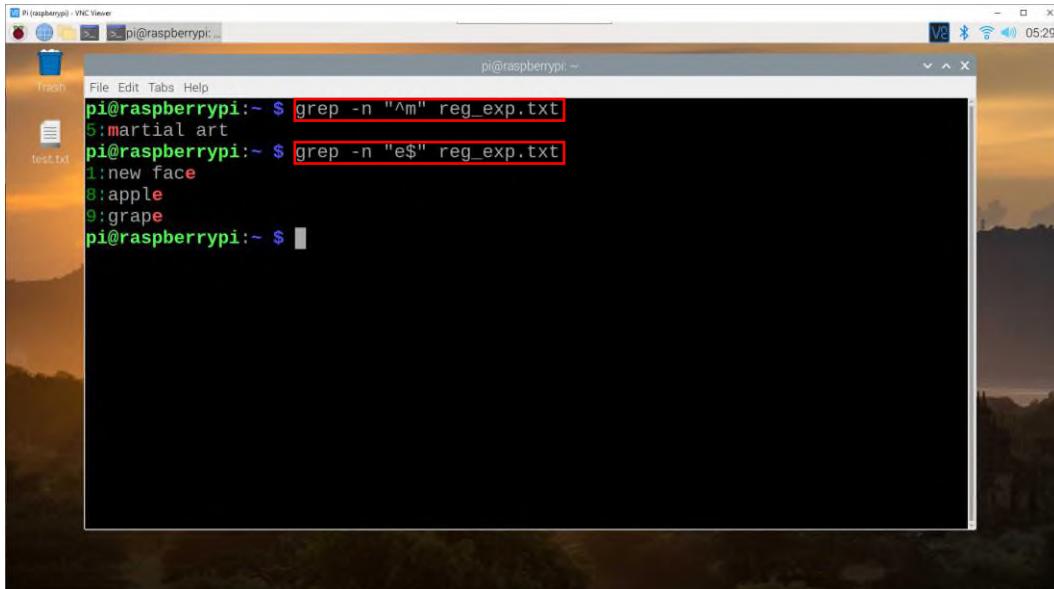
123\_abc\_d4e5

xyz123\_abc\_d4e5

123\_abc\_d4e5.xyz

xyz123\_abc\_d4e5.xyz

## Ví dụ về biểu thức chính quy



```
pi@raspberrypi:~ $ grep -n '^m' reg_exp.txt
5:martial art
pi@raspberrypi:~ $ grep -n 'e$' reg_exp.txt
1:new face
8:apple
9:grape
pi@raspberrypi:~ $
```

grep -n "^m" reg\_exp.txt

- Tìm mẫu trong đó phần đầu của chuỗi là m từ reg\_exp.txt và in cùng với số dòng.

grep -n "e\$" reg\_exp.txt

- Tìm mẫu trong đó phần cuối của chuỗi là e từ reg\_exp.txt và in cùng với số dòng.

Biểu hiện	Nghĩa
^	So khớp vị trí ở đầu chuỗi
\$	So khớp với vị trí ở cuối chuỗi

## Biểu hiện thông thường

```

pi@raspberrypi:~ $ grep -n "m..." reg_exp.txt
5:martial art
11:tomato
12:@email
pi@raspberrypi:~ $ grep -n "..m.." reg_exp.txt
11:tomato
12:@email
pi@raspberrypi:~ $ grep -n "ap*" reg_exp.txt
1:new face
5:martial art
8:apple
9:grape
10:banana
11:tomato
12:@email
13:123_abc_d4e5
14:xyz123_abc_d4e5
15:123_abc_d4e5.xyz
16:xyz123_abc_d4e5.xyz
pi@raspberrypi:~ $

```

Biểu hiện	Nghĩa
.	Thay thế bất kỳ ký tự nào bằng số lượng dấu chấm
*	Khớp với ký tự phía trước * và khớp khi ký tự liên quan xuất hiện hơn 0 lần

### grep -n "m..." reg\_exp.txt

- Tìm chuỗi khớp với bất kỳ ký tự nào bằng số lượng . (3) mà sinh với m từ reg\_exp.txt và in cùng với số dòng.

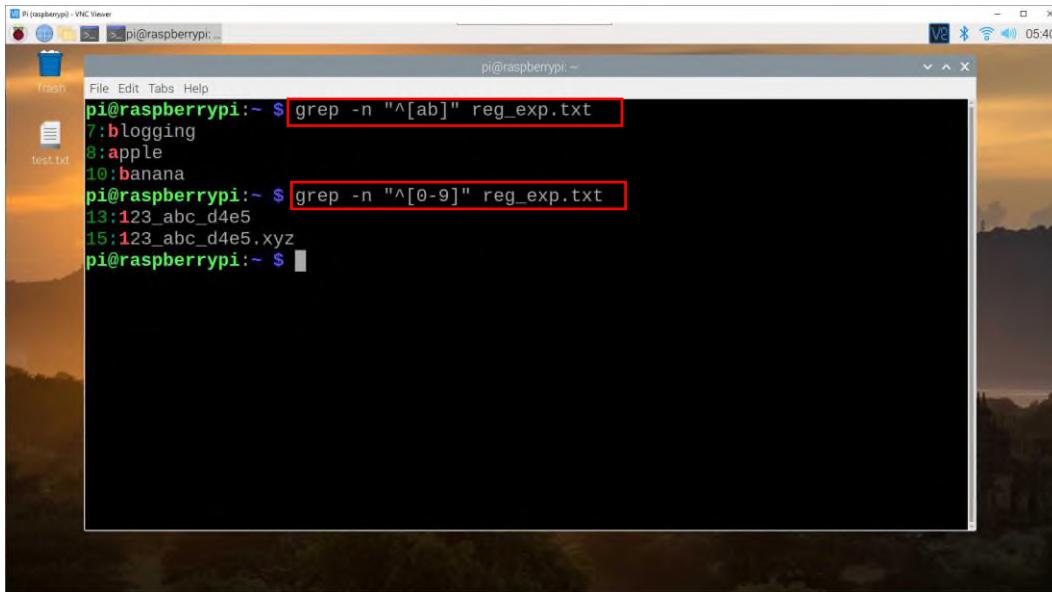
### grep -n "..m.." reg\_exp.txt

- Tìm chuỗi khớp với mẫu bắt đầu bằng 2 ký tự bất kỳ và có m và kết thúc bằng 2 ký tự bất kỳ từ reg\_exp.txt và in cùng với số dòng.

### grep -n "ap\*" reg\_exp.txt

- Tìm chuỗi khớp với mẫu bắt đầu bằng a và có nhiều hơn 0 p từ reg\_exp.txt và in cùng với số dòng.

## Biểu hiện thông thường



```
pi@raspberrypi:~ $ grep -n '^ab' reg_exp.txt
7:blogging
8:apple
10:banana
pi@raspberrypi:~ $ grep -n '^0-9' reg_exp.txt
13:123_abc_d4e5
15:123_abc_d4e5.xyz
pi@raspberrypi:~ $
```

grep -n "^[ab]" reg\_exp.txt

- ▶ Tìm chuỗi khớp với mẫu bắt đầu bằng a hoặc b từ reg\_exp.txt và in cùng với số dòng.

grep -n "^[0-9]" reg\_exp.txt

- ▶ Tìm chuỗi khớp với mẫu bắt đầu bằng các số từ 0 đến 9, nói cách khác, số nguyên từ reg\_exp.txt và in cùng với số dòng.

## Biểu hiện thông thường

grep -n "@" reg\_exp.txt

- Tìm chuỗi khớp với mẫu ký tự đặc biệt @ từ reg\_exp.txt và in cùng với số dòng.

grep -n "." reg\_exp.txt

- Tìm chuỗi khớp với mẫu của một ký tự đặc biệt. Từ reg\_exp.txt và in cùng với số dòng.
- Trong ví dụ trong slide ngay trước đó, . là một siêu ký tự để cập đến một ký tự nhất định nhưng khi được sử dụng cùng với ký tự thoát được gọi là \, nó sẽ mất đi ý nghĩa như một siêu ký tự và được hiểu đơn giản là một ký tự của.

## Biểu hiện thông thường

```

pi@raspberrypi:~ $ grep -E "g"\{1,2\} reg_exp.txt
pingpong
sleeping
jogging
blogging
grape
pi@raspberrypi:~ $ grep "n\+a" reg_exp.txt
banana
pi@raspberrypi:~ $

```

### grep -E "g"\{1,2\} reg\_exp.txt

- Tìm chuỗi khớp với mẫu trong đó g được sử dụng nhiều hơn một lần nhưng ít hơn hai lần từ reg\_exp.txt và in cùng với số dòng.
- Khi sử dụng mẫu sử dụng {} trong grep, -E phải được thêm làm tùy chọn.
- Mẫu được sử dụng ở dạng "Ký tự"\{n,m}

### grep "n\+a" reg\_exp.txt

- Tìm chuỗi khớp với mẫu trong đó ký tự, n phía trước a đã được sử dụng nhiều lần từ reg\_exp.txt và in cùng với số dòng.

Biểu hiện	Nghĩa
\	So khớp với ký tự đặc biệt
{n, m}	So khớp với ký tự đứng trước xuất hiện 'n' lần nhưng không quá m
{n, }	Chỉ khớp với ký tự trước đó khi nó xuất hiện 'n' lần trở lên
\+	Khớp với chuỗi trong đó ký tự phía trước \+ xuất hiện nhiều lần

## find : Tìm kiếm tập tin và thư mục

find	Nghĩa
Chức năng	Tìm kiếm các tệp phù hợp với tiêu chí tại vị trí đã chỉ định
Cú pháp	find [path] [tùy chọn] [expression]
Tùy chọn	<ul style="list-style-type: none"> <li>-name: tên tệp      Tìm kiếm theo tên tệp</li> <li>-type                  Tìm kiếm theo loại tệp</li> <li>-user loginID        Tìm kiếm các tệp mà người dùng được chỉ định sở hữu</li> <li>-perm Quyền        Tìm kiếm các tệp phù hợp với quyền được chỉ định</li> <li>-exec cmd {} \;      Chạy một lệnh trên mỗi tệp được tìm thấy</li> </ul>
Ví dụ	<pre>\$ find . -name hello. \$ find /tmp -user pi0 -exec rm {} \;</pre>

## find : Tìm kiếm tập tin và thư mục

```

Pi (raspberrypi) - VNC Viewer
File Edit Tabs Help
pi@raspberrypi:~ $ ls
a.txt.cp a.txt.sl b.txt Desktop Downloads Pictures Templates
pi@raspberrypi:~ $ find . -name b.txt
./b.txt
pi@raspberrypi:~ $ find ./Desktop -user pi
./Desktop
./Desktop/test.txt
pi@raspberrypi:~ $ 

```

-name options: Tìm kiếm tập tin theo tên.

- Có thể tìm thấy tập hoặc thư mục có tên 'b.txt' trong thư mục hiện tại.

-user options: Tìm kiếm các tập mà người dùng được chỉ định sở hữu.

- Người dùng có thể tìm thấy tập pi hoặc thư mục trong thư mục hiện tại/Vị trí máy tính để bàn.

Tùy chọn	<ul style="list-style-type: none"> <li>-name: tên tập</li> <li>-type</li> <li>-user loginID</li> <li>-perm Quyền</li> <li>-exec cmd {} \;</li> </ul>	<ul style="list-style-type: none"> <li>Tìm kiếm theo tên tập</li> <li>Tìm kiếm theo loại tập</li> <li>Tìm kiếm các tập mà người dùng được chỉ định sở hữu</li> <li>Tìm kiếm các tập phù hợp với quyền được chỉ định</li> <li>Chạy một lệnh trên mỗi tập được tìm thấy</li> </ul>
----------	------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<https://www.geeksforgeeks.org/find-command-in-linux-with-examples/?ref=lbp>

## find : Tìm kiếm tập tin và thư mục

```

pi@raspberrypi: ~ $ find /tmp -user pi -exec rm {} \;
rm: cannot remove '/tmp/ssh-oqkVh7c4d2sn': Is a directory
rm: cannot remove '/tmp/.vnc-1000'
rm: cannot remove '/tmp/.vnc-1000'
rm: cannot remove '/tmp/.vnc-1000'
rm: cannot remove '/tmp/.vnc-1000'
find: '/tmp/.vnc-0': Permission denied
rm: cannot remove '/tmp/ssh-4KZyhFLEtWYa': Is a directory
rm: cannot remove '/tmp/dhcpcd-pi'
find: '/tmp/systemd-private-e5dec764c6534a60a38a4c331854f41f-timesyncd.service-R840
Permission denied
find: '/tmp/pulse-PKhhtXMmr18n': Permission denied
find: '/tmp/systemd-private-e5dec764c6534a60a38a4c331854f41f-timesyncd.service-R840
lg': Permission denied
find: '/tmp/.vnc-vncservice': Permission denied
find: '/tmp/systemd-private-e5dec764c6534a60a38a4c331854f41f-colord.service-pMnksQ': Permis
sion denied
pi@raspberrypi: ~ $ touch /tmp/aaa
pi@raspberrypi: ~ $ find /tmp -user pi -ok rm {} \;
< rm ... /tmp/ssh-oqkVh7c4d2sn > ? n
< rm ... /tmp/.vnc-1000 > ? n
< rm ... /tmp/.vnc-1000/shm > ? n
< rm ... /tmp/.vnc-1000/shm/SysV > ? n
find: '/tmp/.vnc-0': Permission denied
< rm ... /tmp/ssh-4KZyhFLEtWYa > ? n
< rm ... /tmp/dhcpcd-pi > ? n
find: '/tmp/systemd-private-e5dec764c6534a60a38a4c331854f41f-rtkit-daemon.service-D1UiXo':
Permission denied
< rm ... /tmp/aaa > ? y
find: '/tmp/pulse-PKhhtXMmr18n': Permission denied
find: '/tmp/systemd-private-e5dec764c6534a60a38a4c331854f41f-timesyncd.service-R840
lg': Permission denied
find: '/tmp/.vnc-vncservice': Permission denied
find: '/tmp/systemd-private-e5dec764c6534a60a38a4c331854f41f-colord.service-pMnksQ': Permis
sion denied
pi@raspberrypi: ~ $

```

Để thực hiện thao tác tương tự trên tất cả các tệp được tìm thấy bằng lệnh find, hãy sử dụng tùy chọn -exec hoặc -ok.

Để tìm và xóa tất cả các tệp thuộc sở hữu của pi trong thư mục /tmp,

- ▶ Đường dẫn tuyệt đối của các tệp được tìm thấy bởi lệnh find sẽ thay thế { } sau lệnh exec.
- ▶ Phải có khoảng trống giữa lệnh rm và { }, giữa { } và \. \ và ; không nên có khoảng trống ở giữa.

Sử dụng -ok thay vì -exec để xác minh từng tệp được tìm thấy bằng lệnh find trước khi xóa chúng. Khi sử dụng tùy chọn -ok, bạn có thể kiểm tra xem nó có liên tục hỏi bạn có muốn xóa không.

**tar**

`tar cf <file.tar> files`

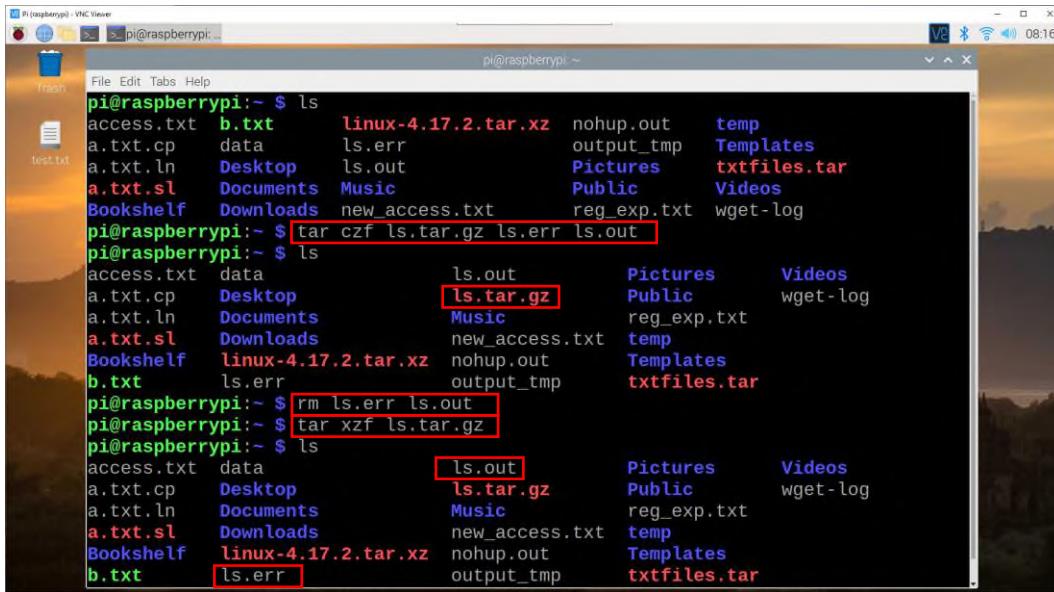
- ▶ Nén các tệp được chỉ định thành tên, file.tar.

```
| tar xf <.tar file>
```

- ▶ Giải nén tệp .tar được chỉ định

<https://www.geeksforgeeks.org/tar-command-linux-examples/?ref=lbp>

## tar



```
pi@raspberrypi:~ $ ls
access.txt  b.txt    linux-4.17.2.tar.xz  nohup.out    temp
a.txt.cp    data      ls.err             output_tmp   Templates
a.txt.ln    Desktop   ls.out            Pictures     txtfiles.tar
a.txt.sl    Documents Music           Public       Videos
Bookshelf   Downloads new_access.txt   reg_exp.txt wget-log
pi@raspberrypi:~ $ tar czf ls.tar.gz ls.err ls.out
pi@raspberrypi:~ $ ls
access.txt  data      ls.out          Pictures     Videos
a.txt.cp    Desktop   ls.tar.gz       Public       wget-log
a.txt.ln    Documents Music          reg_exp.txt
a.txt.sl    Downloads new_access.txt temp
Bookshelf   linux-4.17.2.tar.xz  nohup.out    Templates
b.txt      ls.err    output_tmp     txtfiles.tar
pi@raspberrypi:~ $ rm ls.err ls.out
pi@raspberrypi:~ $ tar xzf ls.tar.gz
pi@raspberrypi:~ $ ls
access.txt  data      ls.out          Pictures     Videos
a.txt.cp    Desktop   ls.tar.gz       Public       wget-log
a.txt.ln    Documents Music          reg_exp.txt
a.txt.sl    Downloads new_access.txt temp
Bookshelf   linux-4.17.2.tar.xz  nohup.out    Templates
b.txt      ls.err    output_tmp     txtfiles.tar
```

### tar czf <file.tar.gz> files

- ▶ Các tệp được chỉ định được nén thành tên, file.tar.gz.
- ▶ Sử dụng phương pháp nén gzip.

### tar xzf <.tar.gz file>

- ▶ Giải nén tệp được chỉ định.tar.gz.

Bài 2.

## Hướng dẫn lệnh hệ điều hành Raspberry Pi

- | 2.1. Kết nối với hệ điều hành Raspberry Pi từ máy tính  
của bạn
- | 2.2. Lệnh cơ bản
- | **2.3. Các lệnh về Hệ thống tệp**
- | 2.4. Các lệnh về điều khiển hệ thống
- | 2.5. Các lệnh về Network
- | 2.6. Các lệnh về Người dùng & Quyền
- | 2.7. Công cụ lập trình

# Các lệnh về Hệ thống tệp

| Chúng ta sẽ học gì trong bài này

- ▶ Tìm hiểu Hệ thống tệp của Linux là gì và nó bao gồm như thế nào.
- ▶ Tìm hiểu và sử dụng các lệnh liên quan đến tệp và thư mục.
- ▶ Tìm hiểu liên kết cứng, liên kết tượng trưng là gì và sử dụng các lệnh liên quan.

### Loại tập tin

- Linux sử dụng một thư mục để quản lý tệp một cách hiệu quả.
- Trong Linux, các tệp được chia thành tệp thông thường, thư mục, liên kết tượng trưng và tệp thiết bị theo mục đích sử dụng của chúng.

Tệp thông  
thường

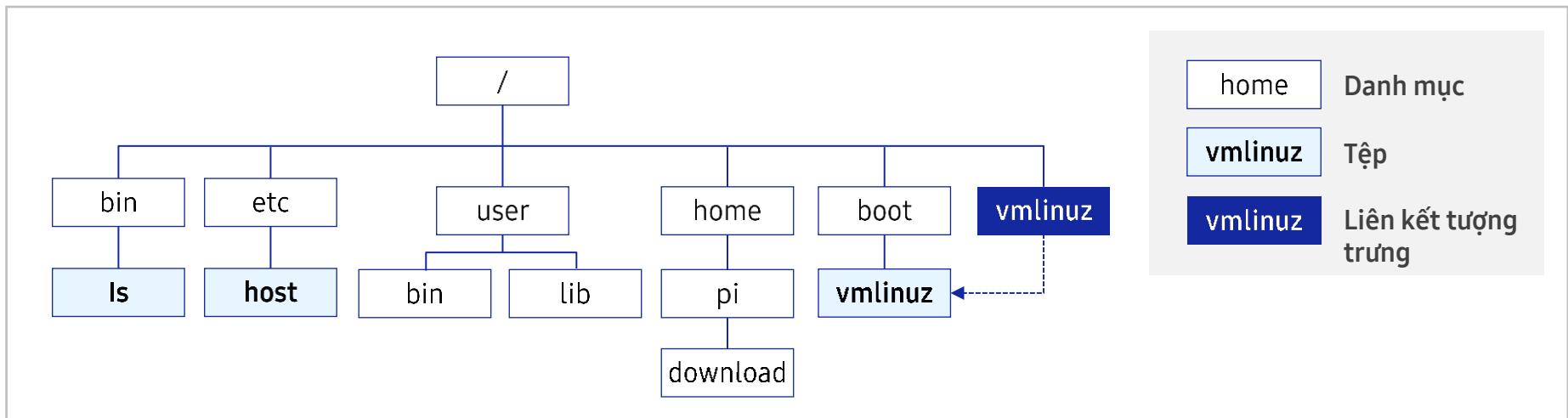
Liên kết tượng  
trưng

Danh mục

Tệp thiết bị

## Làm quen với tên đường dẫn

- Linux tổ chức thư mục theo thứ bậc để quản lý tệp hiệu quả → cấu trúc cây
- Điểm bắt đầu cho tất cả thư mục là thư mục gốc (root), được biểu thị bằng dấu gạch chéo (/)



- Thư mục con: Một thư mục dưới thư mục (bin, etc, usr, home, boot)
- Thư mục mẹ : Hiển thị dưới dạng '..'
- Mọi thư mục đều có thư mục mẹ trừ thư mục gốc

### Thư mục làm việc

- | Thư mục đang sử dụng được gọi là thư mục làm việc hay thư mục hiện tại
- | Thư mục hiện tại được đánh dấu bằng dấu ‘.’
- | Kiểm tra vị trí thư mục hiện tại bằng lệnh pwd

### Thư mục chính

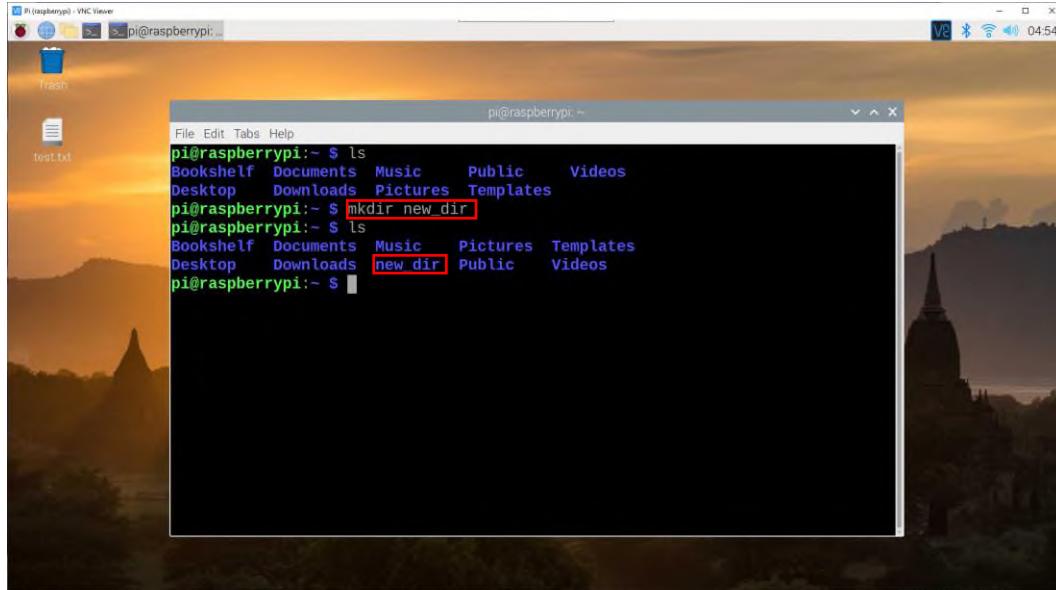
- | Chỉ định khi tạo tài khoản người dùng lần đầu với thư mục được chỉ định cho mỗi người dùng
- | Người dùng có thể làm việc với các tệp hoặc thư mục con trong thư mục chính của họ
- | Thư mục chính được biểu thị bằng ký hiệu ‘~’ -> ~pi

## Mô tả thư mục chính

Thư mục	Kiểu tệp tin
<b>dev</b>	Thư mục chứa tệp thiết bị.
<b>home</b>	Vị trí tạo thư mục chính người dùng.
<b>media</b>	Thư mục gán thiết bị bên ngoài như CD-ROM và USB.
<b>opt</b>	Vị trí cài đặt gói bổ sung.
<b>root</b>	Thư mục chính của tài khoản gốc. Được nhầm lẫn vì thư mục này khác với thư mục gốc (/).
<b>sys</b>	Thư mục này chứa tệp tin liên quan đến Linux kernel.
<b>usr</b>	Có rất nhiều tệp tin, bao gồm tệp thực thi chính, tệp thư viện và tệp tiêu đề. Để tham khảo, usr là viết tắt của Unix System Resource.
<b>bin</b>	Có một tệp thực thi (lệnh).

### mkdir

mkdir <tên thư mục>



- ▶ Tạo một thư mục trống dưới tên. <tên thư mục>
- ▶ So sánh trước và sau lệnh mkdir bằng lệnh ls.

```
$ ls
$ mkdir new_dir
$ ls
```

## Các lệnh với thư mục

- █ `mkdir <Tên thư mục>` (Tạo thư mục)
- █ `rmdir <Tên thư mục>` (Xóa thư mục)

```
pi@raspberrypi:~ $ mkdir example1
pi@raspberrypi:~ $ mkdir example1/child
pi@raspberrypi:~ $ tree
.
├── Bookshelf
│   └── 000_RPi_BeginnersGuide_DIGITAL.pdf
├── Desktop
│   └── test.txt
├── Documents
├── Downloads
└── example1
    └── child
.
├── Bookshelf
│   └── 000_RPi_BeginnersGuide_DIGITAL.pdf
├── Desktop
│   └── test.txt
├── Documents
├── Downloads
└── example1
.
└── test.txt

11 directories, 2 files
pi@raspberrypi:~ $ rmdir -p example1/child
pi@raspberrypi:~ $ tree
.
├── Bookshelf
│   └── 000_RPi_BeginnersGuide_DIGITAL.pdf
├── Desktop
│   └── test.txt
├── Documents
├── Downloads
└── example1
.
└── test.txt

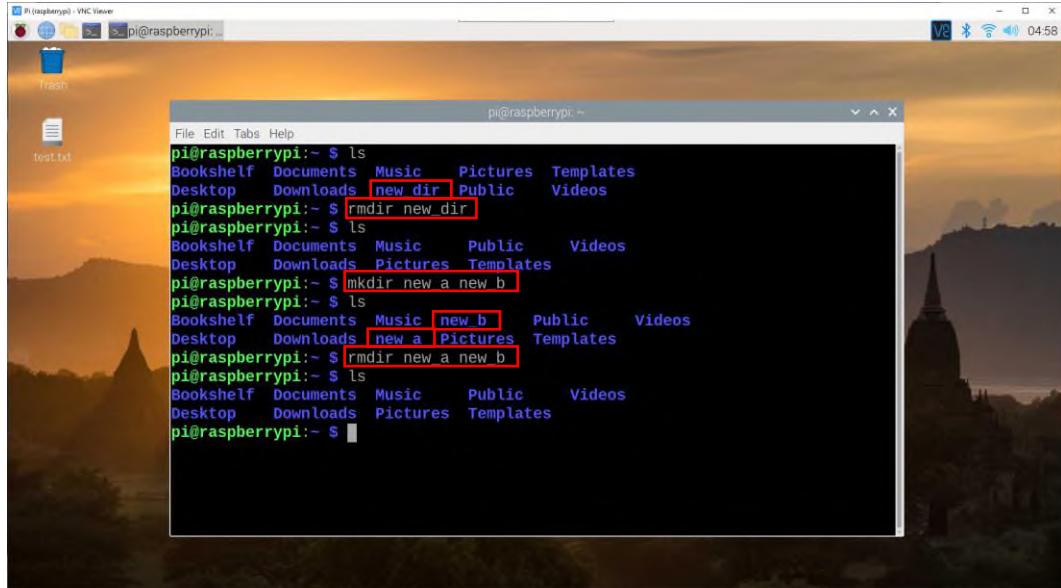
9 directories, 2 files
pi@raspberrypi:~ $
```

- ▶ Nếu option `-p` được sử dụng, thư mục cấp trên của `<Tên thư mục>` sẽ bị xóa cùng nhau.

```
$ mkdir example1
$ mkdir example1/child
$ tree
$ rmdir -p example1/child
$ tree
```

## rmkdir

### rmkdir <Tên thư mục>



```
pi@raspberrypi:~ $ ls
Bookshelf Documents Music Pictures Templates
Desktop Downloads new_dir Public Videos
pi@raspberrypi:~ $ rmdir new_dir
pi@raspberrypi:~ $ ls
Bookshelf Documents Music Public Videos
Desktop Downloads Pictures Templates
pi@raspberrypi:~ $ mkdir new_a new_b
pi@raspberrypi:~ $ ls
Bookshelf Documents Music new_b Public Videos
Desktop Downloads new_a Pictures Templates
pi@raspberrypi:~ $ rmdir new_a new_b
pi@raspberrypi:~ $ ls
Bookshelf Documents Music Public Videos
Desktop Downloads Pictures Templates
pi@raspberrypi:~ $
```

- ▶ Sử dụng lệnh rmkdir để xóa thư mục <tên thư mục>. Nó không bị xóa nếu một tệp tồn tại trong thư mục sẽ bị xóa.
- ▶ Nếu một số đối số được thêm vào nó, thì có thể đồng thời tạo và xóa một số thư mục.

```
$ ls
$ rmkdir new_dir
$ ls
$ mkdir new_a new_b
$ ls
$ rmkdir new_a new_b
$ ls
```

## Touch: tạo tệp (file) trong linux

| touch <option> <tên tệp>

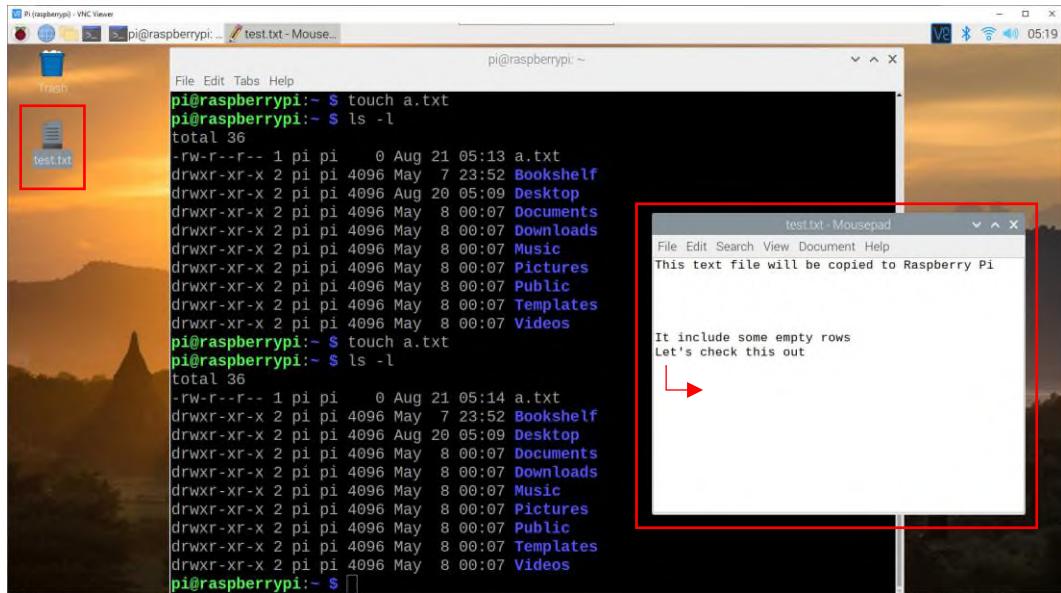
```
pi@raspberrypi:~$ touch a.txt
pi@raspberrypi:~$ ls -l
total 36
-rw-r--r-- 1 pi pi    0 Aug 21 05:13 a.txt
drwxr-xr-x 2 pi pi 4096 May  7 23:52 Bookshelf
drwxr-xr-x 2 pi pi 4096 Aug 20 05:09 Desktop
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Documents
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Downloads
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Music
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Pictures
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Public
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Templates
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Videos
pi@raspberrypi:~$ touch a.txt
pi@raspberrypi:~$ ls -l
total 36
-rw-r--r-- 1 pi pi    0 Aug 21 05:14 a.txt
drwxr-xr-x 2 pi pi 4096 May  7 23:52 Bookshelf
drwxr-xr-x 2 pi pi 4096 Aug 20 05:09 Desktop
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Documents
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Downloads
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Music
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Pictures
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Public
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Templates
drwxr-xr-x 2 pi pi 4096 May  8 00:07 Videos
pi@raspberrypi:~$
```

- ▶ Nếu các định dạng như touch <tên tệp> được sử dụng, nó sẽ tạo một tệp mới dưới tên <tên tệp>.
- ▶ Nếu một tệp đã tồn tại, thời gian của tệp liên quan sẽ được thay đổi thành hiện tại.
- ▶ lsKiểm tra xem ngày chỉnh sửa đã được thay đổi chưa.

```
$ touch a.txt
$ ls -l
$ touch a.txt
$ ls -l
```

## Lệnh “cat”: xem nội dung của một tệp/file

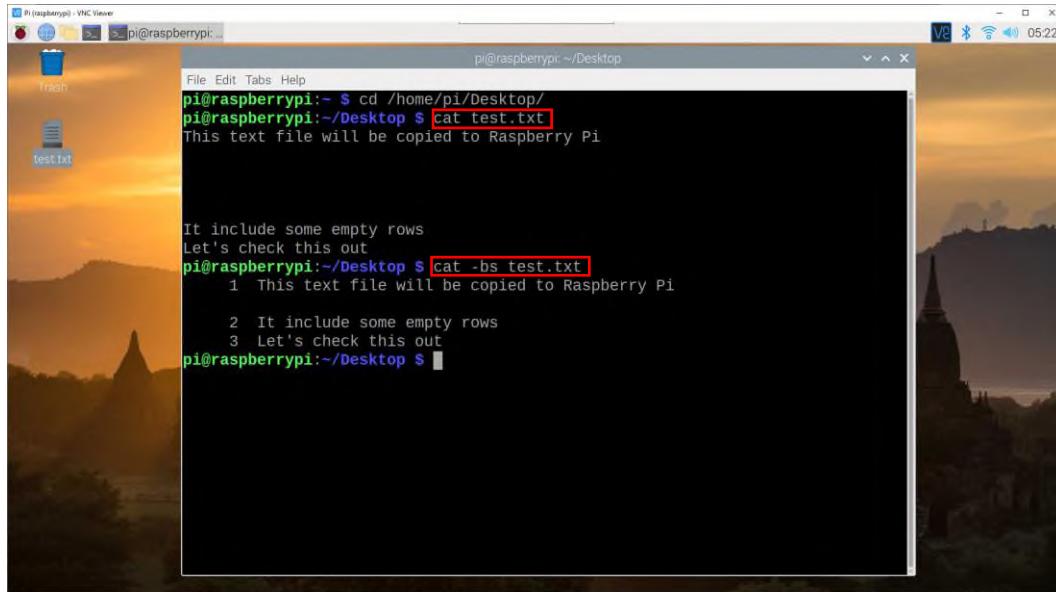
| cat <option> <tên tệp>



- ▶ Các chi tiết của tập tin được hiển thị trong màn hình.
- ▶ Để kiểm tra lệnh, hãy chỉnh sửa test.txt trong màn hình và lưu lại.
- ▶ Vì bất tiện khi kiểm tra đầu ra, hãy sử dụng Enter ở cuối câu cuối cùng để thêm một dòng trống khác và lưu lại.

## Lệnh “cat”: xem nội dung tệp/file

| cat <option> <tên tệp>



```
pi@raspberrypi:~ $ cd /home/pi/Desktop/
pi@raspberrypi:~/Desktop $ cat test.txt
This text file will be copied to Raspberry Pi

It include some empty rows
Let's check this out
pi@raspberrypi:~/Desktop $ cat -bs test.txt
1 This text file will be copied to Raspberry Pi

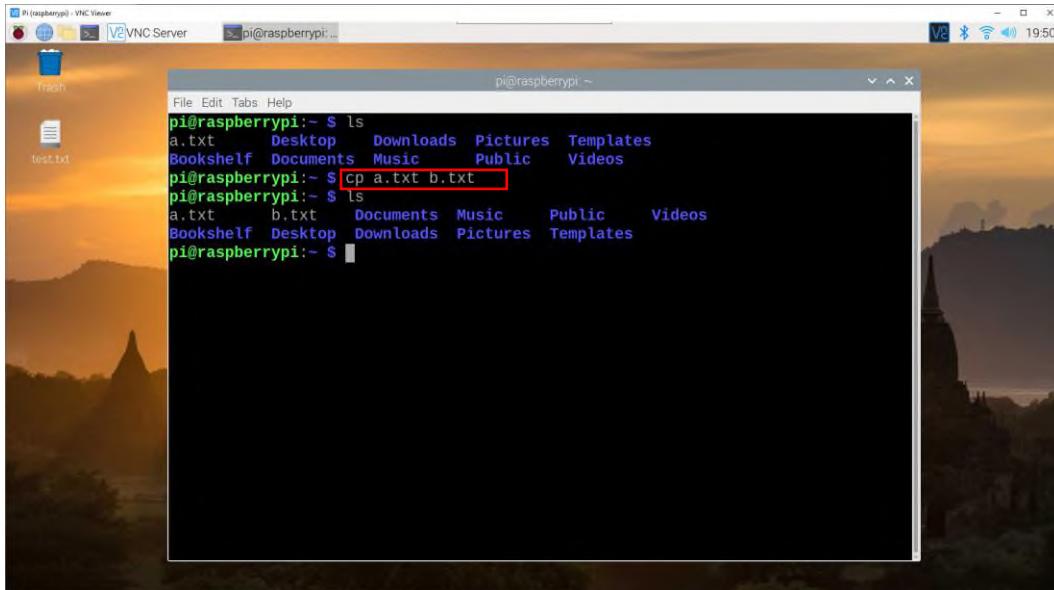
2 It include some empty rows
3 Let's check this out
pi@raspberrypi:~/Desktop $
```

- ▶ Các tùy chọn được đề xuất là –b, -s
- ▶ -b: Thêm số dòng trừ dòng trống rồi in.
- ▶ -s: Hợp nhất dòng trống liên tục thành một rồi in.
- ▶ Có thể sử dụng một số tùy chọn cùng một lúc, giống như ví dụ:

```
$ cd /home/pi/Desktop/
$ cat test.txt
$ cat -bs test.txt
```

### Lệnh “cp”: sao chép (copy) tệp/file

| cp <tên tệp cần copy> <tên tệp copy và tạo mới>



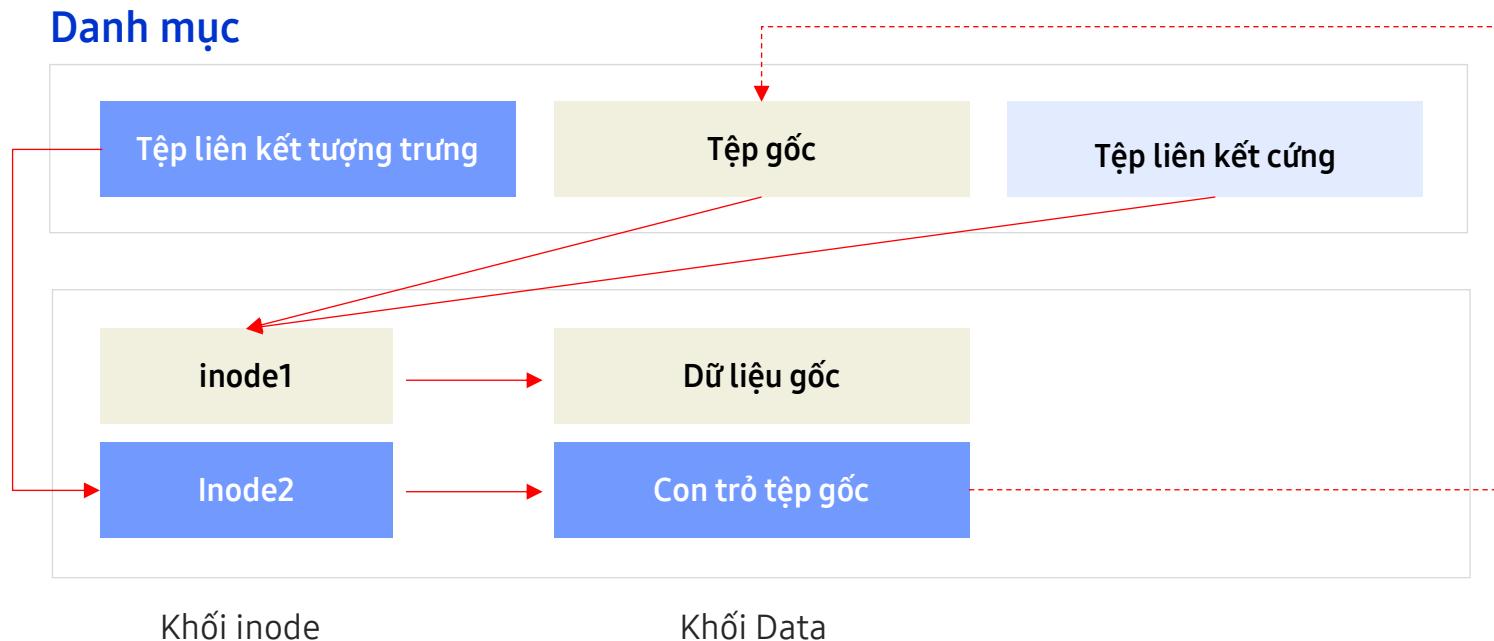
```
pi@raspberrypi:~$ ls
a.txt  Desktop  Downloads  Pictures  Templates
Bookshelf  Documents  Music  Public  Videos
pi@raspberrypi:~$ cp a.txt b.txt
pi@raspberrypi:~$ ls
a.txt  b.txt  Documents  Music  Public  Videos
Bookshelf  Desktop  Downloads  Pictures  Templates
pi@raspberrypi:~$
```

- ▶ Lệnh cp là viết tắt của copy.
- ▶ Tập tin có thể được sao chép.

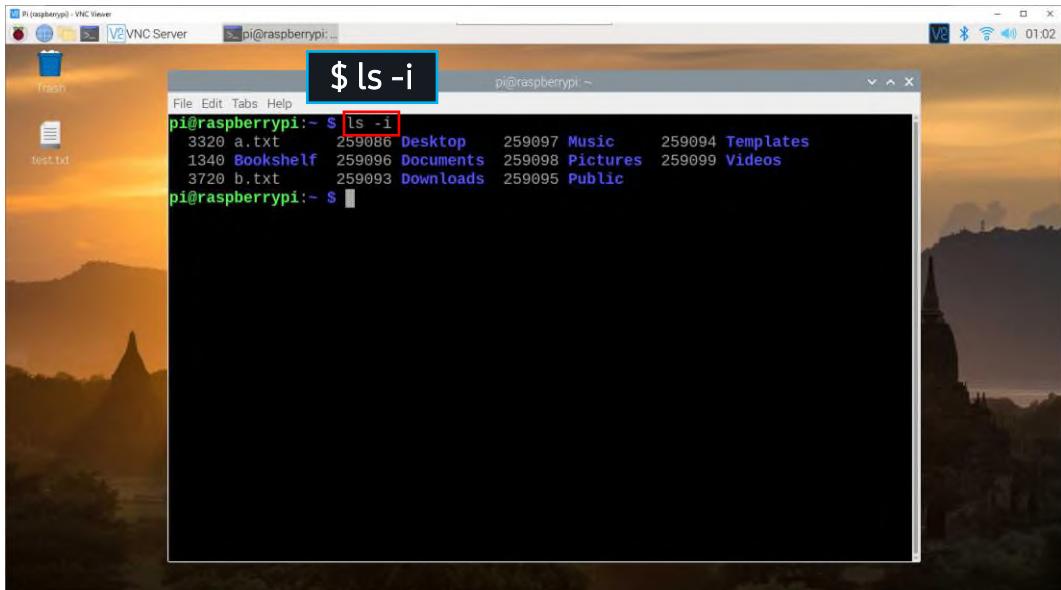
```
$ ls
$ cp a.txt b.txt
$ ls
```

## liên kết tập tin

- Liên kết tệp cung cấp cho tệp hiện có một tên khác (mới) trong cấu trúc thư mục
- Liên kết cứng: Tạo một tệp mới sử dụng cùng một dữ liệu nhưng có tên khác
- Liên kết tương trưng: Tạo một tệp mới trỏ đến tệp gốc



## Các thành phần của tệp Linux



Tạo mới file: \$ touch a.txt

tệp = tên tệp + inode + khối data

- ▶ Tên tệp: Tên tệp mà người dùng sử dụng để truy cập tệp.
- ▶ inode: Cấu trúc dữ liệu chứa thông tin về tệp. Hiển thị bên ngoài dưới dạng số, lưu trữ nội bộ loại tệp, quyền, chủ sở hữu, kích thước, địa chỉ khối dữ liệu, v.v.

ls -i hiển thị số inode

- ▶ Số trước tên tệp là số inode.
- ▶ Tên tệp khác nhau nhưng cùng số inode có nghĩa là cùng một tệp.
- ▶ Ví dụ sử dụng lệnh ls -i để kiểm tra số inode. Ví dụ) Số inode của a.txt → 3320

## In : Liên kết cứng

### Liên kết cứng

- ▶ Liên kết cứng là một tệp sử dụng cùng một inode với tệp gốc, làm tăng số lượng liên kết trong tệp gốc được sao chép.
- ▶ Khi bạn chỉnh sửa một tệp, các chỉnh sửa của bạn được áp dụng giống như một liên kết tương trưng. Khi bạn chỉnh sửa một liên kết cứng, bản gốc cũng được sửa đổi.
- ▶ Nếu bạn xóa tệp gốc, liên kết cứng sẽ gửi dữ liệu qua inode đến địa chỉ nơi tệp gốc được lưu trữ và nội dung của tệp gốc sẽ vẫn còn trong tệp liên kết cứng.
- ▶ **Vì liên kết cứng được tạo dưới dạng một tệp chứa thông tin vị trí của dữ liệu gốc nên để xóa tệp gốc, tất cả các liên kết cứng đã tạo phải được xóa.**

## In : Liên kết cứng

```

pi@raspberrypi:~ ls -l
total 36
-rw-r--r-- 1 pi pi 0 Aug 21 05:14 a.txt
drwxr-xr-x 2 pi pi 4096 May 7 23:52 Bookshelf
-rw-r--r-- 1 pi pi 0 Aug 21 19:50 b.txt
drwxr-xr-x 2 pi pi 4096 Aug 20 05:09 Desktop
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Documents
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Downloads
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Music
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Pictures
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Public
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Templates
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Videos
pi@raspberrypi:~ ln a.txt a.txt.ln
pi@raspberrypi:~ ls -l
total 36
-rw-r--r-- 2 pi pi 0 Aug 21 05:14 a.txt
-rw-r--r-- 2 pi pi 0 Aug 21 05:14 a.txt.ln
drwxr-xr-x 2 pi pi 4096 May 7 23:52 Bookshelf
-rw-r--r-- 1 pi pi 0 Aug 21 19:50 b.txt
drwxr-xr-x 2 pi pi 4096 Aug 20 05:09 Desktop
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Documents
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Downloads
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Music
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Pictures
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Public
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Templates
drwxr-xr-x 2 pi pi 4096 May 8 00:07 Videos
pi@raspberrypi:~ 

```

**In a.txt a.txt.ln**

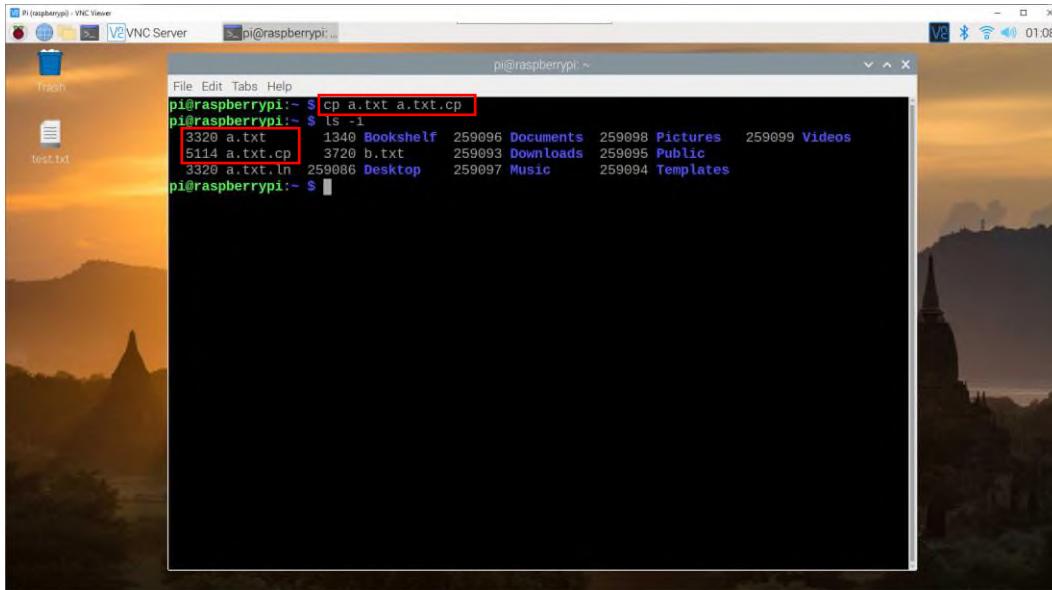
Number of links in a.txt file increases from 1 to 2

In	
Chức năng	Tạo liên kết tập tin
Chức năng	ln [tùy chọn] source_tập target_tập
Lựa chọn	-s Tạo tệp liên kết tương trưng
Ví dụ	ln test lntest ln -s test lntest

Sử dụng lệnh: ls -i để xác định inode của các tệp?

## In : Liên kết cứng

### Sự khác biệt giữa liên kết và sao chép



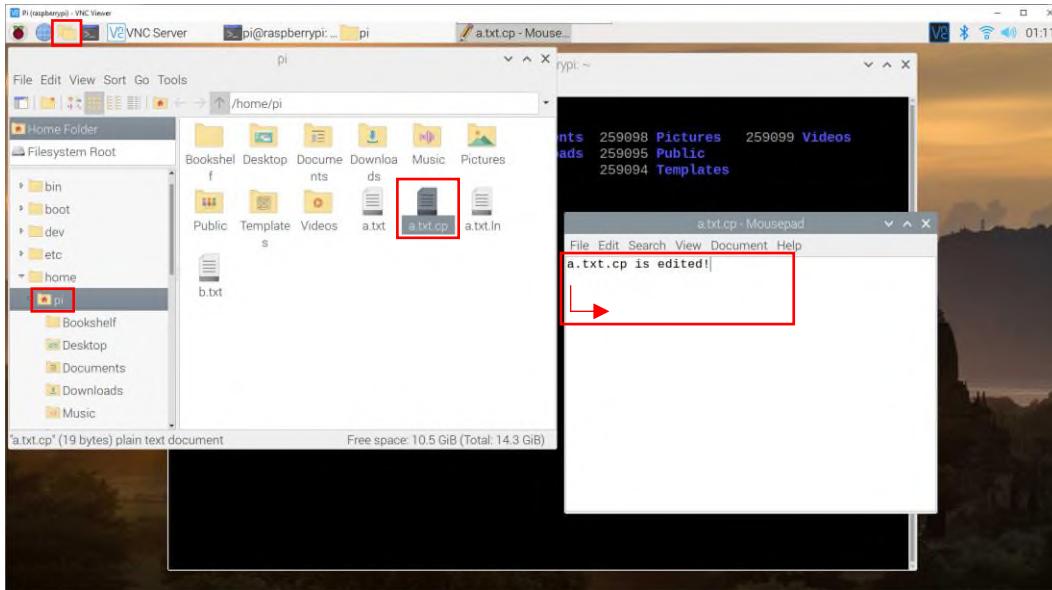
```
pi@raspberrypi:~ $ cp a.txt a.txt.cp
pi@raspberrypi:~ $ ls -l
3320 a.txt      1340 Bookshelf  259096 Documents  259098 Pictures   259099 Videos
5114 a.txt.cp    3720 b.txt     259093 Downloads  259095 Public
3320 a.txt.ln  259086 Desktop   259097 Music     259094 Templates
pi@raspberrypi:~ $
```

- ▶ Các sửa đổi đối với tệp data1.cp không được phản ánh trong tệp data1.
- ▶ Nếu a.txt.ln được chỉnh sửa, kết quả của việc sửa đổi sẽ được phản ánh trong a.txt.
- ▶ Đầu tiên, tạo a.txt.cp bằng cách sao chép a.txt.

```
$ cp a.txt a.txt.cp
```

## In : Liên kết cứng

### Sự khác biệt giữa liên kết và sao chép

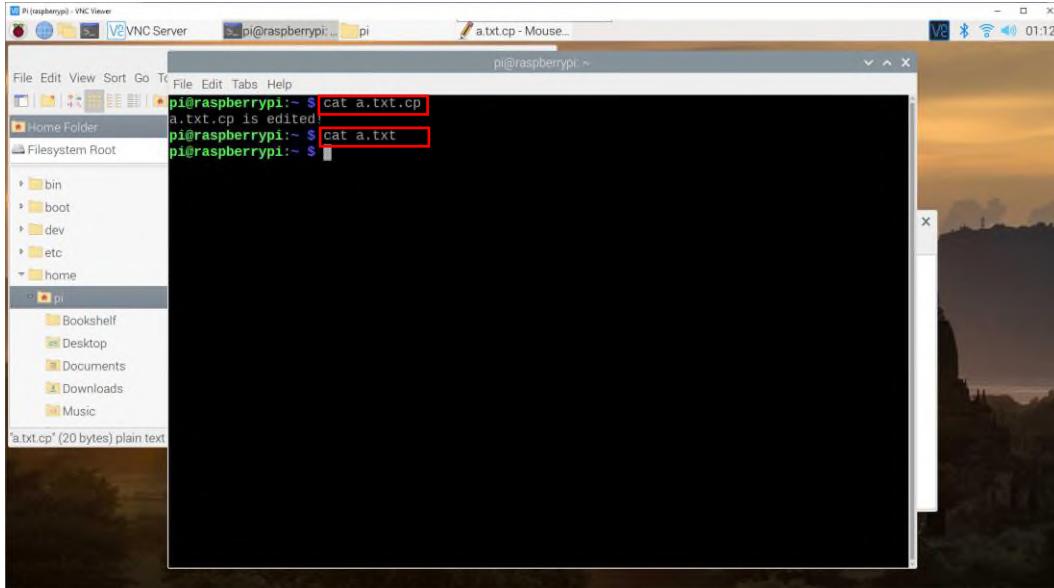


- Trình quản lý tệp ở trên cùng bên trái → a.txt.cp → Chỉnh sửa thông tin chi tiết và lưu lại. (Thêm một dòng trống bằng cách sử dụng Enter.)

Chỉnh sửa nội dung của file:  
**\$ nano a.txt.cp**

### In : Liên kết cứng

#### Sự khác biệt giữa liên kết và sao chép

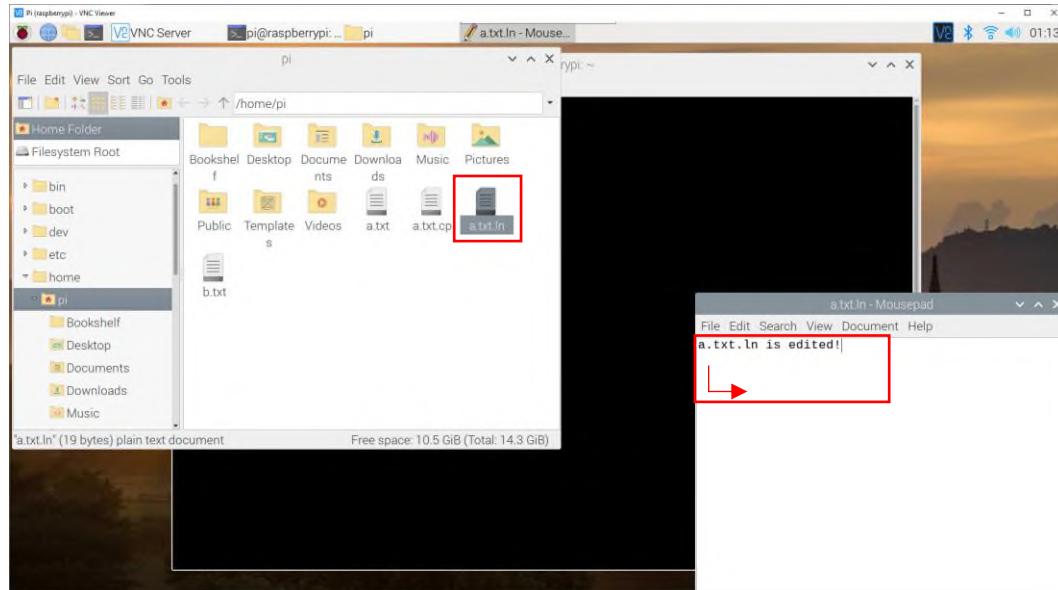


- Mặc dù `a.txt.cp` đã được chỉnh sửa nhưng bạn có thể thấy rằng không có thay đổi nào trong `a.txt`.

```
$ cat a.txt.cp  
$ cat a.txt
```

## In : Liên kết cứng

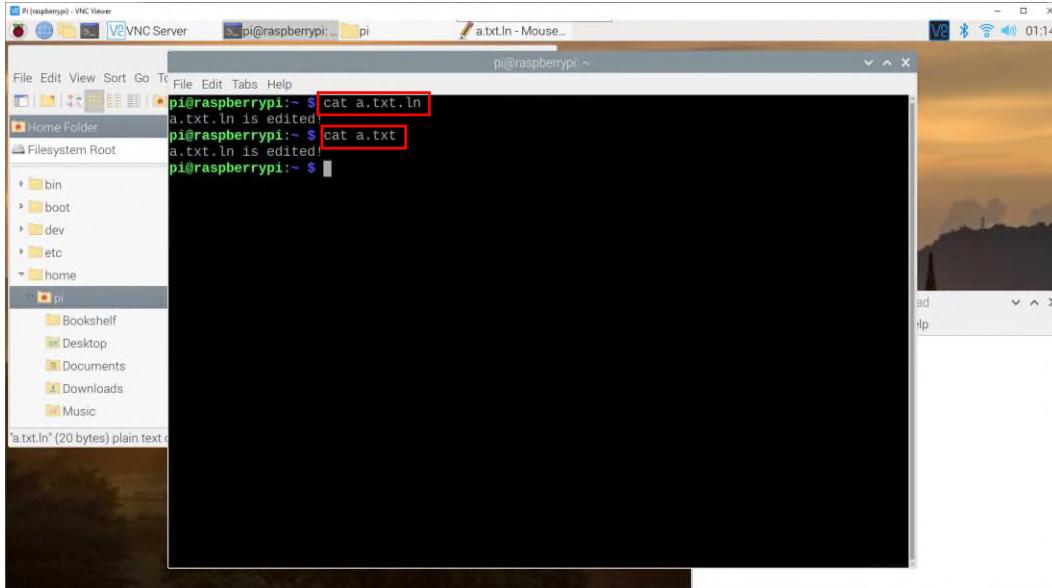
### Sự khác biệt giữa liên kết và sao chép



- Mở a.txt.ln, chỉnh sửa chi tiết và lưu lại. (Ngoài ra, thêm một dòng trống ở cuối bằng cách sử dụng Enter.)

## In : Liên kết cứng

### Sự khác biệt giữa liên kết và sao chép



- Đối với việc sửa đổi a.text.on, bạn có thể thấy a.txt đã bị ảnh hưởng và được lưu lại.

```
$ cat a.txt.ln  
$ cat a.txt
```

## In -s : Tạo liên kết tượng trưng

```

pi@raspberrypi:~ $ ls
a.txt  a.txt.ln  b.txt  Documents  Music  Public  Videos
a.txt.cp  Bookshef Desktop  Pictures  Templates
pi@raspberrypi:~ $ ln -s a.txt a.txt.sl
pi@raspberrypi:~ $ ls -i
3320 a.txt  5125 a.txt.sl  259086 Desktop  259097
5114 a.txt.cp  1349 Bookshef  259096 Documents  259098
3320 a.txt.ln  3720 b.txt  259093 Downloads  259095
pi@raspberrypi:~ $ ls -l a.txt.sl
lrwxrwxrwx 1 pi pi 5 Aug 22 01:17 a.txt.sl -> a.txt
pi@raspberrypi:~ $

```

The terminal window shows the creation of a symbolic link named 'a.txt.sl' pointing to the file 'a.txt'. The command used was 'ln -s a.txt a.txt.sl'. The output of 'ls -i' shows that 'a.txt' has an inode of 5125, while its symbolic link 'a.txt.sl' has an inode of 259097. The command 'ls -l a.txt.sl' shows the symbolic link's details, indicating it points to 'a.txt'.

- ▀ Số inode của liên kết tượng trưng khác với số của tệp gốc
- ▀ Lệnh ls -l hiển thị loại tệp là l (chữ thường l) và cho bạn biết tệp nào là tệp gốc bằng cách sử dụng '-' trong tên tệp
  - ▶ Mặc dù tệp liên kết tượng trưng có tên là 'a.txt.sl' đã được tạo, nhưng bạn có thể thấy rằng inode khác nhau đối với 'a.txt' và 'a.txt.sl'.
  - ▶ Nếu bạn kiểm tra 'a.txt.sl', bạn có thể thấy rằng nó trỏ đến 'a.txt'.

### Sự khác biệt giữa Liên kết tượng trưng và Liên kết cứng

| Liên kết tượng trưng có một số khác biệt so với liên kết cứng như sau:

- ▶ Loại tệp được chỉ định bởi l. (chữ thường l)
- ▶ Chỉ có một liên kết cứng vì không có tên nào được thêm vào tệp gốc.
- ▶ Tên tệp gốc được hiển thị sau tên tệp. (-> a.txt)
- ▶ Số inode khác với tệp gốc vì tệp gốc và tệp liên kết tượng trưng là các tệp riêng biệt.
- ▶ Liên kết tượng trưng có thể được tạo trong một thư mục.
- ▶ Liên kết tượng trưng có thể được tạo ngay cả khi các hệ thống tệp khác nhau.

## Sự khác biệt giữa Liên kết tượng trưng và Liên kết cứng

```

pi@raspberrypi:~ $ ls
a.txt  a.txt.ln  b.txt  Documents  Music  Public  Videos
a.txt.cp  Bookshelf  Desktop  Downloads  Pictures  Templates
pi@raspberrypi:~ $ ln -s a.txt a.txt.sl
pi@raspberrypi:~ $ ls -i
3320 a.txt  5125 a.txt.sl  259086 Desktop  259097 Music  259094 Templates
5114 a.txt.cp  1340 Bookshelf  259096 Documents  259098 Pictures  259099 Videos
3320 a.txt.ln  3720 b.txt  259093 Downloads  259095 Public
pi@raspberrypi:~ $ ls -l a.txt
lrwxrwxrwx 1 pi pi 5 Aug 22 0
pi@raspberrypi:~ $ rm a.txt
pi@raspberrypi:~ $ ls -i
5114 a.txt.cp  1340 Books  259098 Pictures  259099 Videos
3320 a.txt.ln  3720 b.txt  259093 Downloads  259095 Public
5125 a.txt.sl  259086 Desktop
pi@raspberrypi:~ $ cat a.txt.sl
cat: a.txt.sl: No such file or directory
pi@raspberrypi:~ $

```

Nội dung của tệp liên kết tượng trưng là đường dẫn của tệp gốc

Khi tệp gốc bị xóa khỏi liên kết tượng trưng, không thể liên kết tượng trưng

- ▶ Nếu bạn xóa a.txt và kiểm tra chi tiết của thư mục chính, màu văn bản của a.txt.sl đã thay đổi. Nó không thể tìm thấy tệp mà nó đang trỏ tới.
- ▶ Ngay cả khi cat được áp dụng, a.txt.sl không thể in được.
- ▶ Nội dung file a.txt vẫn được gắn với liên kết cứng a.txt.ln (kiểm tra thông qua lệnh cat)

Bài 2.

## Hướng dẫn lệnh hệ điều hành Raspberry Pi

- | 2.1. Kết nối với hệ điều hành Raspberry Pi từ máy tính  
của bạn
- | 2.2. Lệnh cơ bản
- | 2.3. Các lệnh về Hệ thống tệp
- | **2.4. Các lệnh về điều khiển hệ thống**
- | 2.5. Các lệnh về Network
- | 2.6. Các lệnh về Người dùng & Quyền
- | 2.7. Công cụ lập trình

# Các lệnh về điều khiển hệ thống

| Chúng ta sẽ học gì trong bài này

- ▶ Tìm hiểu những lệnh nào kiểm soát Hệ thống trong Linux.
- ▶ Tìm hiểu khái niệm cơ bản về quy trình và điều khiển quy trình bằng các lệnh.
- ▶ Tìm hiểu các lệnh để xem thông tin ngắn gọn liên quan đến nền trước, công việc nền và hệ thống.

### Khái niệm về quy trình

- | Process: Một chương trình hiện đang chạy trên hệ thống
- | Mối quan hệ cha-con của quá trình
  - ▶ Các tiến trình có mối quan hệ cha-con.
  - ▶ Nếu cần, tiến trình cha có thể tạo các tiến trình con và các tiến trình con có thể tạo các tiến trình con khác.
  - ▶ Tất cả các quy trình đều có quy trình mẹ ngoại trừ các quy trình systemd và kthreadd được chạy bởi bộ lập lịch trình khi khởi động.
  - ▶ Khi tiến trình con hoàn thành nhiệm vụ của nó, nó sẽ trả kết quả về tiến trình cha và kết thúc.

### Loại quy trình

#### Quy trình daemon

- ▶ **Một quá trình nền chạy dài để đáp ứng các yêu cầu từ các dịch vụ.**

#### Quy trình Orphan

- ▶ **Khi một tiến trình cha kết thúc trước trong khi tiến trình đó vẫn đang chạy.**

#### Quy trình Zombie

- ▶ Khi một quy trình con đã kết thúc thực thi nhưng vẫn còn trong danh sách bảng quy trình.
- ▶ Quy trình Zombie có thể xuất hiện dưới dạng 'Quy trình không còn tồn tại' trong danh sách quy trình.
- ▶ Nếu quy trình zombie tăng lên, bảng quy trình có thể hết dung lượng. Nó có thể ngăn các quy trình thông thường chạy.

## Danh sách quy trình

### I Lệnh để xem danh sách các quy trình hiện đang chạy: ps

- ▶ Lựa chọn UNIX (SVR4): Đặt thành một gói. Bắt đầu bằng dấu gạch nối. (Ví dụ: -ef)
- ▶ Lựa chọn BSD: Được đóng gói. Không bắt đầu bằng dấu gạch ngang. (Ví dụ: phụ trợ)
- ▶ Lựa chọn GNU: Bắt đầu bằng hai dấu gạch ngang. (Ví dụ: --pid)
- ▶ Một tiến trình cha kết thúc trước trong khi tiến trình vẫn đang chạy.

### ps

- Danh sách tiến trình của thiết bị đầu cuối hiện tại

### ps -f

- Danh sách tiến trình của thiết bị đầu cuối hiện tại
  - In thông tin chi tiết về tiến trình xử lý: PPID, số thiết bị đầu cuối, thời gian bắt đầu, v.v.

```
pi@raspberrypi:~$ ps
  PID TTY      TIME CMD
 964 pts/0    00:00:00 bash
1535 pts/0    00:00:00 ps
pi@raspberrypi:~$ ps -f
UID        PID  PPID C STT   TIME CMD
pi        964  950  0 Aug22 pts/0    00:00:00 bash
pi       1536  964  0 02:22 pts/0    00:00:00 ps -f
pi@raspberrypi:~$
```

### ps -a

#### In thông tin tiến trình xử lý trên thiết bị đầu cuối

- ▶ In thông tin về quy trình được thực hiện trên tất cả các thiết bị đầu cuối và quy trình được liên kết với người dùng hiện tại.

### ps -e

#### In một danh sách đầy đủ của tiến trình xử lý (tùy chọn UNIX)

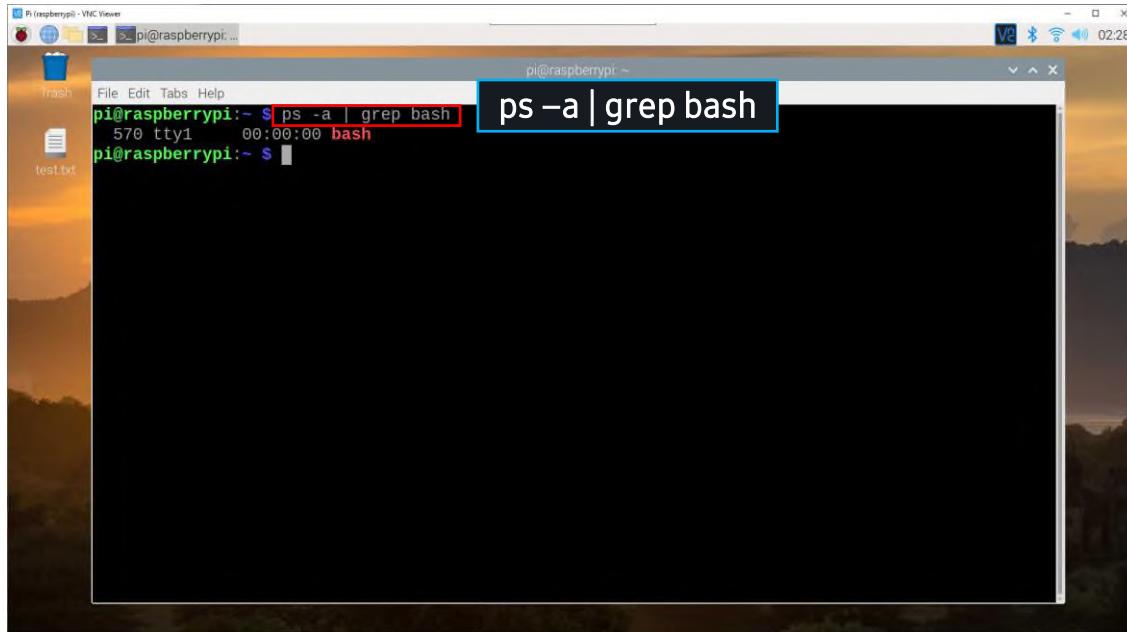
- ▶ Tùy chọn -e in tất cả các tiến trình xử lý đang chạy trên hệ thống.
- ▶ Những giá trị của TTY với ? hầu hết là các tiến trình nền chạy bởi hệ thống dưới dạng tiến trình nền.

The screenshot shows a terminal window titled 'pi@raspberrypi: ~'. It displays two command outputs:

```
pi@raspberrypi:~$ ps -a
  PID TTY      TIME CMD
 570 pts/0    00:00:00 bash
1542 pts/0    00:00:00 ps
pi@raspberrypi:~$ ps -e
  PID TTY      TIME CMD
  1 ?        00:00:00 systemd
  2 ?        00:00:00 kthreadd
  3 ?        00:00:00 rcu_gp
  4 ?        00:00:00 rcu_par_gp
  8 ?        00:00:00 mm_percpu_wq
  9 ?        00:00:00 rcu_tasks_rude_
 10 ?       00:00:00 rcu_tasks_trace
 11 ?       00:00:00 ksoftirqd/0
 12 ?       00:00:00 rcu_sched
 13 ?       00:00:00 migration/0
 14 ?       00:00:00 cpuhp/0
 15 ?       00:00:00 cpuhp/1
 16 ?       00:00:00 migration/1
 17 ?       00:00:00 ksoftirqd/1
 20 ?       00:00:00 cpuhp/2
 21 ?       00:00:00 migration/2
 22 ?       00:00:00 ksoftirqd/2
 25 ?       00:00:00 cpuhp/3
 26 ?       00:00:00 migration/3
 27 ?       00:00:00 ksoftirqd/3
```

### Liên kết lệnh ps và grep

| Nối các lệnh ps và grep với | để truy xuất thông tin về một quy trình cụ thể



```
pi@raspberrypi:~ $ ps -a | grep bash
 570 pts/1    00:00:00 bash
pi@raspberrypi:~ $
```

## kill

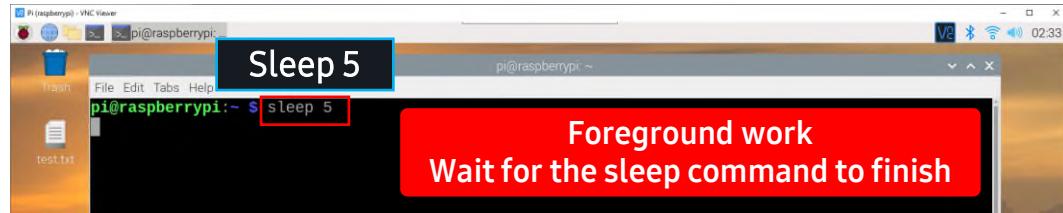
Chức năng	Gửi tín hiệu đã chỉ định đến tiến trình	
Mẫu	kill [ký hiệu] PID...	<span style="border: 1px solid black; padding: 2px;">Ctrl</span>
Ký hiệu	<ul style="list-style-type: none"><li>-2 : Gửi tín hiệu ngắn (+C)</li><li>-9 : Buộc quá trình chấm dứt.</li><li>-15 : Quá trình sắp xếp các tệp liên quan và kết thúc quá trình. Có thể có các quá trình không chấm dứt.</li></ul>	
Ví dụ	kill 1001      kill -15 1001      kill -9 1001	

Tìm tiến trình đã thực thi 'man' và chấm dứt nó với kill.

```
pi@raspberrypi:~ $ ps
  PID TTY          TIME CMD
 249 pts/1    00:00:00 bash
 263 pts/1    00:00:00 man
 273 pts/1    00:00:00 pager
 281 pts/1    00:00:00 ps
pi@raspberrypi:~ $ kill -9 263
pi@raspberrypi:~ $ ps
  PID TTY          TIME CMD
 249 pts/1    00:00:00 bash
 282 pts/1    00:00:00 ps
[1]+  Killed                  man ls
pi@raspberrypi:~ $
```

## Tiến trình bե női

- Một quy trình mà bạn chạy trực tiếp và phải đợi nó hoàn thành và tạo đầu ra

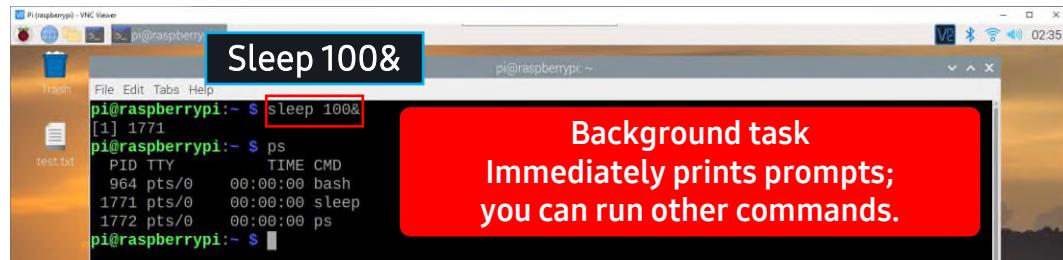


```
pi@raspberrypi:~$ sleep 5
```

Foreground work  
Wait for the sleep command to finish

## Tiến trình ngầm

- Không giống như các quy trình nền trước, không cần đợi các quy trình nền kết thúc trước khi chạy các quy trình khác. Quá trình nền chạy trong nền. Khi bạn chạy lệnh, dấu nhắc sẽ in ngay lập tức



```
pi@raspberrypi:~$ sleep 100&
```

Background task  
Immediately prints prompts;  
you can run other commands.

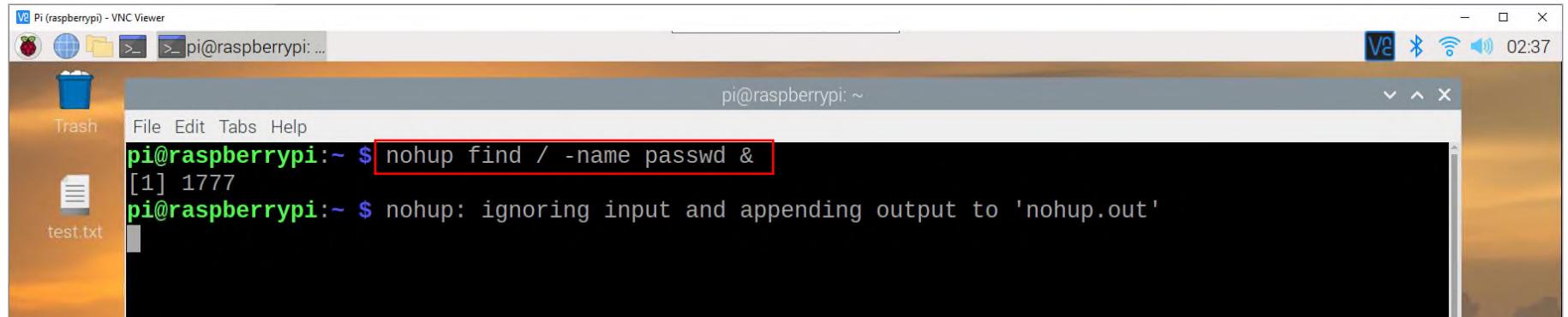
### nohup

| Lệnh chạy một công việc nền cho đến khi hoàn thành ngay cả sau khi thoát khỏi thiết bị đầu cuối hoặc đăng xuất

**nohup**

Chức năng Tiếp tục chạy các công việc nền sau khi đăng xuất

Định dạng Nohup command&



### date

| In dữ liệu hệ thống hiện tại và thông tin thời gian

### w

| In danh sách người dùng hiện đang trực tuyến

```
pi@raspberrypi:~$ date
Mon 23 Aug 2021 03:56:48 A
pi@raspberrypi:~$ w
 03:56:49 up 4:56, 2 users
USER   TTY      FROM          LOGIN@    IDLE   JCPU   PCPU WHAT
pi     tty1     -          22:59   4:57m  0.27s  0.21s -bash
pi     tty7     :0          22:59   4:56m 58.58s  0.55s /usr/bin/lxsession -s LXDE-pi -e
File Edit View LXDE
pi@raspberrypi:~$
```

### whoami

| In thông tin của người dùng hiện đang đăng nhập

### uname -a

| In thông tin về kernel

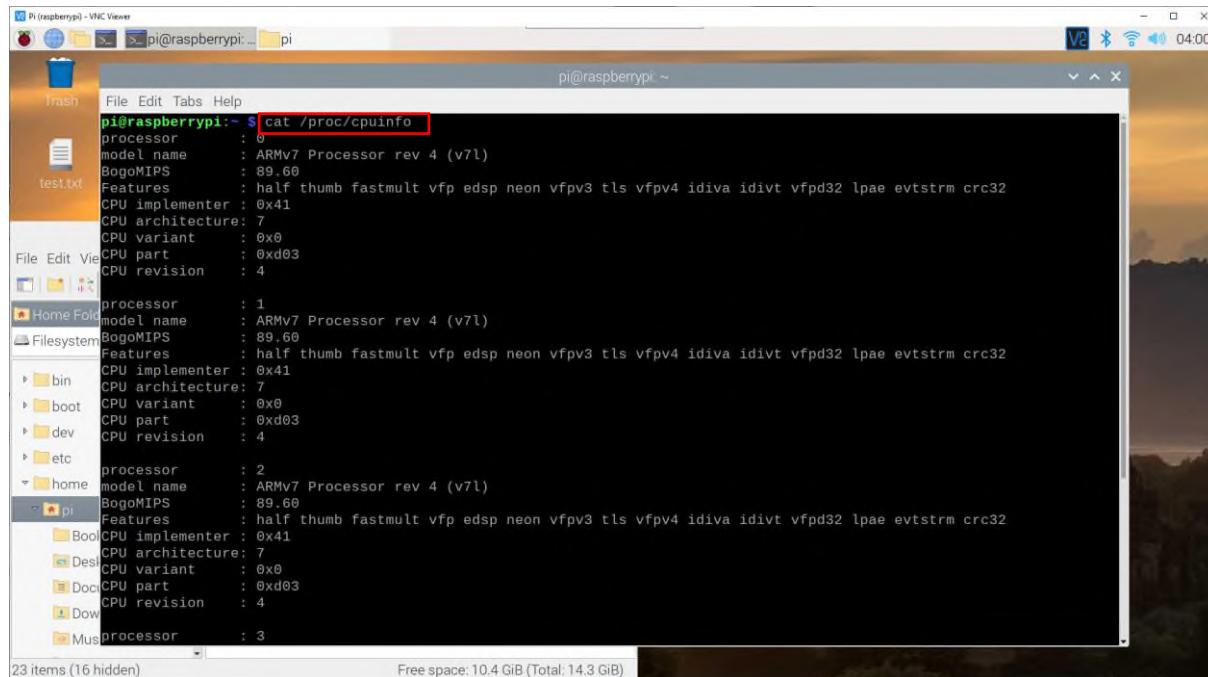
```
pi@raspberrypi:~$ whoami  
pi  
pi@raspberrypi:~$ uname -a  
Linux raspberrypi 5.10.52-v7+ #1 18:10:09 BST 2021 armv7l GNU/Linux  
pi@raspberrypi:~$
```

## 2.4. Các lệnh về điều khiển hệ thống

Bài 02

### cat/proc/cpuinfo

#### In thông tin CPU



A screenshot of a VNC viewer window titled "Pi (raspberrypi) - VNC Viewer". The window shows a terminal session on a Raspberry Pi. The command "cat /proc/cpuinfo" is entered and its output is displayed. The terminal window has a dark background with white text. The desktop environment visible in the background includes icons for Trash, test.txt, Home Folder, and Filesystem. The status bar at the bottom shows "23 items (16 hidden)" and "Free space: 10.4 GiB (Total: 14.3 GiB)".

```
pi@raspberrypi:~ cat /proc/cpuinfo
processor       : 0
model name     : ARMv7 Processor rev 4 (v7l)
BogoMIPS       : 89.60
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x0
CPU part       : 0xd03
CPU revision   : 4

processor       : 1
model name     : ARMv7 Processor rev 4 (v7l)
BogoMIPS       : 89.60
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x0
CPU part       : 0xd03
CPU revision   : 4

processor       : 2
model name     : ARMv7 Processor rev 4 (v7l)
BogoMIPS       : 89.60
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x0
CPU part       : 0xd03
CPU revision   : 4

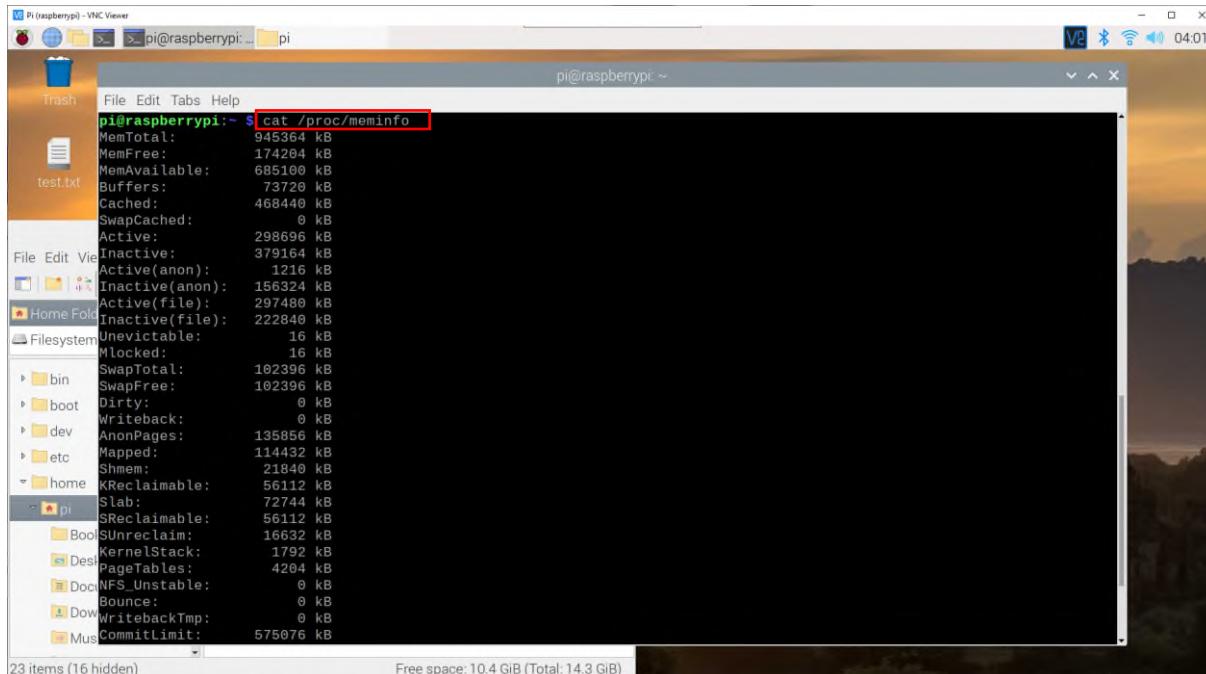
processor       : 3
processor       : 0
model name     : ARMv7 Processor rev 4 (v7l)
BogoMIPS       : 89.60
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x0
CPU part       : 0xd03
CPU revision   : 4
```

## 2.4. Các lệnh về điều khiển hệ thống

Bài 02

### cat/proc/cpuinfo

#### In thông tin bộ nhớ chính



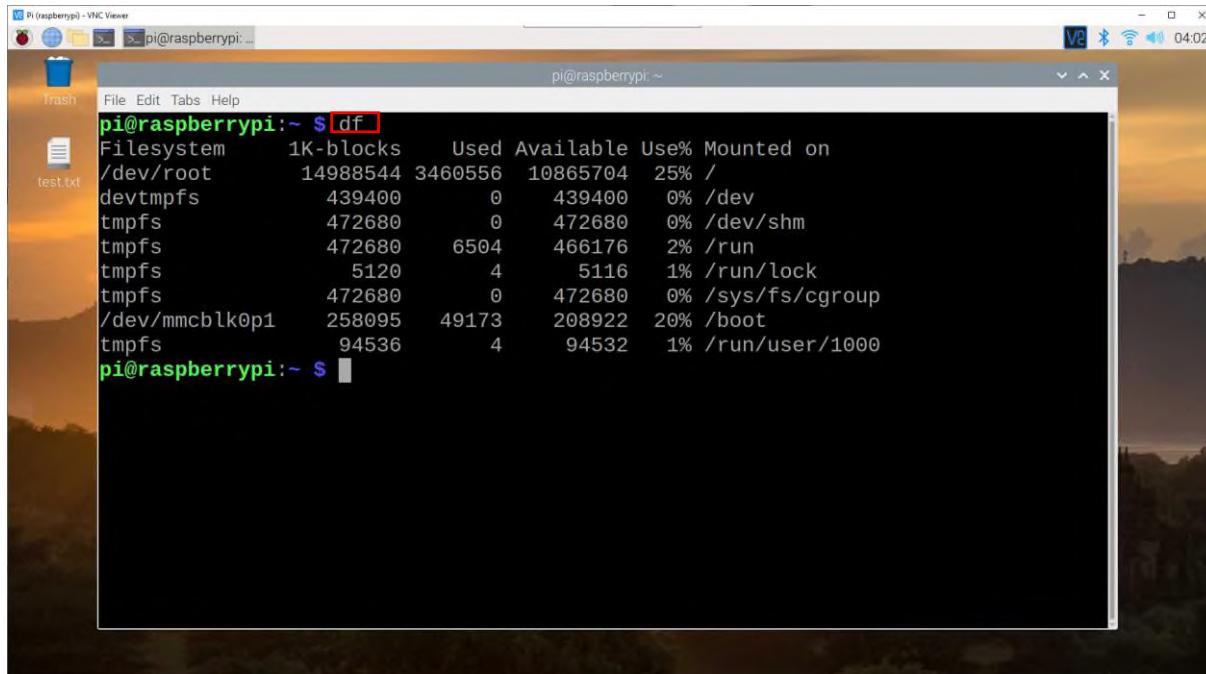
```
pi@raspberrypi:~ cat /proc/meminfo
MemTotal:       945364 kB
MemFree:        174284 kB
MemAvailable:   685100 kB
Buffers:         73720 kB
Cached:          468440 kB
SwapCached:      0 kB
Active:          298696 kB
Inactive:        379164 kB
Active(anon):    1216 kB
Inactive(anon): 156324 kB
Active(file):   297480 kB
Inactive(file): 222840 kB
Unevictable:     16 kB
Mlocked:         16 kB
SwapTotal:      102396 kB
SwapFree:       102396 kB
Dirty:           0 kB
Writeback:       0 kB
AnonPages:      135856 kB
Mapped:          114432 kB
Shmem:           21840 kB
KReclaimable:   56112 kB
Slab:            72744 kB
SReclaimable:   56112 kB
SUncacheable:   16632 kB
KernelStack:    1792 kB
PageTables:     4204 kB
Bounce:          0 kB
WritebackTmp:    0 kB
CommitLimit:    575076 kB
```

## 2.4. Các lệnh về điều khiển hệ thống

## Bài 02

### df

#### In dung lượng đĩa đang sử dụng



```
pi@raspberrypi:~ $ df
Filesystem      1K-blocks   Used   Available  Use% Mounted on
/dev/root        14988544 3460556  10865704  25% /
devtmpfs          439400     0    439400   0% /dev
tmpfs            472680     0    472680   0% /dev/shm
tmpfs            472680   6504   466176   2% /run
tmpfs             5120      4    5116   1% /run/lock
tmpfs            472680     0    472680   0% /sys/fs/cgroup
/dev/mmcblk0p1    258095  49173   208922  20% /boot
tmpfs            94536      4    94532   1% /run/user/1000
pi@raspberrypi:~ $
```

### which

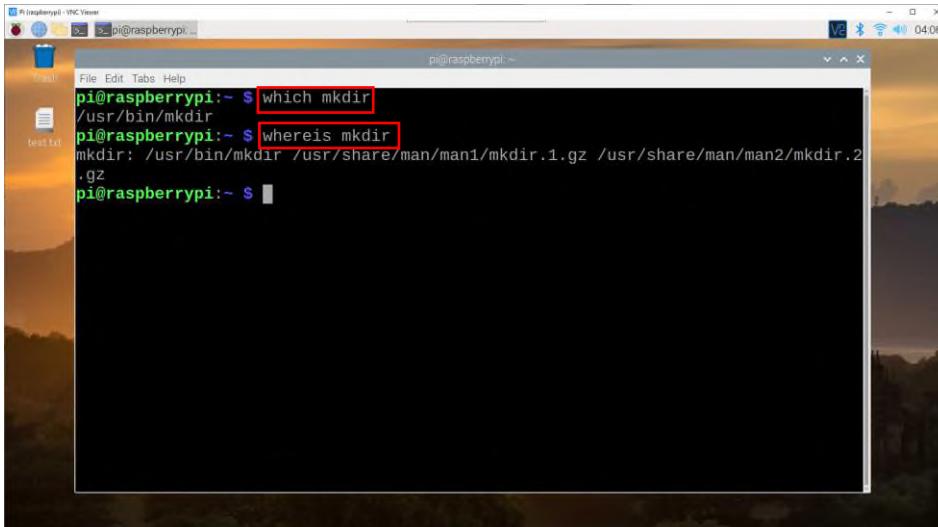
#### || which <ứng dụng>

- ▶ In đường dẫn vị trí của ứng dụng được chỉ định làm mặc định.

### whereis

#### || whereis <ứng dụng>

- ▶ In đường dẫn vị trí của ứng dụng được chỉ định.



The screenshot shows a terminal window titled 'pi@raspberrypi: ~'. The window contains the following command history:

```
pi@raspberrypi:~ $ which mkdir
/usr/bin/mkdir
pi@raspberrypi:~ $ whereis mkdir
mkdir: /usr/bin/mkdir /usr/share/man/man1/mkdir.1.gz /usr/share/man/man2/mkdir.2
.gz
pi@raspberrypi:~ $
```

The first command, 'which mkdir', is highlighted with a red rectangle. The second command, 'whereis mkdir', is also highlighted with a red rectangle.

Bài 2.

## Hướng dẫn lệnh hệ điều hành Raspberry Pi

| 2.1. Kết nối với hệ điều hành Raspberry Pi từ máy tính  
của bạn

| 2.2. Lệnh cơ bản

| 2.3. Các lệnh về Hệ thống tệp

| 2.4. Các lệnh về điều khiển hệ thống

### **| 2.5. Các lệnh về Network**

| 2.6. Các lệnh về Người dùng & Quyền

| 2.7. Công cụ lập trình

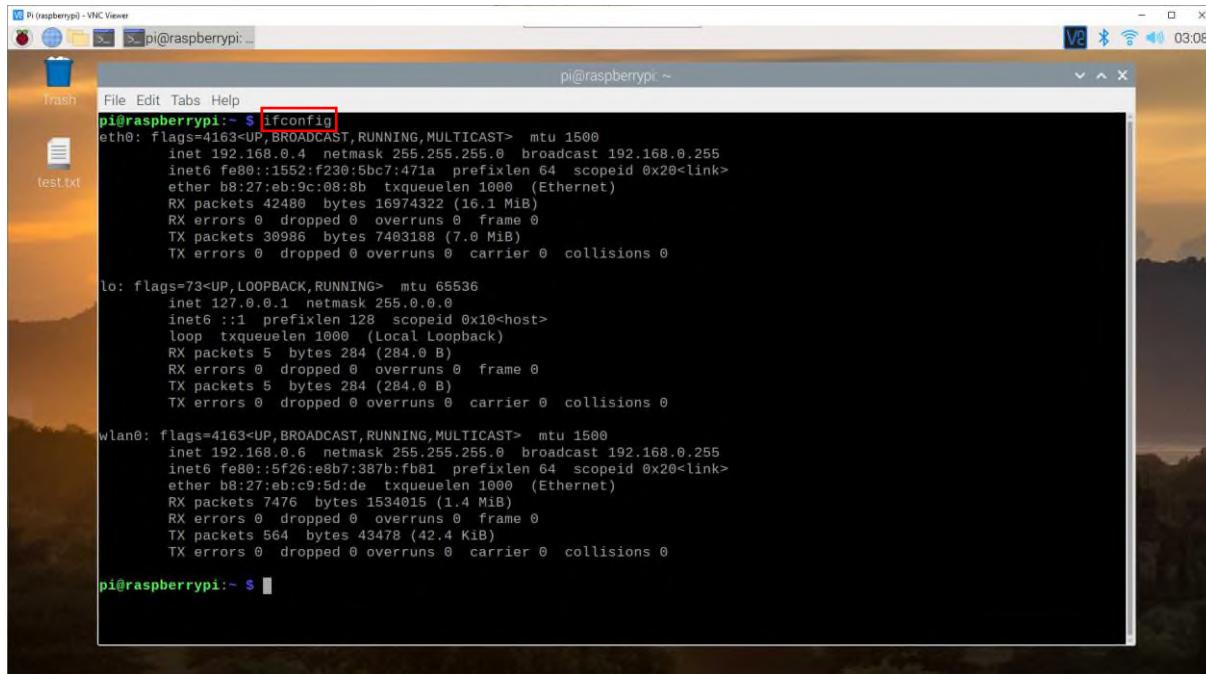
# Các lệnh về Network

| Chúng ta sẽ học gì trong bài này

- ▶ Tìm hiểu những lệnh nào được sử dụng để định cấu hình hoặc kiểm tra cài đặt liên quan đến mạng trong Linux.
- ▶ Tìm hiểu các lệnh liên quan đến mạng cùng với kiến thức nền tảng liên quan.

## ifconfig

### Lệnh kiểm tra và thiết lập thông tin giao diện mạng



```
pi@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.4  netmask 255.255.255.0  broadcast 192.168.0.255
                inet6 fe80::15b2:f230:5bc7:471a  prefixlen 64  scopeid 0x20<link>
                  ether b8:27:eb:9c:08:b8  txqueuelen 1000  (Ethernet)
                    RX packets 42480  bytes 16974322 (16.1 MiB)
                    RX errors 0  dropped 0  overruns 0  frame 0
                    TX packets 30986  bytes 7403188 (7.0 MiB)
                    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
                inet6 ::1  prefixlen 128  scopeid 0x10<host>
                  loop  txqueuelen 1000  (Local Loopback)
                    RX packets 5  bytes 284 (284.0 B)
                    RX errors 0  dropped 0  overruns 0  frame 0
                    TX packets 5  bytes 284 (284.0 B)
                    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.6  netmask 255.255.255.0  broadcast 192.168.0.255
                inet6 fe80::5f26:e8b7:387b:fb81  prefixlen 64  scopeid 0x20<link>
                  ether b8:27:eb:c9:5d:de  txqueuelen 1000  (Ethernet)
                    RX packets 7476  bytes 1534815 (1.4 MiB)
                    RX errors 0  dropped 0  overruns 0  frame 0
                    TX packets 564  bytes 43478 (42.4 KiB)
                    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

pi@raspberrypi:~ $
```

## netstat

■ Lệnh in thông tin như trạng thái kết nối mạng, bảng định tuyến và trạng thái giao diện

```
pi@raspberrypi:~$ netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp     0      0 0.0.0.0:ssh              0.0.0.0:*
tcp     0      0 localhost:ipp            0.0.0.0:*
tcp     0      0 0.0.0.0:5900             0.0.0.0:*
tcp     0      0 192.168.0.4:5900          192.168.0.5:53462 ESTABLISHED
tcp     0      0 localhost:35687          localhost:46002 ESTABLISHED
tcp     0      0 localhost:46002          localhost:35687 ESTABLISHED
tcp6    0      0 [::]:ssh                [::]:*
tcp6    0      0 localhost:ipp            [::]:*
tcp6    0      0 [::]:5900               [::]:*
udp    0      0 0.0.0.0:38812            0.0.0.0:*
udp    0      0 0.0.0.0:bootpc           0.0.0.0:*
udp    0      0 0.0.0.0:ipp              0.0.0.0:*
udp    0      0 0.0.0.0:mdns             0.0.0.0:*
udp6   0      0 [::]:42636              [::]:*
udp6   0      0 [::]:mdns               [::]:*
raw6   0      0 [::]:ipv6-icmp          [::]:*
7

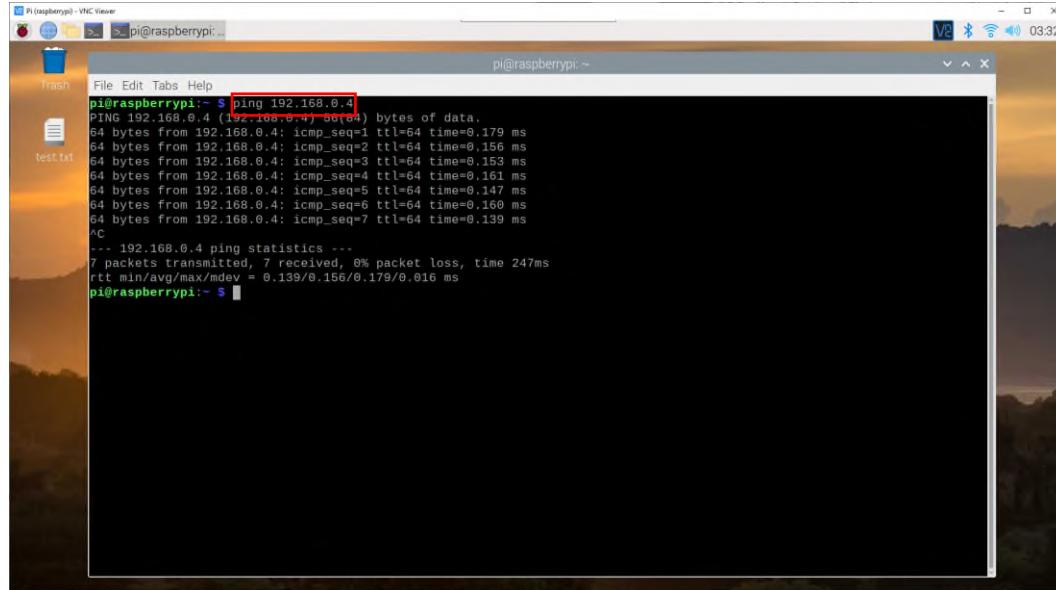
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type            State           I-Node Path
unix  2      [ ]        DGRAM           LISTENING      18950  /run/user/1000/systemd/notify
unix  2      [ ACC ]     STREAM          LISTENING      18953  /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM          LISTENING      18958  /run/user/1000/gnupg/S.gpg-agent.extra
unix  2      [ ACC ]     STREAM          LISTENING      18959  /run/user/1000/gnupg/S.dirmngr
unix  2      [ ACC ]     STREAM          LISTENING      18960  /run/user/1000/gnupg/S.gpg-agent.browser
unix  2      [ ACC ]     STREAM          LISTENING      18961  /run/user/1000/bus
unix  2      [ ACC ]     STREAM          LISTENING      18962  /run/user/1000/gnupg/S.gpg-agent
unix  2      [ ACC ]     STREAM          LISTENING      18963  /run/user/1000/gnupg/S.gpg-agent.ssh
unix  2      [ ACC ]     STREAM          LISTENING      18964  /run/user/1000/pulse/native
unix  2      [ ACC ]     STREAM          LISTENING      19172  /run/user/1000/pcmfam-socket--0
unix  2      [ ACC ]     STREAM          LISTENING      18399  /run/user/1000/menu-cached-0
unix  2      [ ACC ]     STREAM          LISTENING      20249  /run/user/1000/.lxterminal-socket-:0.0
```

## netstat

Loại	Nghĩa
<b>ESTABLISHED</b>	Máy chủ và khách được kết nối với nhau
<b>SYN_SENT</b>	Khách hàng đã yêu cầu kết nối với máy chủ từ xa
<b>SYN_RECV</b>	Trạng thái phản hồi đã được thực hiện liên quan đến yêu cầu của khách hàng từ máy chủ nhưng chưa nhận được tin nhắn của khách hàng
<b>FIN_WAIT1</b>	Ổ cắm đã đóng và kết nối đã kết thúc
<b>FIN_WAIT2</b>	Trạng thái kết nối đã kết thúc, nhưng ổ cắm đang chờ nó kết thúc từ máy chủ từ xa
<b>TIME_WAIT</b>	Trạng thái nơi quá trình xử lý gói kết thúc, nhưng socket đang chờ quá trình xử lý gói chậm trong trường hợp.
<b>CLOSE</b>	Ổ cắm chưa được sử dụng
<b>CLOSE_WAIT</b>	Trạng thái nơi máy chủ từ xa đã kết thúc và đang đợi để ổ cắm có thể đóng
<b>LAST_ACK</b>	Đang chờ gói ACK cuối cùng (Máy chủ từ xa, trạng thái TẮT ổ cắm)
<b>LISTEN</b>	Trạng thái nơi ổ cắm đang chờ gói đến máy chủ
<b>CLOSING</b>	Trạng thái nơi dữ liệu được truyền đi bị mất
<b>UNKNOWN</b>	Trạng thái không thể kiểm tra trạng thái ổ cắm

### ping

#### Lệnh kiểm tra xem máy chủ lưu trữ bên ngoài có thể truy cập mạng không



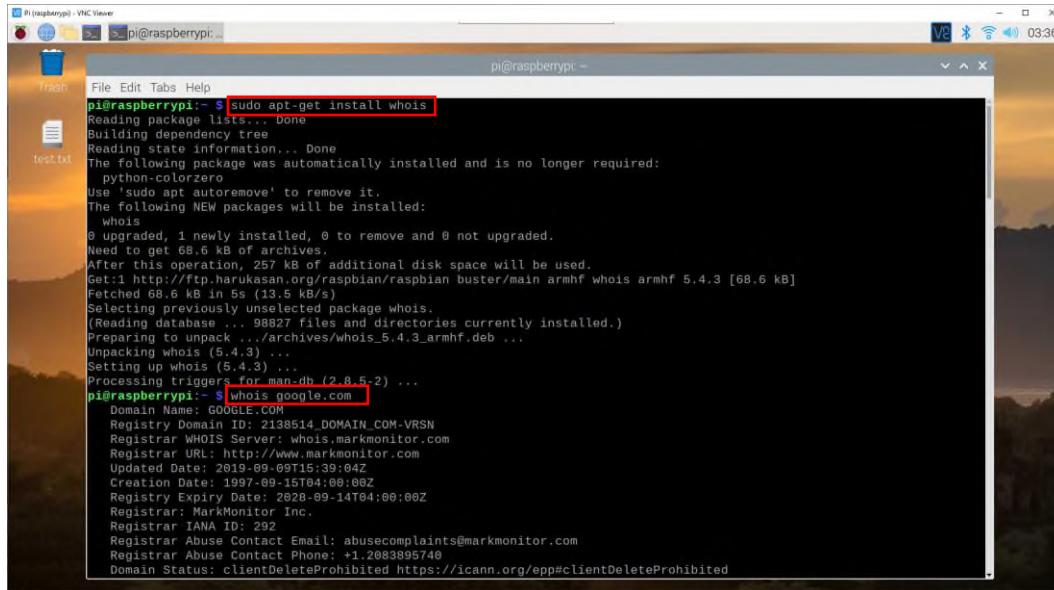
```
pi@raspberrypi: ~
```

```
pi@raspberrypi: $ ping 192.168.0.4
PING 192.168.0.4 (192.168.0.4) 64 bytes of data.
64 bytes from 192.168.0.4: icmp_seq=1 ttl=64 time=0.179 ms
64 bytes from 192.168.0.4: icmp_seq=2 ttl=64 time=0.156 ms
64 bytes from 192.168.0.4: icmp_seq=3 ttl=64 time=0.153 ms
64 bytes from 192.168.0.4: icmp_seq=4 ttl=64 time=0.161 ms
64 bytes from 192.168.0.4: icmp_seq=5 ttl=64 time=0.147 ms
64 bytes from 192.168.0.4: icmp_seq=6 ttl=64 time=0.160 ms
64 bytes from 192.168.0.4: icmp_seq=7 ttl=64 time=0.139 ms
^C
--- 192.168.0.4 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 247ms
rtt min/avg/max/mdev = 0.139/0.156/0.179/0.016 ms
pi@raspberrypi: ~
```

- ▶ Sử dụng định dạng ping <địa chỉ ip>.
- ▶ Để thoát, bạn có thể ngắt bằng Ctrl + c.

## whois

| Lệnh cho bạn biết chủ sở hữu của tên miền là ai



```
pi@raspberrypi: ~$ sudo apt-get install whois
Reading package lists... Done
Building dependency tree
Building state information... Done
The following package was automatically installed and is no longer required:
  python-colorzero
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  whois
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 68.6 kB of archives.
After this operation, 257 kB of additional disk space will be used.
Get:1 http://ftp.harukasan.org/raspbian/raspbian buster/main armhf whois armhf 5.4.3 [68.6 kB]
Fetched 68.6 kB in 5s (13.5 kB/s)
Selecting previously unselected package whois.
(Reading database ... 98827 files and directories currently installed.)
Preparing to unpack .../archives/whois_5.4.3_armhf.deb ...
Unpacking whois (5.4.3) ...
Setting up whois (5.4.3) ...
Processing triggers for man-db (2.8.5-2) ...
pi@raspberrypi: ~$ whois google.com
Domain Name: GOOGLE.COM
Registry Domain ID: 2188514.DOMAIN.COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2018-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895740
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
```

- ▶ Cài đặt whois trước bằng cách sử dụng cài đặt apt-get.
- ▶ whois <Tên miền> có thể được sử dụng. Nó thường được sử dụng để xác minh xem tên miền có đang được người khác sử dụng hay không.

## 2.5. Các lệnh về Network

## Bài 02

### dig

#### Lệnh in thông tin DNS liên quan đến tên miền (Viết tắt của Domain Information Groper)

```
pi@raspberrypi: ~$ sudo apt-get install dnsutils
Reading package list... done
Building dependency tree
Reading state information... done
The following package was automatically installed and is no longer required:
  python-colorzero
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libirs161
Suggested packages:
  rblcheck
The following NEW packages will be installed:
  dnsutils libirs161
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 597 kB of archives.
After this operation, 952 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.harukasan.org/raspbian/raspbian buster/main armhf libirs161 armhf 1:9.11.5.P4+dfsg-5.1+deb10u5 [236 kB]
Get:2 http://ftp.harukasan.org/raspbian/raspbian buster/main armhf dnsutils armhf 1:9.11.5.P4+dfsg-5.1+deb10u5 [361 kB]
Fetched 597 kB in 4s (151 kB/s)
Selecting previously unselected package libirs161:armhf.
(Reading database ... 98851 files and directories currently installed.)
Preparing to unpack .../libirs161_1%3a9.11.5.P4+dfsg-5.1+deb10u5_armhf.deb ...
Unpacking libirs161:armhf (1:9.11.5.P4+dfsg-5.1+deb10u5) ...
Selecting previously unselected package dnsutils.
Preparing to unpack .../dnsutils_1%3a9.11.5.P4+dfsg-5.1+deb10u5_armhf.deb ...
Unpacking dnsutils (1:9.11.5.P4+dfsg-5.1+deb10u5) ...
Setting up libirs161:armhf (1:9.11.5.P4+dfsg-5.1+deb10u5) ...
Setting up dnsutils (1:9.11.5.P4+dfsg-5.1+deb10u5) ...
Processing triggers for libc-bin (2.28-10+rpi1) ...
Processing triggers for man-db (2.8.5-2) ...
pi@raspberrypi: ~$ dig google.com

<<>> DIG 9.11.5.P4-5.1+deb10u5-Raspbian <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26175
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 4096
; QUESTION SECTION:
;google.com.           IN      A
; ANSWER SECTION:
google.com.        21     IN      A      172.217.25.110

;; Query time: 2 msec
;; SERVER: 210.220.163.82#53(210.220.163.82)
;; WHEN: Mon Aug 23 03:41:42 KST 2021
;; MSG SIZE  rcvd: 55
pi@raspberrypi: ~$
```

- Nó được sử dụng rộng rãi để kiểm tra xem truy vấn DNS và phản hồi đối với bộ tên miền theo quan điểm của người dùng chung có hoạt động bình thường không sau khi cấu hình máy chủ tên miền DNS và cài đặt tên miền hoàn tất.
- Sử dụng lệnh đào sau khi cài đặt bằng apt-get install dnsutils.

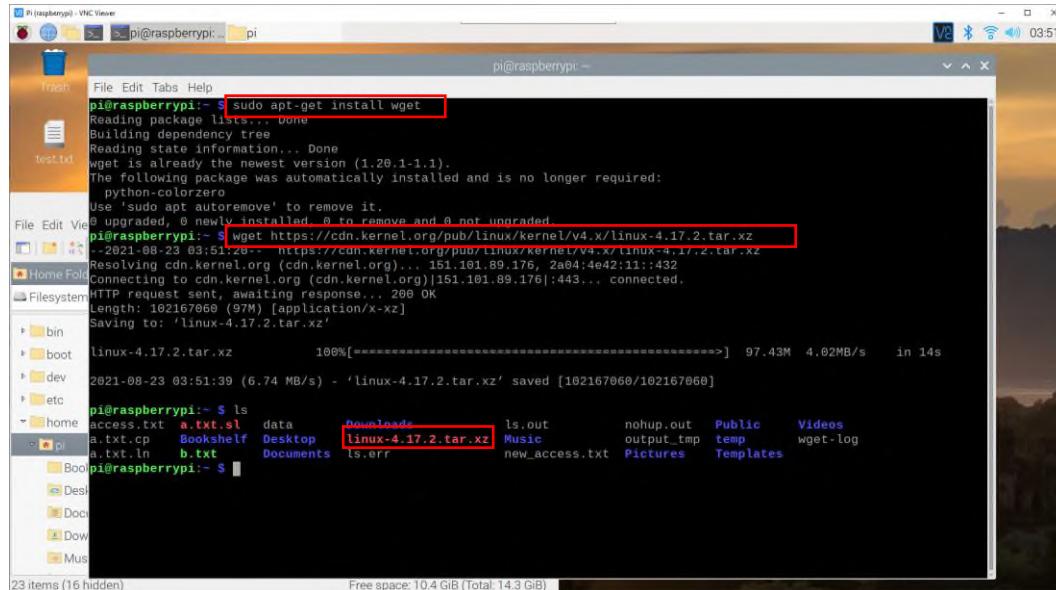
## wget

### I Lệnh tải tệp trên web (Web Get)

- ▶ Wget là một trình tải xuống mạng không tương tác. Nói cách khác, nó thực hiện chức năng tải dữ liệu từ mạng. Nó hỗ trợ các giao thức HTTP, HTTPS, FTP và có thể lấy dữ liệu từ HTTP proxy. Nếu wget không yêu cầu bất kỳ tương tác nào, điều đó có nghĩa là nó có thể chạy ở trạng thái nền ngay cả khi người dùng chưa đăng nhập.
- ▶ wget là một chương trình chắc chắn hoạt động rất tốt trong môi trường mạng chậm hoặc không ổn định. Nếu kết nối bị mất do môi trường mạng không ổn định, thì cũng có chức năng tải xuống từ điểm bị ngắt kết nối.

# wget

## Lệnh tải tệp trên web (Web Get)



The screenshot shows a terminal window on a Raspberry Pi system. The user has run the command `sudo apt-get install wget` to install the wget package. Subsequently, they have run the command `wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.17.2.tar.xz` to download the specified Linux kernel tarball from the official website. The terminal output shows the progress of the download, indicating it was completed at 03:51:29 on August 23, 2021, with a speed of 6.74 MB/s.

```

pi@raspberrypi: ~
pi@raspberrypi: ~ $ sudo apt-get install wget
Reading package lists... done
Building dependency tree
Reading state information... done
wget is already the newest version (1.20.1-1.1).
The following package was automatically installed and is no longer required:
  python-colorzero
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi: ~ $ wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.17.2.tar.xz
--2021-08-23 03:51:26 -> https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.17.2.tar.xz
Resolving cdn.kernel.org (cdn.kernel.org)... 151.101.89.176, 2a04:4e42:11:432
Connecting to cdn.kernel.org (cdn.kernel.org)|151.101.89.176|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 102167869 (97M) [application/x-xz]
Saving to: 'linux-4.17.2.tar.xz'

2021-08-23 03:51:39 (6.74 MB/s) - 'linux-4.17.2.tar.xz' saved [102167060/102167868]

pi@raspberrypi: ~ $ ls
access.txt  a.txt  e1  data  Downloads  ls.out      nohup.out  Public    Videos
a.txt.cp   a.txt.e1  bookshelf  Desktop  linux-4.17.2.tar.xz  Music      output_tmp  temp      wget-log
a.txt.ln   b.txt   Documents  ls.err     new_access.txt  Pictures   Templates
Boo...
pi@raspberrypi: ~ $ 

```

- ▶ Đầu tiên, cài đặt bằng cách sử dụng sudo apt-get install wget.
- ▶ Viết địa chỉ của tệp để tải xuống sau wget.
- ▶ Ví dụ, tải xuống kho lưu trữ tar của nhân Linux.
- ▶ Địa chỉ :  
<https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.17.2.tar.xz>
- ▶ Wget kiểm tra địa chỉ IP của miền và kết nối với máy chủ từ xa để bắt đầu quá trình chuyển.
- ▶ Trong khi tải xuống, wget hiển thị tên tệp, kích thước tệp, tốc độ tải xuống và thời gian hoàn tất tải xuống ước tính cùng với thanh tiến trình. Khi quá trình tải xuống hoàn tất, tệp đã tải xuống có thể được tìm thấy trong thư mục làm việc hiện tại của bạn.

Bài 2.

## Hướng dẫn lệnh hệ điều hành Raspberry Pi

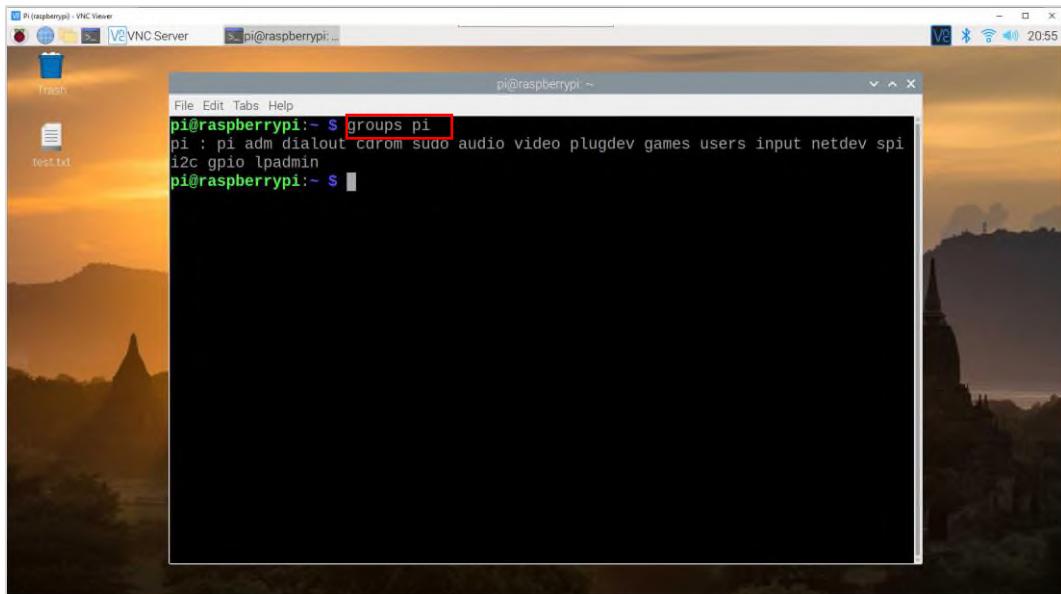
- | 2.1. Kết nối với hệ điều hành Raspberry Pi từ máy tính  
của bạn
- | 2.2. Lệnh cơ bản
- | 2.3. Các lệnh về Hệ thống tệp
- | 2.4. Các lệnh về điều khiển hệ thống
- | 2.5. Các lệnh về Network
- | **2.6. Các lệnh về Người dùng & Quyền**
- | 2.7. Công cụ lập trình

# Các lệnh về Người dùng và Quyền

| Chúng ta sẽ học gì trong bài này

- ▶ Tìm hiểu rằng việc đặt quyền của người dùng hoặc nhóm là quan trọng vì Linux cho phép những người dùng và nhóm khác nhau sử dụng một CTNH cùng một lúc.
- ▶ Tìm hiểu rằng ngay cả khi bạn chạy hoặc sửa đổi nó, bạn sẽ không thể thực thi tác vụ mình muốn vì bạn không thể kiểm soát các quyền.
- ▶ Tìm hiểu cách quản lý người dùng và nhóm trong Linux cũng như cách gán hoặc xóa quyền truy cập vào tệp hoặc thư mục trong Linux.

## Các lệnh về Group



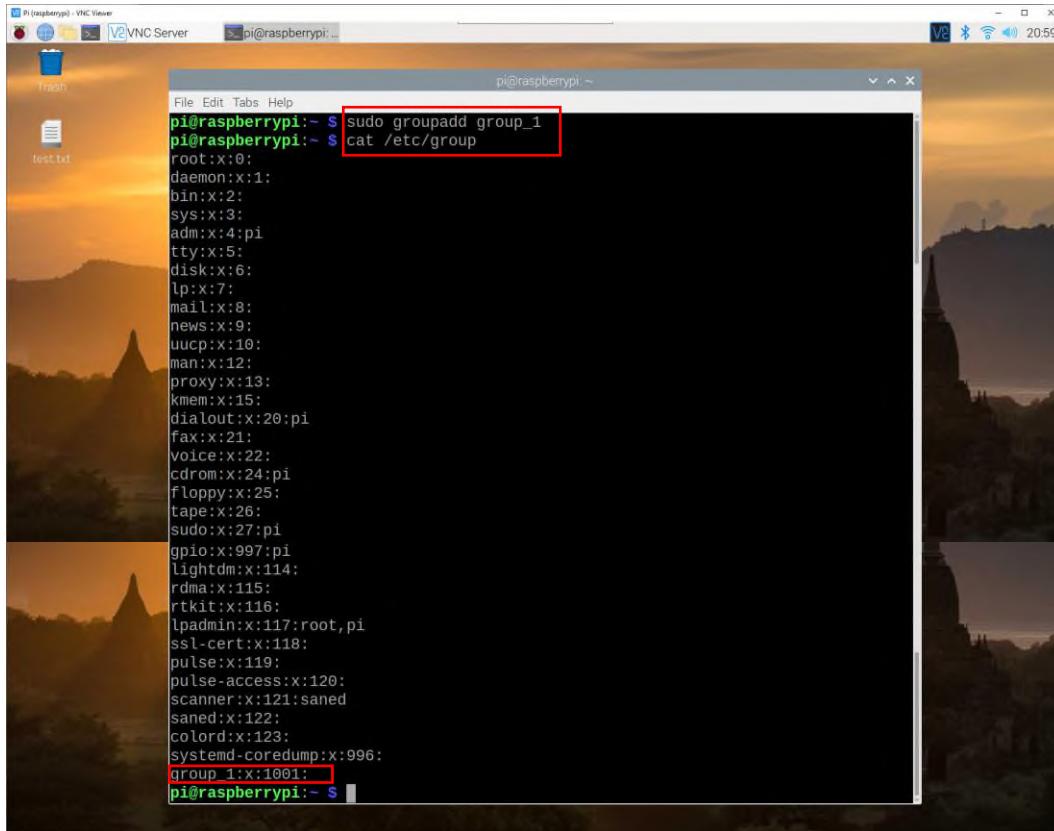
### Người dùng group

- ▶ Việc quản lý đúng hệ điều hành Raspberry Pi phần lớn được phân loại thành nhóm, chủ sở hữu và những người khác.
  - ▶ Raspberry Pi hiếm khi được sử dụng như một máy chủ mà nhiều người dùng có thể truy cập cùng một lúc. Tìm hiểu cách sử dụng các lệnh chỉ dành cho quản lý nhóm và người dùng.
- ▶ groups <user>
- ▶ Lệnh kiểm tra nhóm mà <người dùng> thuộc về
  - ▶ Bạn có thể thấy rằng nhóm mà người dùng pi thuộc về cũng thuộc một số nhóm chứ không chỉ pi.

```
$ groups pi
```

## Các lệnh Group

### | groupadd <tên nhóm>



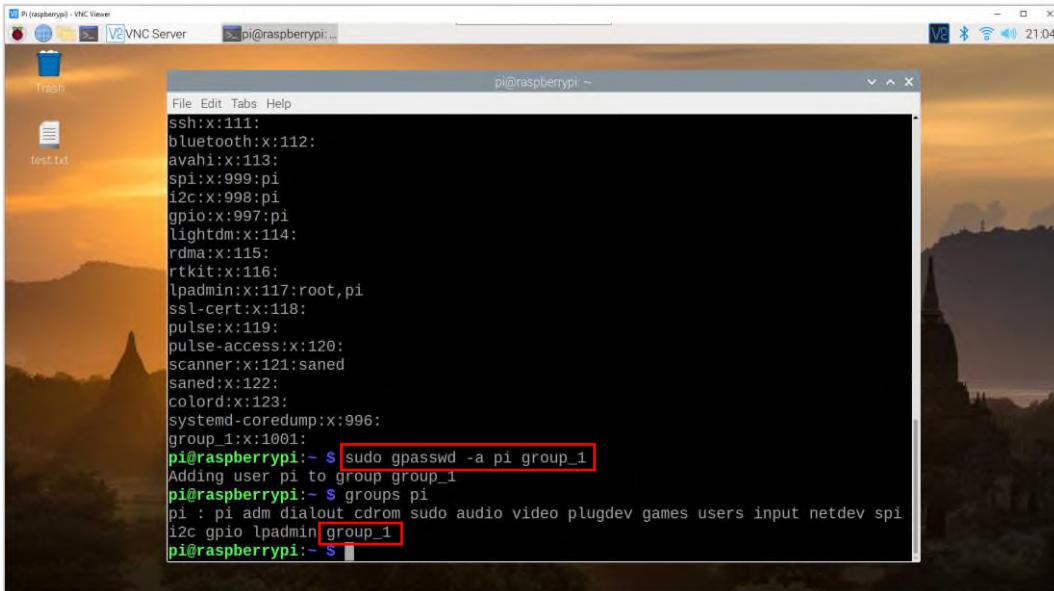
```
pi@raspberrypi:~ $ sudo groupadd group_1
pi@raspberrypi:~ $ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
pi:x:21:
fax:x:22:
voice:x:22:
cdrom:x:24:
floppy:x:25:
tape:x:26:
sudo:x:27:
pi:x:997:
gpio:x:997:
lightdm:x:114:
rdma:x:115:
rtkit:x:116:
lpadmin:x:117:
root,pi:x:118:
ssl-cert:x:118:
pulse:x:119:
pulse-access:x:120:
scanner:x:121:
saned:x:122:
colord:x:123:
systemd-coredump:x:996:
group_1:x:1001:
pi@raspberrypi:~ $
```

- ▶ Sử dụng lệnh groupadd để thêm một nhóm mới.
- ▶ lệnh groupadd yêu cầu quyền root.
- ▶ Sử dụng sudo trước lệnh cho nhóm với tư cách là người dùng pi.
- ▶ Các nhóm đã thêm được thêm vào tệp /etc/group.
- ▶ Sử dụng cat để xem nhóm nào đã được thêm vào.

```
$ sudo groupadd group_1
$ cat etc/group
```

## Các lệnh Group

| gpasswd <option> <user> <group name>



- ▶ Bạn có thể đặt mật khẩu cho nhóm hoặc thêm người dùng vào nhóm.
  - ▶ Thêm người dùng pi vào nhóm group\_1 và kiểm tra xem người dùng pi thuộc nhóm nào.

```
$ sudo gpasswd -a pi group_1  
$ groups pi
```

## Các lệnh Group

| groupdel <group name>

```
File Edit Tabs Help
i2c:x:998:pi
gpio:x:997:pi
lightdm:x:114:
rdma:x:115:
rtkit:x:116:
lpadmin:x:117:root,pi
ssl-cert:x:118:
pulse:x:119:
pulse-access:x:120:
scanner:x:121:saned
saned:x:122:
colord:x:123:
systemd-coredump:x:996:
group_1:x:1001:
pi@raspberrypi:~ $ sudo gpasswd -a pi group_1
Adding user pi to group group_1
pi@raspberrypi:~ $ groups pi
pi : pi adm dialout cdrom sudo audio video plugdev games users input netdev spi
i2c gpio lpadmin group_1
pi@raspberrypi:~ $ sudo groupdel group_1
pi@raspberrypi:~ $ groups pi
pi : pi adm dialout cdrom sudo audio video plugdev games users input netdev spi
i2c gpio lpadmin
pi@raspberrypi:~ $
```

- ▶ Sử dụng lệnh groupdel để xóa một nhóm

```
$ sudo groupdel group_1  
$ groups pi
```

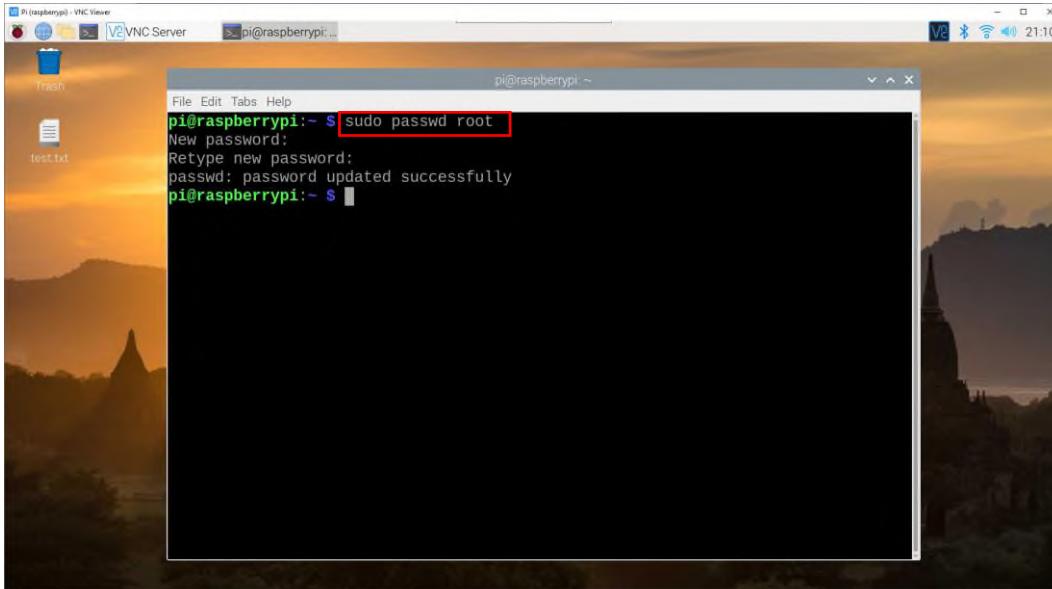
## Các lệnh về người dùng

### I Quản lý người dùng

- ▶ Tất cả hệ điều hành Linux đều có một người dùng tên là root, theo mặc định là Người giám sát. Nếu hệ điều hành Raspberry Pi được cài đặt, sẽ có một tài khoản người dùng mặc định có tên là pi.
- ▶ Mật khẩu cho người dùng pi được đặt làm mật khẩu được nhập trong quá trình thiết lập Hệ điều hành Raspberry Pi. Người dùng root không thể sử dụng nó vì mật khẩu không được đặt. Người dùng root là quản trị viên có quyền truy cập vào tất cả các tài nguyên của Hệ điều hành Raspberry Pi bằng một tài khoản đặc biệt. Ban đầu, không có mật khẩu nào được đặt, vì vậy tài khoản người dùng root không thể sử dụng được.

## Các lệnh về người dùng

### passwd <User ID>

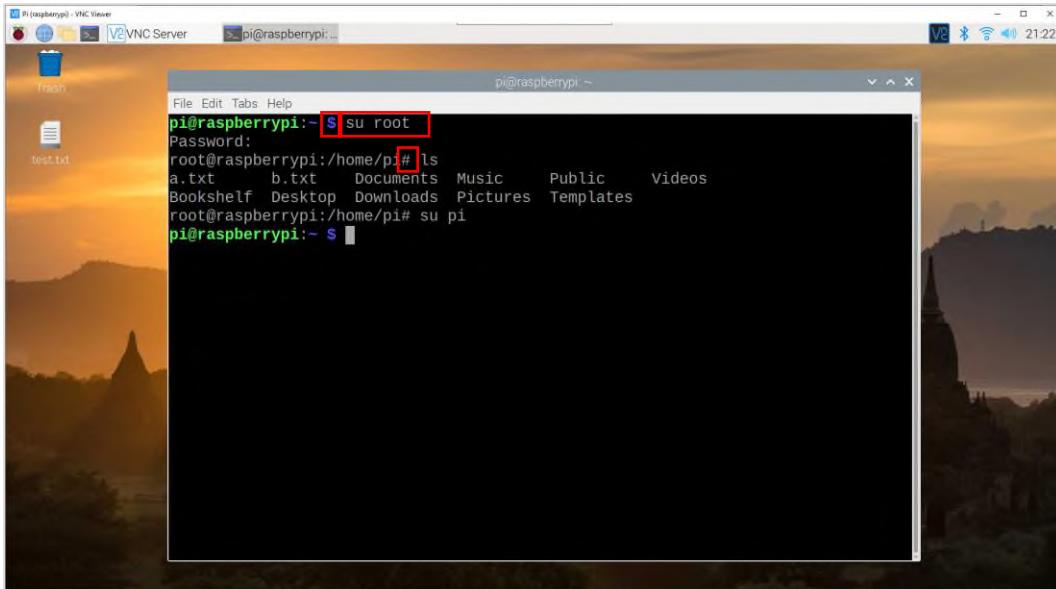


- ▶ Đặt mật khẩu cho người dùng.
- ▶ Bạn không thể thấy mật khẩu bạn đã nhập. Không hoảng loạn. Đặt mật khẩu, nhập lại một lần nữa và xác nhận.
- ▶ Bây giờ bạn có thể sử dụng người dùng root.

```
$ sudo passwd root
```

## Các lệnh về người dùng

| su <User ID>

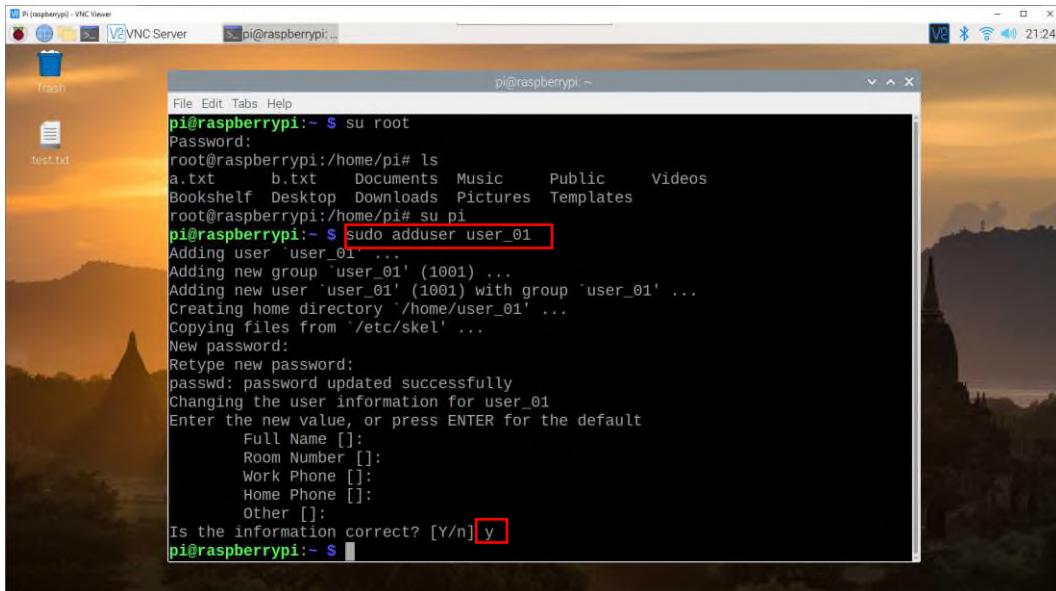


- ▶ Lệnh su là viết tắt của Chuyển đổi người dùng.
- ▶ Chuyển sang người dùng, ID người dùng.
- ▶ Khi người dùng được chuyển đổi, tất cả các quyền của người dùng đã chuyển đổi cũng bị thay đổi.
- ▶ Nếu người dùng mà bạn muốn chuyển sang có mật khẩu, họ phải nhập mật khẩu đó.
- ▶ Để tham khảo, trong bash shell, người dùng root sử dụng dấu nhắc # và người dùng bình thường sử dụng dấu nhắc \$.
- ▶ Sau khi chuyển đổi, bạn có thể thấy sự thay đổi từ pi@raspberrypi thành root@raspberrypi.

\$ su root

## Các lệnh về người dùng

### adduser <User ID>



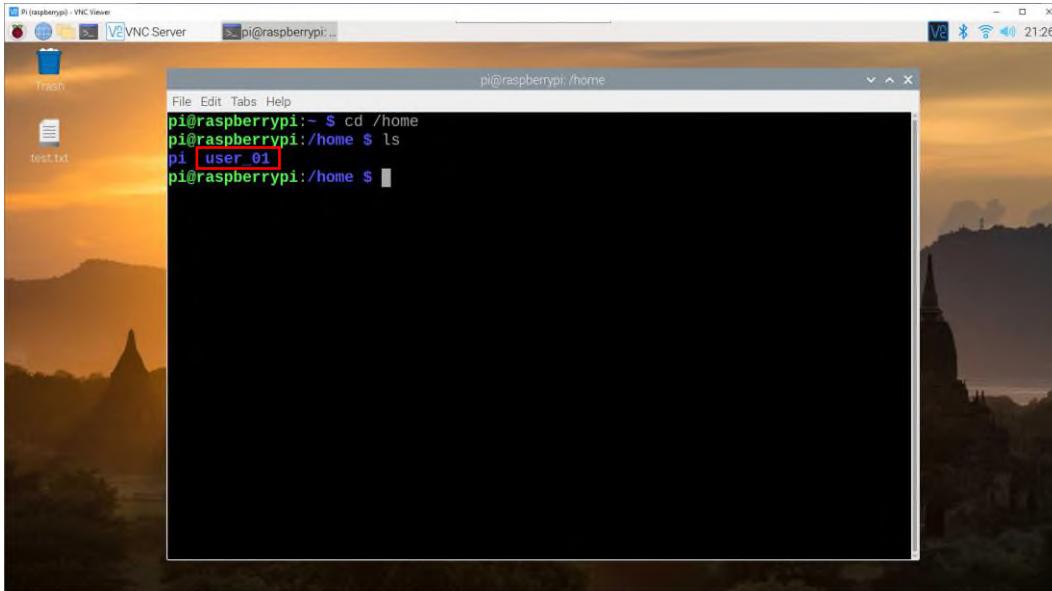
```
pi@raspberrypi:~$ su root
Password:
root@raspberrypi:/home/pi# ls
a.txt      b.txt      Documents  Music      Public      Videos
Bookshelf  Desktop   Downloads  Pictures  Templates
root@raspberrypi:/home/pi# su pi
pi@raspberrypi:~$ sudo adduser user_01
Adding user `user_01' ...
Adding new group `user_01' (1001) ...
Adding new user `user_01' (1001) with group `user_01' ...
Creating home directory `/home/user_01' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for user_01
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
pi@raspberrypi:~$
```

- ▶ Thêm người dùng mới. Đối với lệnh useradd, bạn phải có quyền của người dùng root.
- ▶ Sau khi thực hiện lệnh và đặt mật khẩu, hãy nhập thông tin tương ứng với Họ và Tên. Nếu muốn, hãy nhập thông tin khác và nhấn Enter.
- ▶ Nếu bạn muốn bỏ qua, hãy để trống phần chi tiết và nhấn Enter.
- ▶ Khi được hỏi thông tin có chính xác không, hãy nhập y và nhấn Enter.

```
$ sudo adduser user_01
```

# Các lệnh về người dùng

## | useradd <User ID>

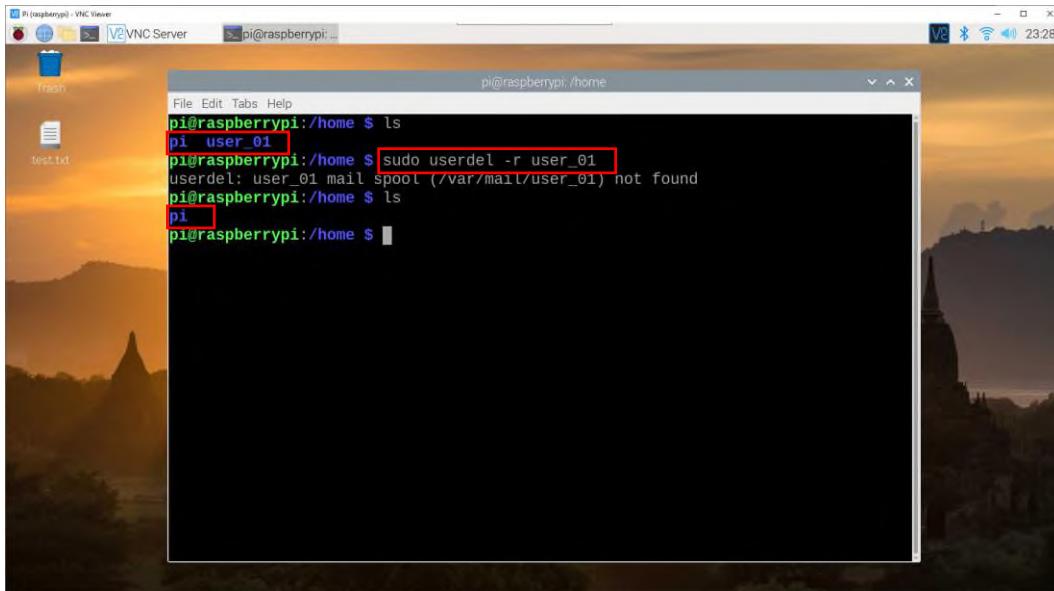


- ▶ Kiểm tra xem user\_01 đã được thêm chưa.

```
$ cd /home
$ ls
```

## Các lệnh về người dùng

### | userdel <User ID>

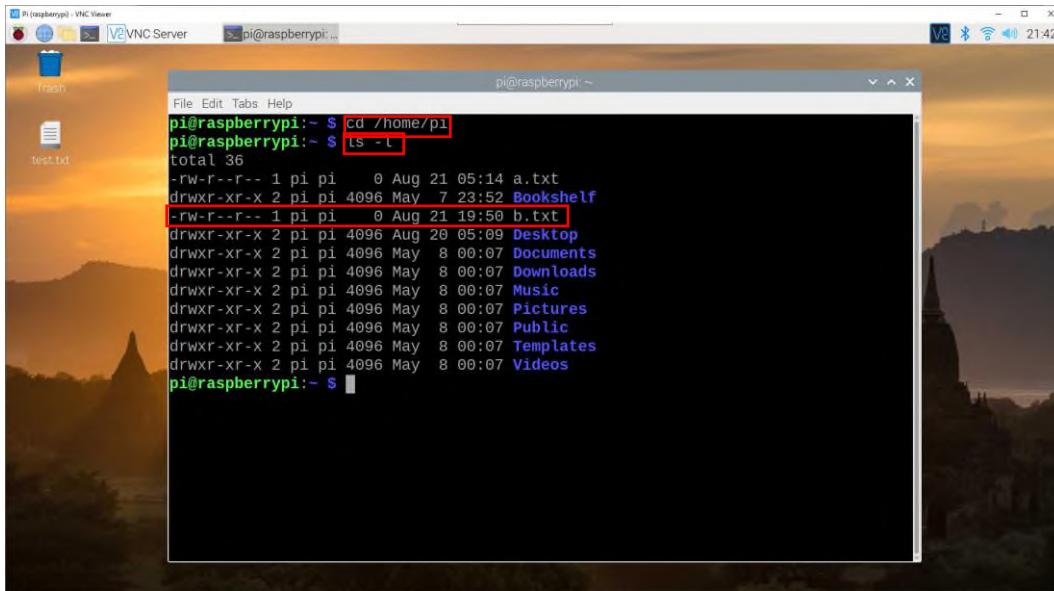


- ▶ Xóa tài khoản người dùng.
- ▶ Lệnh yêu cầu quyền của người dùng root
- ▶ Nếu được sử dụng mà không có tùy chọn -r, chỉ người dùng bị xóa và giám đốc /home/user\_01 vẫn còn. Để xóa giám đốc này và tất cả thông tin liên quan đến tài khoản cần xóa, hãy thêm tùy chọn -r.

```
$ sudo userdel -r user_01
```

## Làm cách nào chúng tôi có thể kiểm tra quyền của tệp?

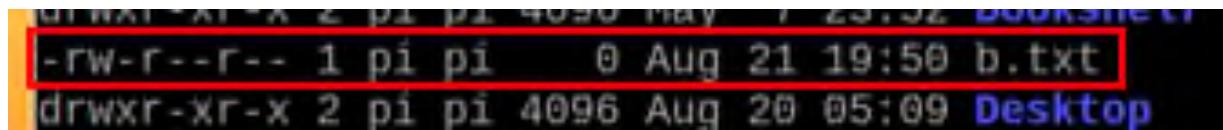
Hãy cùng tìm hiểu về tệp



- Sử dụng lệnh `ls -l` để liệt kê các quyền, số lượng tệp chứa, chủ sở hữu, nhóm, kích thước tệp, ngày chỉnh sửa, tên tệp, v.v.
- Ngay cả khi bạn biết cách xem thông tin chi tiết về một tệp, bạn vẫn không hiểu nó. Hãy xem xét các quyền để đọc, ghi và thực thi tệp.

```
$ cd /home/pi
$ ls -l
```

## Thuộc tính tệp



```

drwxr-xr-x 2 pi pi 4096 Aug 20 05:09 Desktop
-rw-r--r-- 1 pi pi 0 Aug 21 19:50 b.txt

```

STT	Giá trị thuộc tính	Giải thích
1	-	Loại tệp (-: tệp chung, thư mục d:)
2	rw-r--r--	Thông báo quyền đọc, ghi và thực thi tệp
3	1	Số liên kết cứng
4	pi	ID đăng nhập của chủ sở hữu tập tin
5	pi	Tên nhóm của chủ sở hữu tập tin
6	0	Kích thước tệp (Byte)
7	Aug 21 19:50	Ngày tệp được sửa đổi lần cuối
8	b.txt	Tên tệp

## Các loại kiểm soát/quyền truy cập

Quyền	Tệp	Danh mục
Đọc (r)	Đọc hoặc sao chép một tập tin	Kiểm tra danh sách thư mục bằng lệnh 'ls' (Các tùy chọn của lệnh 'ls' yêu cầu quyền thực thi)
Viết (w)	Chỉnh sửa, di chuyển và xóa tệp (Bạn phải có quyền ghi trong thư mục)	Tạo hoặc xóa tệp
Thực thi (x)	Chạy tập tin (Đối với tập lệnh di động hoặc tệp thực thi)	Di chuyển hoặc sao chép tệp vào thư mục bằng lệnh 'cd'

## Làm thế nào để chỉ ra kiểm soát truy cập

- | Cho biết các ký tự có thể đọc, ghi và thực thi tệp theo danh mục người dùng
- | Quyền đọc được biểu thị bằng "r", quyền ghi bằng "w", quyền thực thi bằng "x" và "-" nếu không có quyền

```
pi@raspberrypi:~ $ cd /home/pi
pi@raspberrypi:~ $ ls -l
total 36
-rw-r--r-- 1 pi pi    0 Aug 21  05:14 a.txt
drwxr-xr-x 2 pi pi 4096 May   7 23:52 Bookshelf
-rw-r--r-- 1 pi pi    0 Aug 21 19:50 b.txt
drwxr-xr-x 2 pi pi 4096 Aug 20  05:09 Desktop
drwxr-xr-x 2 pi pi 4096 May   8  00:07 Documents
drwxr-xr-x 2 pi pi 4096 May   8  00:07 Downloads
drwxr-xr-x 2 pi pi 4096 May   8  00:07 Music
drwxr-xr-x 2 pi pi 4096 May   8  00:07 Pictures
drwxr-xr-x 2 pi pi 4096 May   8  00:07 Public
drwxr-xr-x 2 pi pi 4096 May   8  00:07 Templates
drwxr-xr-x 2 pi pi 4096 May   8  00:07 Videos
pi@raspberrypi:~ $
```

## Làm thế nào để chỉ ra kiểm soát truy cập

rw-

r--

r--

Chủ sở hữu

nhóm

Người dùng khác

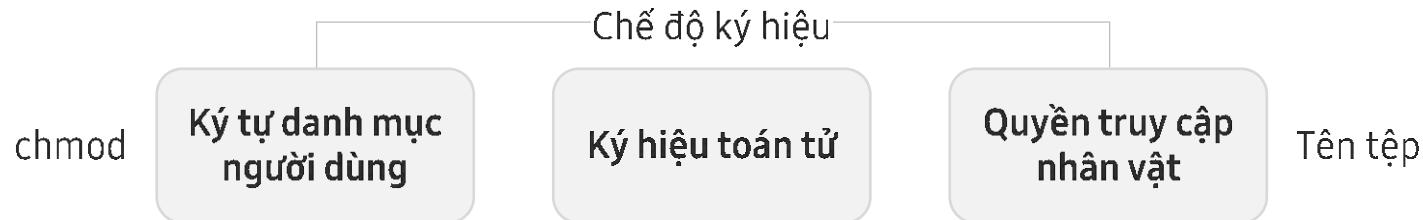
Quyền truy cập	Giải thích
rwxr-xr-x	Chủ sở hữu có quyền đọc, ghi và thực thi, trong khi các nhóm và người dùng khác chỉ có quyền đọc và thực thi.
r-xr-xr-x	Chủ sở hữu, nhóm và những người dùng khác đều có quyền đọc và thực thi.
rw-----	Chỉ chủ sở hữu mới có quyền truy cập đọc và ghi. Nhóm và những người dùng khác không có quyền nào cả.
rw-rw-rw-	Chủ sở hữu, nhóm và những người dùng khác đều có quyền đọc và ghi.
rwxrwxrwx	Chủ sở hữu, nhóm và những người dùng khác đều có quyền đọc, viết và thực thi.
rwx-----	Chỉ chủ sở hữu mới có quyền đọc, viết và thực thi. Nhóm và những người dùng khác không có quyền gì cả.
r-----	Chỉ chủ sở hữu mới có quyền đọc.

## chmod : Lệnh thay đổi kiểm soát truy cập

<b>chmod</b>	Chức năng	Thay đổi quyền truy cập vào tệp hoặc thư mục
	Mẫu	chmod [option] tập tin chế độ quyền hoặc tên thư mục
	Tùy chọn	R: Có thể thay đổi tất cả các thư mục con.

- | Chế độ ký hiệu: Hiển thị quyền sử dụng chữ cái và ký hiệu để thay đổi quyền truy cập
- | Chế độ số: Sử dụng số để thay đổi quyền truy cập

## chmod : Chế độ ký hiệu



Mục	Ký tự/ký hiệu	Giải thích
Ký tự danh mục người dùng	u	chủ sở hữu tập tin
	g	Nhóm chủ sở hữu thuộc về
	o	Những người dùng khác bên cạnh chủ sở hữu và nhóm
	a	Tất cả người sử dụng
Ký hiệu toán tử	+	Cho phép
	-	Xóa quyền
	=	Cài đặt quyền truy cập
Quyền truy cập nhân vật	r	đọc quyền
	w	viết quyền
	x	thực thi quyền

## chmod : Ví dụ về cài đặt kiểm soát truy cập bằng chế độ tương trưng

Quyền	Giải thích
u+w	Cấp(+) quyền viết (w) cho chủ sở hữu (u)
u-x	Xóa (-) quyền thực thi (x) của chủ sở hữu (u)
g+w	Cấp quyền (+) ghi (w) cho nhóm (g)
o-r	Xóa (-) quyền đọc (r) cho người dùng khác (o)
g+wx	Cấp quyền (+) ghi (w) và thực thi (x) cho nhóm (g)
a+rwx	Cấp quyền (+) đọc (r), viết (w) và thực thi (x) cho tất cả người dùng
u=rwx	Cấp (=) quyền đọc (r), viết (w), thực thi (x) cho chủ sở hữu (u)
go+w	Cấp quyền (+) ghi (w) cho nhóm (g) và những người dùng khác (o)
u+x.go+w	Cấp (+) quyền thực thi cho chủ sở hữu (u) và quyền ghi (w) cho nhóm (g) và những người dùng khác (o)

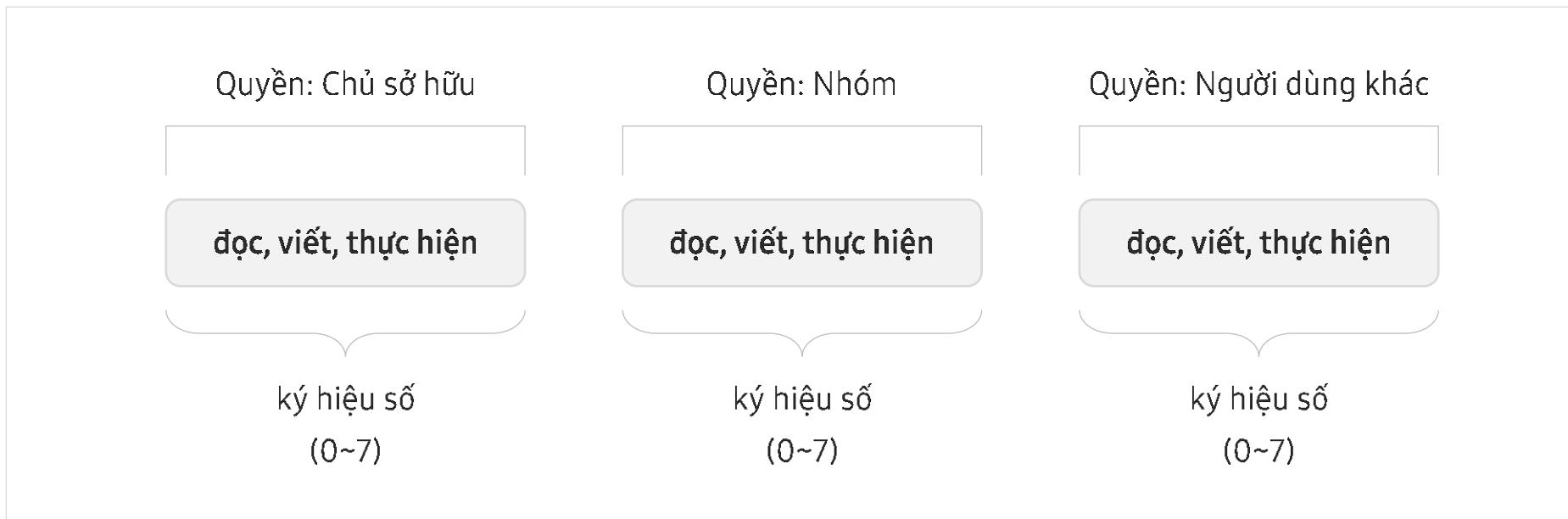
### chmod : Ví dụ về thay đổi quyền truy cập bằng chế độ tương trưng

- | Kiểm tra quyền truy cập hiện tại: rw-r--r--
- | Xóa quyền ghi của chủ sở hữu: u-w

```
pi@raspberrypi:~ $ ls -l b.txt
-rw-r--r-- 1 pi pi 0 Aug 21 19:50 b.txt
pi@raspberrypi:~ $ chmod u-w b.txt
pi@raspberrypi:~ $ ls -l b.txt
-r--r--r-- 1 pi pi 0 Aug 21 19:50 b.txt
pi@raspberrypi:~ $
```

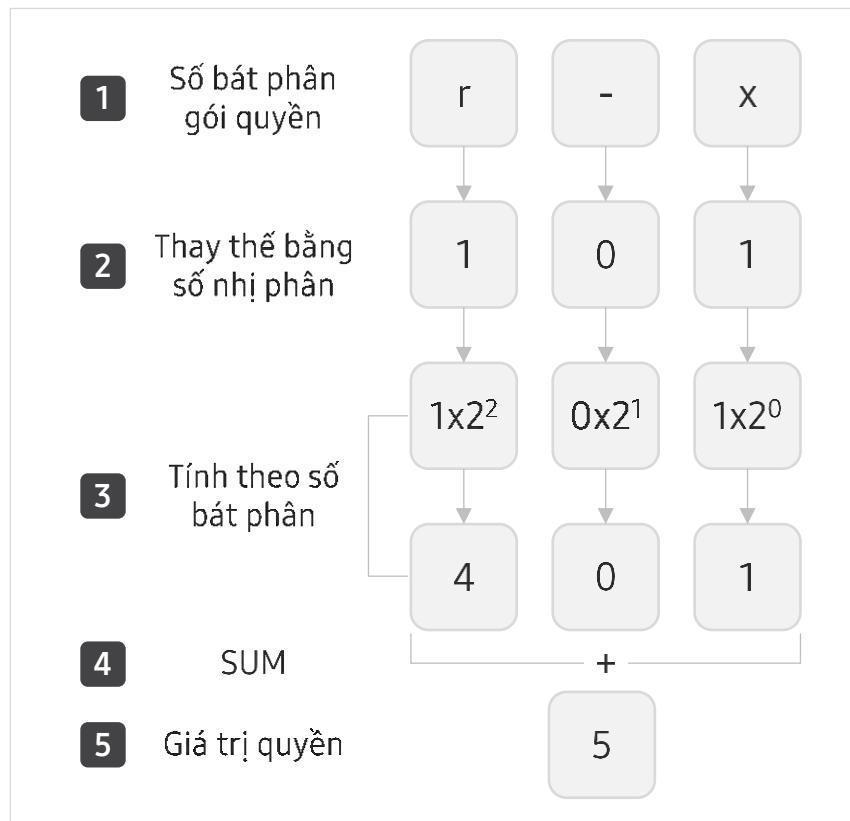
## chmod : Chế độ số

- | Ở chế độ số, mỗi quyền được biểu thị bằng 0 và 1, sau đó được chuyển đổi thành một số
- | Được biểu thị bằng 0 đến 7 theo sự kết hợp của quyền cho mỗi danh mục



## chmod : Quá trình chuyển đổi quyền thành số

### I Quá trình chuyển đổi quyền thành số



- Trong trường hợp r-x, chuyển đổi quyền thành 1, không có quyền thành 0.
- Chuyển đổi sang số nhị phân 1, 0, 1.
- Nếu bạn chuyển đổi một số nhị phân thành số thập phân cho mỗi chữ số, nó sẽ trở thành 4, 0, 1.
- Khi ba số được cộng lại với nhau, giá trị quyền cuối cùng là 5.

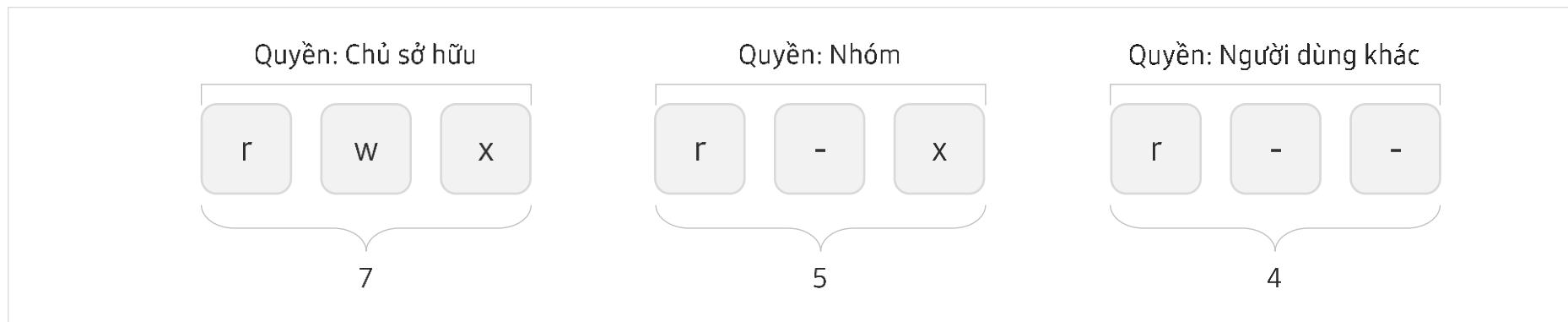
## chmod : Quá trình chuyển đổi quyền thành số

Mối tương quan giữa quyền truy cập và số lượng

Quyền truy cập	Trao đổi	Số	Sự định nghĩa
rwx	111 → 4+2+1	7	Đọc, Viết, Thực hiện
rw-	110 → 4+2+0	6	Đọc, Viết
r-x	101 → 4+0+1	5	Đọc và thực hiện
r--	100 → 4+0+0	4	Đọc
-wx	011 → 0+2+1	3	Viết, thực hiện
-w-	010 → 0+2+0	2	Viết
--x	001 → 0+0+1	1	Thực hiện
---	00 → 0+0+0	0	Không cho phép

## Đặt tất cả các quyền bằng số

| Tất cả các quyền truy cập được viết dưới dạng số



| Tất cả các quyền truy cập được viết dưới dạng số

Quyền truy cập	Chế độ số bát phân	Quyền truy cập	Chế độ số bát phân
rwxrwxrwx	777	rw-r--r--	644
rwxr-xr-x	755	rwx-----	700
rw-rw-rw	666	rw-r-----	640
r-xr-xr-x	555	r-----	400

## Thay đổi quyền truy cập bằng chế độ số

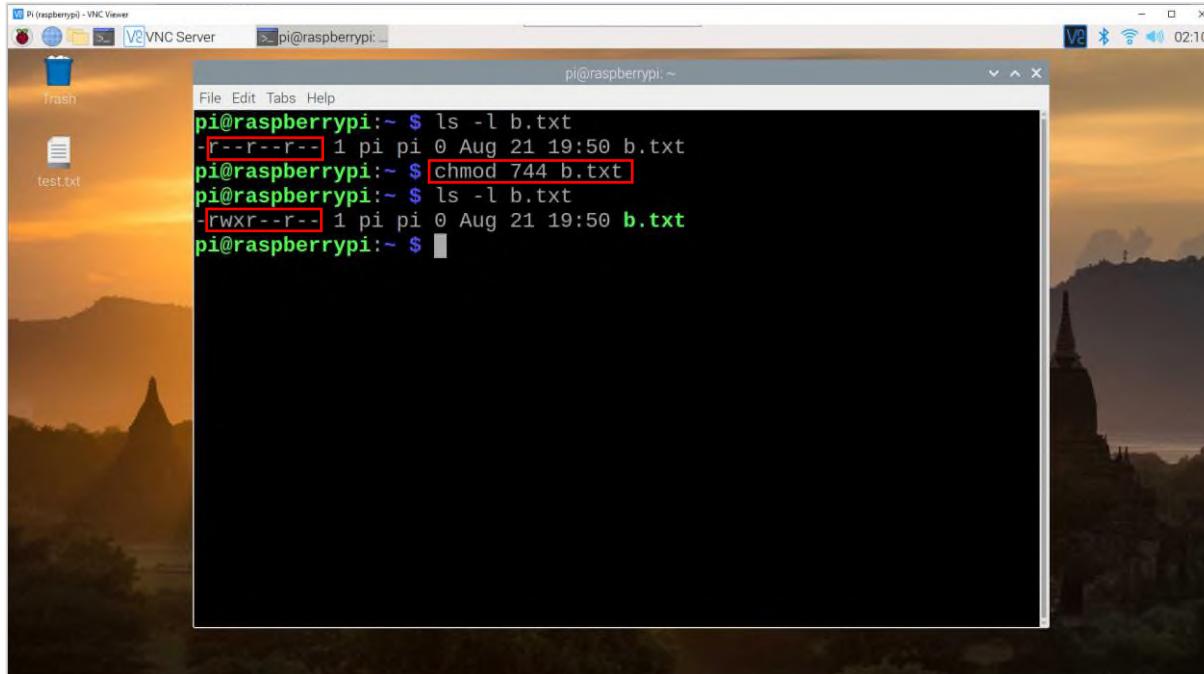
### Thay đổi quyền truy cập bằng chế độ số



- Không cần chỉ định danh mục người dùng vì mỗi vị trí trong số đại diện cho một danh mục người dùng..
- Vì bạn phải luôn sử dụng số có ba chữ số, nên bạn phải chỉ ra không chỉ quyền của danh mục người dùng mà còn cả quyền của các nhóm và người dùng khác.

### chmod : Thay đổi quyền truy cập bằng chế độ số (ví dụ)

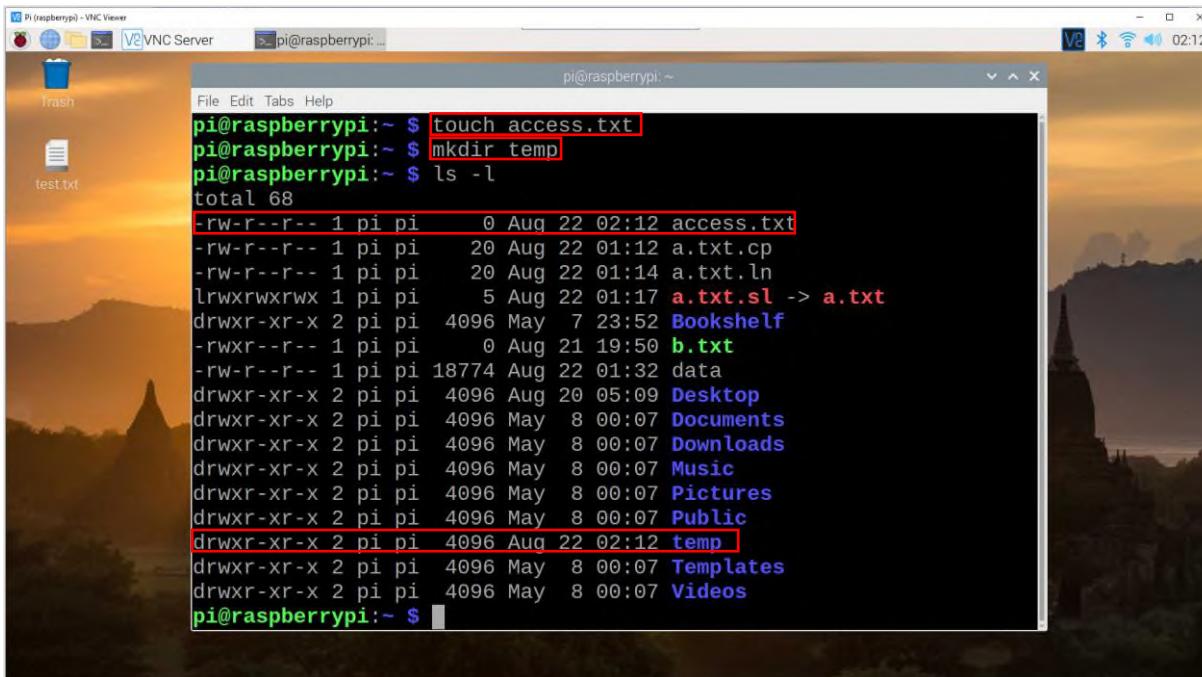
| Thay đổi quyền của chủ sở hữu : rwxr--r-- tương đương với 744



```
pi@raspberrypi:~ $ ls -l b.txt
-r--r--r-- 1 pi pi 0 Aug 21 19:50 b.txt
pi@raspberrypi:~ $ chmod 744 b.txt
pi@raspberrypi:~ $ ls -l b.txt
-rwxr--r-- 1 pi pi 0 Aug 21 19:50 b.txt
pi@raspberrypi:~ $
```

## Quyền truy cập cơ bản

Trên Linux, quyền truy cập mặc định được đặt tự động khi tạo tệp hoặc thư mục



```
pi@raspberrypi:~ $ touch access.txt
pi@raspberrypi:~ $ mkdir temp
pi@raspberrypi:~ $ ls -l
total 68
-rw-r--r-- 1 pi pi      0 Aug 22 02:12 access.txt
-rw-r--r-- 1 pi pi     20 Aug 22 01:12 a.txt.cp
-rw-r--r-- 1 pi pi     20 Aug 22 01:14 a.txt.ln
lrwxrwxrwx 1 pi pi      5 Aug 22 01:17 a.txt.sl -> a.txt
drwxr-xr-x 2 pi pi  4096 May  7 23:52 Bookshelf
-rw-r--r-- 1 pi pi      0 Aug 21 19:50 b.txt
-rw-r--r-- 1 pi pi 18774 Aug 22 01:32 data
drwxr-xr-x 2 pi pi  4096 Aug 20 05:09 Desktop
drwxr-xr-x 2 pi pi  4096 May  8 00:07 Documents
drwxr-xr-x 2 pi pi  4096 May  8 00:07 Downloads
drwxr-xr-x 2 pi pi  4096 May  8 00:07 Music
drwxr-xr-x 2 pi pi  4096 May  8 00:07 Pictures
drwxr-xr-x 2 pi pi  4096 May  8 00:07 Public
drwxr-xr-x 2 pi pi  4096 Aug 22 02:12 temp
drwxr-xr-x 2 pi pi  4096 May  8 00:07 Templates
drwxr-xr-x 2 pi pi  4096 May  8 00:07 Videos
pi@raspberrypi:~ $
```

## umask : Kiểm tra và thay đổi quyền truy cập mặc định

<b>umask</b>	Chức năng	Cho phép một người xem hoặc đặt mask tạo chế độ tệp (mask: quyền cho các tệp mới được tạo)
	Định dạng	umask [option] [Mask value]
	Tùy chọn	-S : in giá trị mask dưới dạng ký tự
	Ví dụ	umask 022      umask

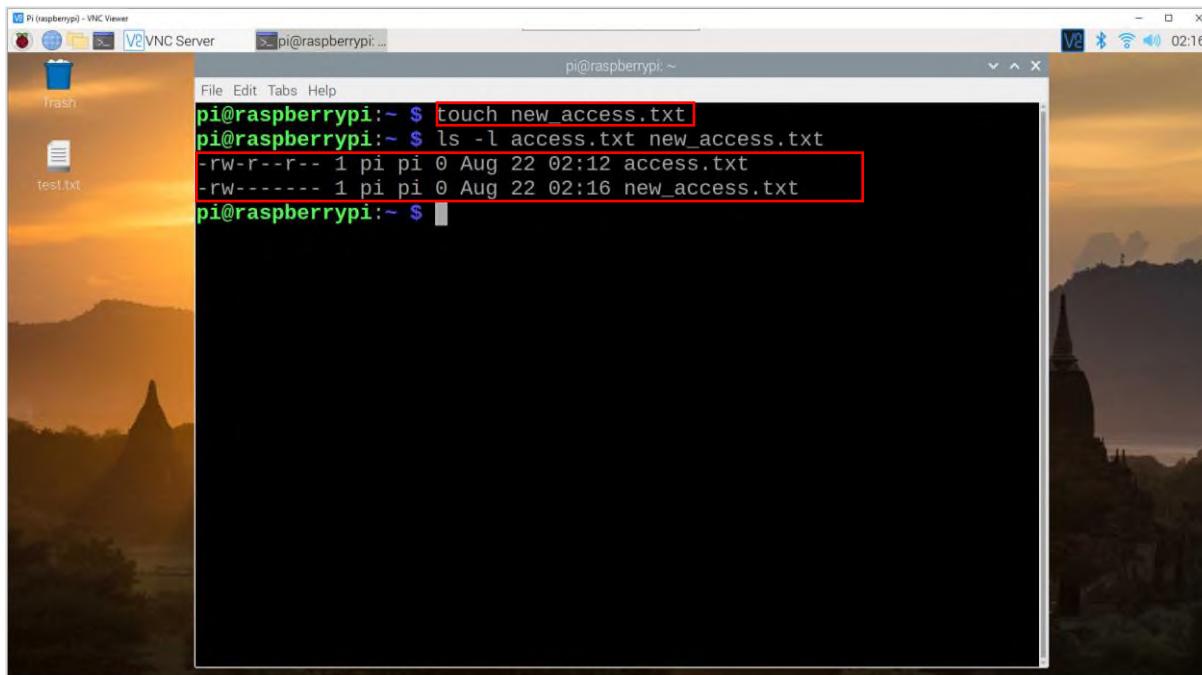
## Định nghĩa các mask values

- █ Mask values chỉ định các quyền không được cấp khi tạo tệp hoặc thư mục
- █ Mask values là 022 có nghĩa là ---- w--w-. Nó chỉ ra rằng nhóm và những người dùng khác không được cấp quyền ghi
- █ Thay đổi umask thành 077 và kiểm tra umask

```
pi@raspberrypi:~ $ umask  
0022  
pi@raspberrypi:~ $ umask -S  
u=rwx,g=rx,o=rx  
pi@raspberrypi:~ $ umask 077  
pi@raspberrypi:~ $ umask -S  
u=rwx,g=,o=  
pi@raspberrypi:~ $
```

## umask - thay đổi các giá trị mask

- | Thay đổi mask values cũng thay đổi quyền truy cập mặc định được áp dụng khi tạo tệp hoặc thư mục
- | Access.rxt: rw-r--r– được tạo dưới dạng umask 022
- | New\_access.txt: rw----- được tạo dưới dạng umask 077



```
pi@raspberrypi:~ $ touch new_access.txt
pi@raspberrypi:~ $ ls -l access.txt new_access.txt
-rw-r--r-- 1 pi pi 0 Aug 22 02:12 access.txt
-rw----- 1 pi pi 0 Aug 22 02:16 new_access.txt
pi@raspberrypi:~ $
```

## Quá trình áp dụng các giá trị mặt nạ



Bài 2.

## Hướng dẫn lệnh hệ điều hành Raspberry

- | 2.1. Kết nối với hệ điều hành Raspberry Pi từ máy tính  
của bạn
- | 2.2. Lệnh cơ bản
- | 2.3. Các lệnh về Hệ thống tệp
- | 2.4. Các lệnh về điều khiển hệ thống
- | 2.5. Các lệnh về Network
- | 2.6. Các lệnh về Người dùng & Quyền
- | **2.7. Công cụ lập trình**

## Công cụ lập trình cho Linux

| Chúng ta sẽ học gì trong bài này

- ▶ Tìm hiểu cách lập trình trong Linux khác với các môi trường khác như Windows khi nói đến cách biên dịch, gỡ lỗi hoặc chạy mã nguồn của bạn.
- ▶ Tìm hiểu cách biên dịch, xây dựng tự động và gỡ lỗi khi lập trình trong Linux, cùng với hiểu biết về gcc, make, makefile và gdb.

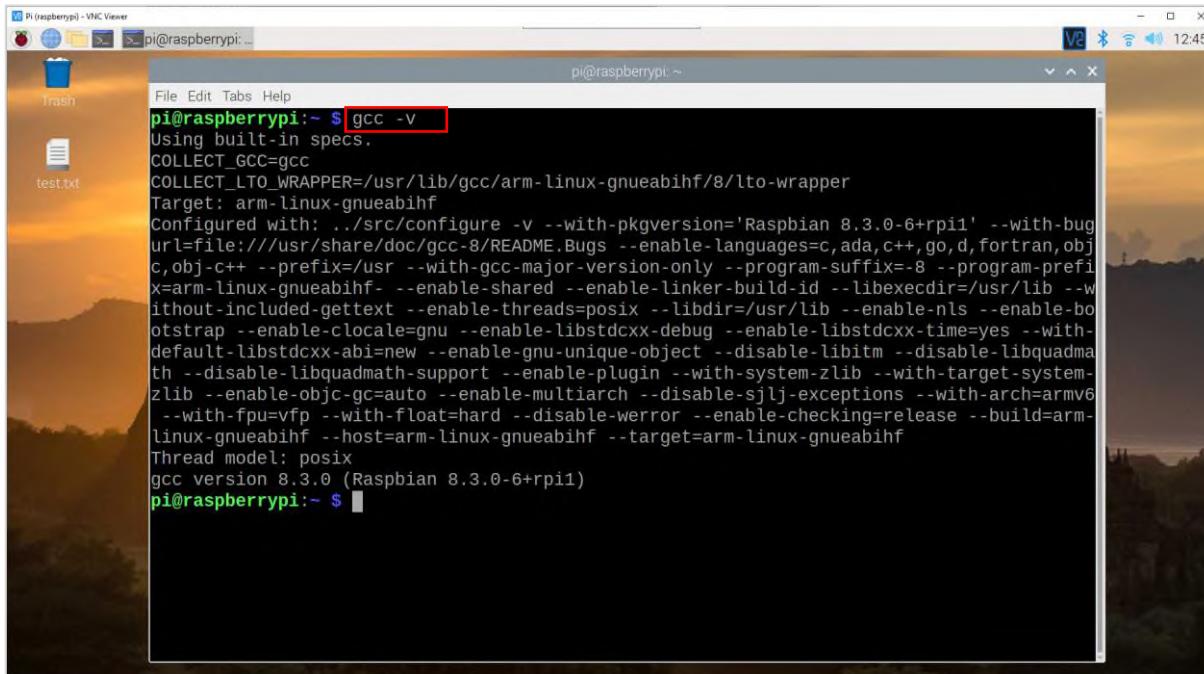
### gcc

#### I gcc là gì?

- ▶ Nó là một Bộ sưu tập trình biên dịch mã nguồn mở từ Dự án GNU. Nó được tạo ra vào năm 1987 bởi Richard Stallman, và nó thực sự là một trình biên dịch tiêu chuẩn cho các nền tảng dựa trên UNIX/Linux.
- ▶ Là trình biên dịch chính thức của hệ thống GNU, nó đã được sử dụng để tạo ra nhiều trình biên dịch và hệ điều hành.

## gcc

gcc -v

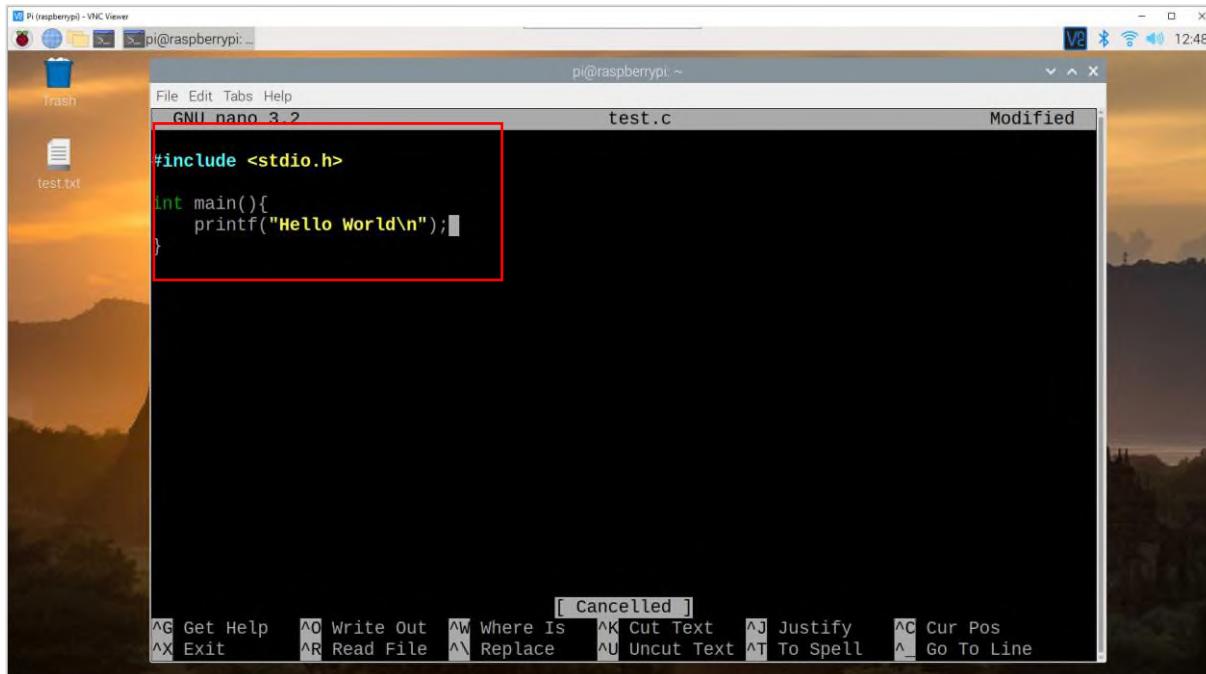


```
pi@raspberrypi:~ $ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/arm-linux-gnueabihf/8/lto-wrapper
Target: arm-linux-gnueabihf
Configured with: ../src/configure -v --with-pkgversion='Raspbian 8.3.0-6+rpi1' --with-bug
url=file:///usr/share/doc/gcc-8/README.Bugs --enable-languages=c,ada,c++,go,d,fortran,obj
c,obj-c++ --prefix=/usr --with-gcc-major-version-only --program-suffix=-8 --program-prefix=arm-linux-gnueabihf- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --w
ithout-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --enable-bo
otstrap --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-
default-libstdcxx-abi=new --enable-gnu-unique-object --disable-libitm --disable-libquadma
th --disable-libquadmath-support --enable-plugin --with-system-zlib --with-target-system-
zlib --enable-objc-gc=auto --enable-multiarch --disable-sjlj-exceptions --with-arch=armv6
--with-fpu=vfp --with-float=hard --disable-werror --enable-checking=release --build=arm-
linux-gnueabihf --host=arm-linux-gnueabihf --target=arm-linux-gnueabihf
Thread model: posix
gcc version 8.3.0 (Raspbian 8.3.0-6+rpi1)
pi@raspberrypi:~ $
```

- ▶ Sử dụng để kiểm tra phiên bản gcc.
- ▶ Gcc được cài đặt theo mặc định mà không cần cài đặt riêng.

## gcc

### Tạo test.c để sử dụng gcc



```
#include <stdio.h>
int main(){
    printf("Hello World\n");
}
```

- ▶ Tạo test.c bằng cách sử dụng lệnh nano test.c.

```
$ nano test.c
```

- ▶ Viết mã như nhìn thấy ở bên trái.
- ▶ Sau khi lưu theo thứ tự Ctrl + x → Chèn y → Enter, thoát khỏi nano.
- ▶ Nano là một trình soạn thảo văn bản được hỗ trợ bởi Linux. Nó sẽ được đề cập nhiều hơn trong Chương 4.

## gcc

```
gcc -o test test.c
```

```
pi@raspberrypi:~ $ gcc -o test test.c
pi@raspberrypi:~ $ ls -al test
-rwxr-xr-x 1 pi pi 7980 Aug 23 12:51 test
pi@raspberrypi:~ $ ./test
Hello World
pi@raspberrypi:~ $
```

Chỉ định xuất ra file

Tên file output

- Tạo tệp đối tượng có tên test bằng cách biên dịch tệp test.c.

```
$ gcc -o test test.c
```

- Sau khi tạo tệp đối tượng, bài kiểm tra là tệp thực thi và có màu xanh lục.
- Nếu bạn chạy bằng cách sử dụng lệnh ./test, bạn có thể thấy kết quả của mã "Hello World".

## make

### Hình thức make file

target: prerequisites recipe

- ▶ Ý nghĩa của từng phần
  - target – Tệp được in tùy thuộc vào điều kiện ở phần sau và công thức
  - prerequisites – Tệp đầu vào được thêm vào để tạo tệp in
  - recipe – Thể hiện cách biên dịch và liên kết với tệp đầu vào

Tham khảo: <https://www.youtube.com/watch?v=FIIH8zfayXI>

## make

### Hiểu với một ví dụ

- ▶ Giả sử có 2 tệp nguồn .c và tệp tiêu đề .h với cấu trúc thư mục như sau.

```
src/  
  - main.c  
  - source.c  
  
include/  
  - main.h  
  - source.h
```

## make

### I main.c và main.h

#### ▶ main.c

```
#include "main.h"
#include "source.h"

int main(int argc, char* argv[]) {
    sayHello("Hello, World");
    return 0;
}
```

#### ▶ main.h

```
#ifndef _MAIN_H
#define _MAIN_H
#include <stdio.h>
#endif
```

## make

### | source.c và source.h

#### ▶ source.c

```
#include "source.h"

void sayHello(const char* str) {
    printf("sayHello
Function: %s\n", str);
}
```

#### ▶ source.h

```
#ifndef _SOURCE_H
#define _SOURCE_H
#include "main.h"
#endif
```

## make

| Để xây dựng 4 tệp, có thể sử dụng lệnh sau:

- ▶ \$ gcc src/\*.c -o program -include
- ▶ Có một số vấn đề nếu bạn xây dựng chương trình của mình bằng phương pháp hàng trên. Các lệnh biên dịch phức tạp phải được nhập không có lỗi chính tả như trên. Khi triển khai một dự án, bên thứ ba có thể khó xác định cách xây dựng dự án.
- ▶ Vì lý do này, bạn nên tự động hóa quá trình xây dựng. Bạn chỉ cần viết tập lệnh một lần và có thể sử dụng make để xây dựng tập lệnh đó.

## makefile

- | Viết makefile liên quan đến 4 tệp

```
CC = gcc
TARGET = program

SRC_DIR = src
INCLUDE_DIR = include
BINARY_DIR = bin

OBJECTS = $(patsubst %.c, %.o, $(wildcard $(SRC_DIR)/*.c))
HEADERS = $(wildcard $(INCLUDE_DIR)/*.h)

$(TARGET): $(OBJECTS)
    @echo "[+] Make Binary File"
    $(CC) -o $(BINARY_DIR)/$@ $(OBJECTS);

%.o: %.c $(HEADERS)
    @echo "[+] Compile $@ File"
    $(CC) -g -c -o $@ $< -I$(INCLUDE_DIR)

clean:
    @rm -f $(SRC_DIR)/*.o
    @rm -f $(BINARY_DIR)/$(TARGET)
    @rmdir $(BINARY_DIR)
```

Tạo mới một thư  
mục “bin” để chứa  
chương trình

## makefile

### Giải thích bổ sung về định dạng của makefile

- ▶ Khi sử dụng biến, hãy truy cập như \$(variable).
- ▶ Trong các khu vực như %.o, %.c, % có nghĩa là chuỗi đó là chuỗi nào không quan trọng.
  - Ví dụ: nếu tệp a.o, b.o là cần thiết, tất cả những gì bạn cần làm là viết %.o.
- ▶ \$@ là một loại biến macro đề cập đến tệp in.
- ▶ \$< là một loại biến vĩ mô đề cập đến tệp đầu vào.

## makefile

### I Các hàm hữu ích do tiện ích makefile cung cấp

#### ▶ subst

- Được sử dụng ở định dạng \$(subst from, to, text).
- Chuyển đổi các chuỗi thuộc về 'from' thành 'to' trong số các chuỗi được gọi là text.

#### ▶ wildcard

- Được sử dụng ở định dạng \$(mẫu wildcard).
- Được sử dụng để lấy đường dẫn của các tệp khớp với mẫu.
- Chẳng hạn nếu nó được viết là \$(wildcard ../\*.c),

Nó đề cập đến tất cả các tệp khớp với các mẫu như ../target/main.c và ../target/source.c.

## makefile

### I Các hàm hữu ích do tiện ích makefile cung cấp

#### ▶ patsubst

- Được sử dụng ở định dạng như \$(mẫu patsubst, replacement, text).
- Thay thế các chuỗi mẫu phù hợp với mẫu bằng chuỗi mẫu thay thế giữa các chuỗi được gọi là text.
- Chẳng hạn, nếu nó được viết là as \$(patsubst %.c, %.o, foo.c.c bar.c),  
'foo.c.c bar.c' thay đổi thành 'foo.c.o bar.o'.

#### ▶ notdir

- Được sử dụng ở định dạng như \$(tên notdir...).
- Chỉ in 'tên tệp' bằng cách xóa 'đường dẫn tệp' trong số các chuỗi được liệt kê.

## gdb

### I gdb là gì?

- ▶ Trình gỡ lỗi cơ bản cho hệ thống phần mềm GNU
- ▶ Mục đích của trình sửa lỗi như GDB là để cho bạn thấy điều gì đang xảy ra trong chương trình của bạn khi các chương trình khác đang chạy hoặc khi chương trình của bạn được thực thi không chính xác.
- ▶ GDB có thể ra mắt chương trình được tạo bằng C, C++, Modula-2.

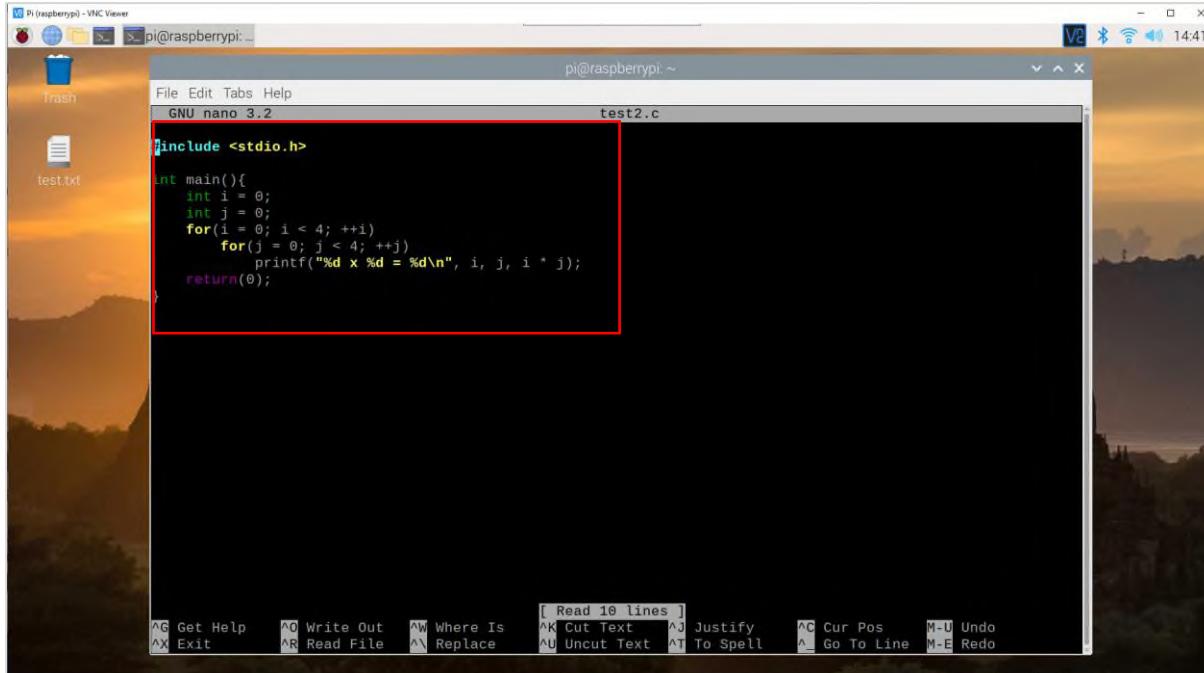
## gdb

### | Khởi động gdb

- ▶ % gdb: Nhập chương trình với lệnh gọi là tệp sau khi thực hiện gdb trước.
- ▶ % gdb program : Phương pháp chung.
- ▶ % gdb program core: Đưa ra đối số đồng thời khi sử dụng tệp lỗi.
- ▶ % gdb program abc: Để gỡ lỗi một chương trình đang chạy, bạn có thể chỉ định ID tiến trình làm đối số thứ hai. Lệnh này kết nối gdb (nếu không có tệp có tên 'abc') để xử lý abc. (gdb tìm tệp lỗi trước.)
- ▶ % gcc -g test.c -o test
- ▶ % gdb test

## gdb

### Hiểu và thực hành gdb với một ví dụ đơn giản



```
#include <stdio.h>
int main(){
    int i = 0;
    int j = 0;
    for(i = 0; i < 4; ++i)
        for(j = 0; j < 4; ++j)
            printf("%d x %d = %d\n", i, j, i * j);
    return(0);
}
```

- ▶ Thực hiện lệnh nano test2.c để tạo tệp có tên test2.c và viết mã nguồn.
- ▶ Lưu theo thứ tự Ctrl + x → Chèn y → Nhập và thoát khỏi nano.

## gdb

### Hiểu và thực hành gdb với một ví dụ đơn giản

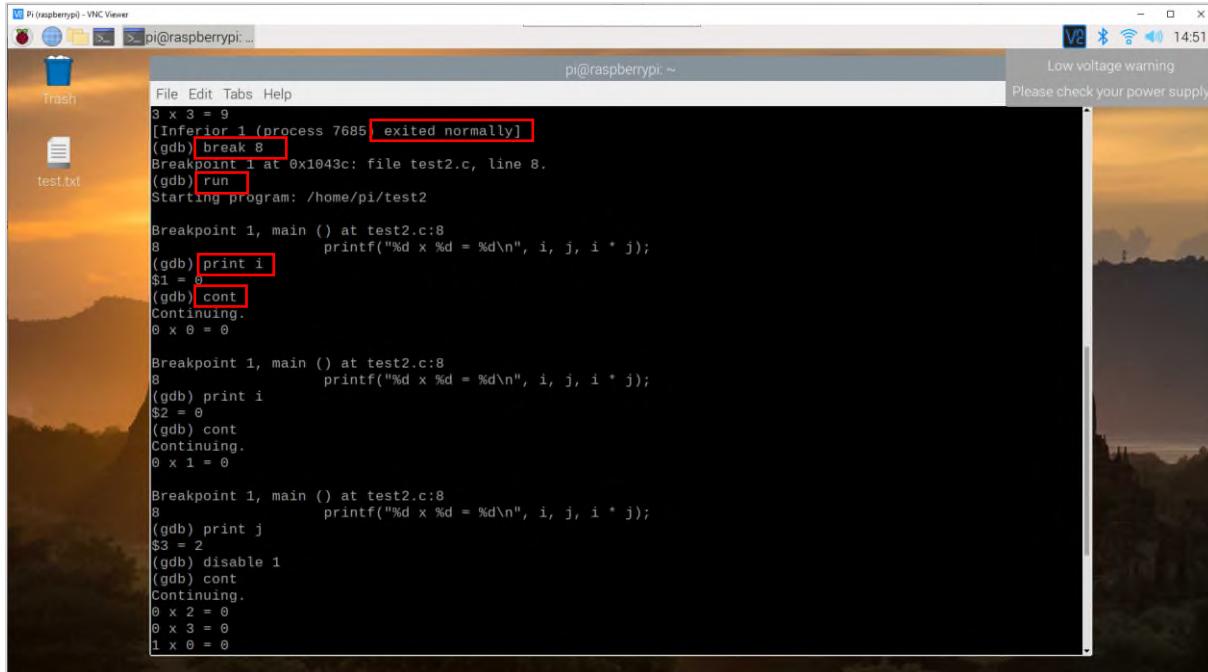
```
pi@raspberrypi:~ $ nano test2.c
pi@raspberrypi:~ $ gcc -g -o test2 test2.c
pi@raspberrypi:~ $ gdb test2
GNU gdb (Raspbian 8.2.1-2) 8.2.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "arm-linux-gnueabihf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from test2...done.
(gdb) run
Starting program: /home/pi/test2
0 x 0 = 0
0 x 1 = 0
0 x 2 = 0
0 x 3 = 0
1 x 0 = 0
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
2 x 0 = 0
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
```

- ▶ Thực thi lệnh `gcc -g -o test2 test2.c` nhưng để chèn thông tin gỡ lỗi, hãy thêm tùy chọn `-g` và biên dịch bằng lệnh này.
- ▶ Thực hiện lệnh `gdb test_2` để gỡ lỗi `test_2`.
- ▶ `gdb` bắt đầu gỡ lỗi. Tại thời điểm này, thực hiện lệnh chạy.

## gdb

### Hiểu và thực hành gdb với một ví dụ đơn giản



A screenshot of a VNC session on a Raspberry Pi. The terminal window shows a gdb session running on the command line. The session starts with a breakpoint at line 8 of test2.c. It then prints the value of variable i, which is 0. After continuing, it prints the value of variable j, which is also 0. Finally, it disables the breakpoint and continues, showing the result of the multiplication (0 x 3 = 0). A desktop environment is visible in the background, including icons for Trash and test.txt, and a power warning message: "Low voltage warning Please check your power supply".

```
pi@raspberrypi: ~
[Inferior 1 (process 7685) exited normally]
(gdb) break 8
Breakpoint 1 at 0x1043c: file test2.c, line 8.
(gdb) run
Starting program: /home/pi/test2

Breakpoint 1, main () at test2.c:8
8          printf("%d x %d = %d\n", i, j, i * j);
(gdb) print i
$1 = 0
(gdb) cont
Continuing.
0 x 0 = 0

Breakpoint 1, main () at test2.c:8
8          printf("%d x %d = %d\n", i, j, i * j);
(gdb) print i
$2 = 0
(gdb) cont
Continuing.
0 x 1 = 0

Breakpoint 1, main () at test2.c:8
8          printf("%d x %d = %d\n", i, j, i * j);
(gdb) print j
$3 = 2
(gdb) disable 1
(gdb) cont
Continuing.
0 x 2 = 0
0 x 3 = 0
1 x 0 = 0
```

- ▶ Nó hiển thị thoát bình thường không có lỗi.
- ▶ Để thêm một điểm ngắt, sử dụng cùng loại lệnh như ngắt n, trong đó n là số dòng.
- ▶ Nếu bạn tạo một điểm ngắt và chạy lại, nó sẽ dừng ở điểm dừng đầu tiên mà bạn gắp.
- ▶ Tại thời điểm này, có thể theo dõi biến để gỡ lỗi, chẳng hạn như xuất giá trị của biến.
- ▶ Nếu muốn chạy lại sau breakpoint thì nhập cont.

## gdb

### Hiểu và thực hành gdb với một ví dụ đơn giản

```
pi@raspberrypi: ~
Breakpoint 1, main () at test2.c:8
8         printf("%d x %d = %d\n", i, j, i * j);
(gdb) print i
$2 = 0
(gdb) cont
Continuing.
0 x 1 = 0

Breakpoint 1, main () at test2.c:8
8         printf("%d x %d = %d\n", i, j, i * j);
(gdb) print j
$3 = 0
(gdb) disable 1
(gdb) cont
Continuing.
0 x 2 = 0
0 x 3 = 0
1 x 0 = 0
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
2 x 0 = 0
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
3 x 0 = 0
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
[Inferior 1 (process 7688) exited normally]
(gdb) quit
pi@raspberrypi: ~
```

- ▶ Khi bạn gặp điểm ngắt một lần nữa và lần này j được xuất ra, bạn có thể thấy rằng giá trị biến được phản ánh tốt theo số vòng lặp.
- ▶ Bạn có thể vô hiệu hóa các điểm dừng bằng lệnh vô hiệu hóa n, trong đó n là số điểm dừng, không phải số dòng. Số điểm dừng là số ngay sau đầu ra điểm dừng dưới dạng từ đầu tiên khi sử dụng lệnh break n.
- ▶ Nếu bạn muốn kết thúc gỡ lỗi khi nó đã thoát bình thường, hãy thoát bằng thoát.

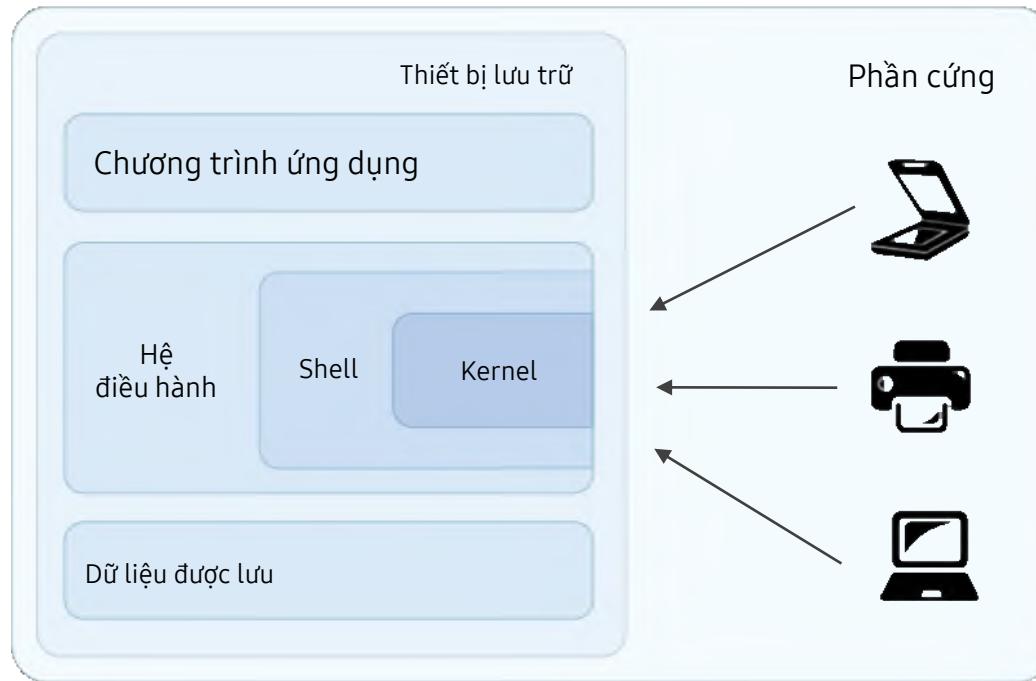
Bài 3

# Lập trình Shell

## | 3.1. Khái quát về Shell

## Shell là gì?

- Linux' Shell là một giao diện được dùng để thực thi lệnh và cho phép hệ điều hành thực hiện
- Shell hoạt động như cầu nối giữa một kernel và một người dùng. Nhận lệnh từ người dùng, thông dịch lệnh và chạy chương trình.



## Chức năng của Shell: Một trình thông dịch lệnh

- | Shell thông dịch và chuyển lệnh giữa người dùng và kernel.
- | Sau khi người dùng đăng nhập, Shell tự động chạy và chờ người dùng nhập một lệnh vào → Shell đăng nhập
- | Shell đăng nhập được gán cho tệp /etc/passwd cho mỗi người dùng.
- | Dấu nhắc: Thông báo Shell đang chờ lệnh của người dùng.

## Chức năng của Shell: Lập trình

- | Shell có một chức năng lập trình nội bộ cho phép người dùng chạy chuỗi lệnh như một lệnh đơn, dẫn đến tự động hóa các thao tác thường được dùng
- | Chương trình Shell là chỉ tập lệnh Shell.

## Chức năng của Shell: Cấu hình người dùng

- | Cung cấp tệp khởi tạo để thiết lập cấu hình người dùng
- | Tệp khởi tạo thiết lập nhiều biến môi trường như đặt đường dẫn cho một lệnh, hoặc đặt quyền cơ bản cho việc tạo tệp tin và thư mục.

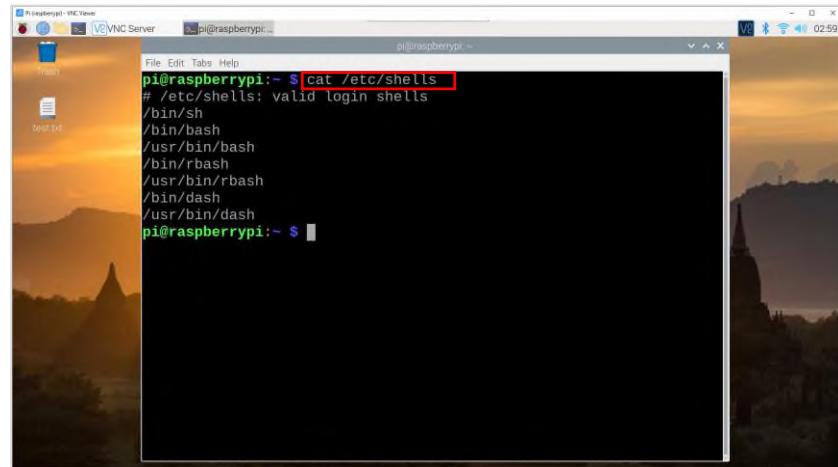
## Kiểu Shell

| Dưới đây là lược đồ về các kiểu Shell.

Tên Shell	Tên lệnh Shell	Nhà phát triển	Ngày ra mắt
Bourne Shell	sh	Stephen Bourne	1977
C Shell	csh	Bill Joy	1978
Korn Shell	ksh	David Korn	1983
bash Shell	bash	Brain Fox	1989
dash Shell	dash	Herbert Xu	Áp dụng cho Linux 1977

## chsh : Thay đổi Shell cơ bản

<b>umask</b>	Chức năng	Allows one to view or set the file mode creation mask (mask: quyền cho các tệp mới được tạo)
	Định dạng	unmask [option] [Mask value]
	Lựa chọn	-s shell: Thay đổi Shell đăng nhập thành Shell được gán(Đường dẫn tuyệt đối)
	Ví dụ	umask 022      umask



```
pi@raspberrypi:~ $ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
pi@raspberrypi:~ $
```

Kiểu Shell có thể thay đổi được gán cho /etc/shells.

## chsh : Thay đổi Shell cơ bản

- Chỉ định Shell để thay đổi làm một đường dẫn tuyệt đối.
- Thêm một trong các chi tiết của cat /etc/shells vào phần sau -\$

```
pi@raspberrypi:~ $ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
pi@raspberrypi:~ $ chsh -s /bin/sh pi
Password:
pi@raspberrypi:~ $
```

## Shell đăng nhập và sub Shell

- | Sub Shell là một shell có thể chạy một Shell khác trong một dấu nhắc.
- | Sub Shell có thể tạo một sub Shell khác.
- | Lệnh thoát sub Shell là ^d( +d), exit, v.v.
- | Sau khi hoàn thiện sub Shell, quay về cấu hình Shell trước đã chạy sub Shell.

## Lệnh dựng sẵn của Shell

- | Shell có một lệnh dựng sẵn.
- | Lệnh dựng sẵn của Shell không có một tệp exe riêng biệt. Lệnh nằm trong Shell.
  - ▶ Ví dụ lệnh: cd trong Shell

## in ra lệnh của Bash Shell

<b>echo</b>	Chức năng	Hiển thị một dòng chuỗi trên màn hình
	Định dạng	echo [-n [String...]]
	Lựa chọn	Không thay đổi dòng ở cuối
	Ví dụ	echo echo text echo -n text

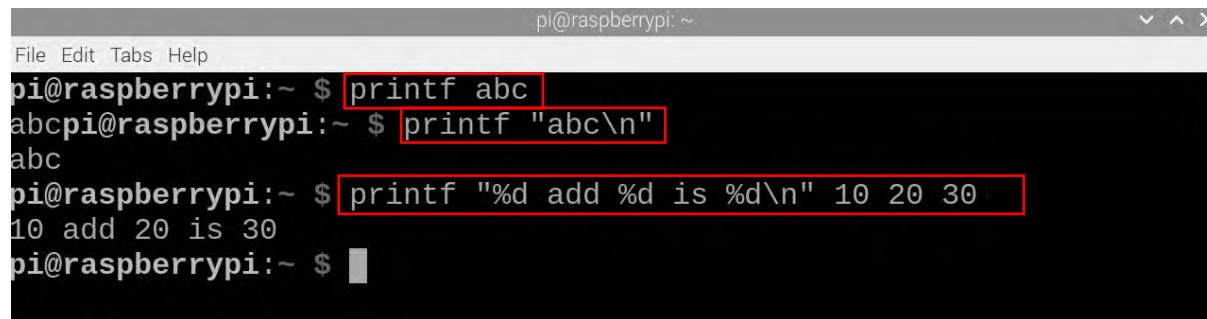
The screenshot shows a terminal window titled "pi@raspberrypi: ~". The window has a menu bar with "File", "Edit", "Tabs", and "Help". The terminal content is as follows:

```
pi@raspberrypi:~ $ echo linux
Linux
pi@raspberrypi:~ $ echo "linux raspberry pi"
Linux raspberry pi
pi@raspberrypi:~ $
```

The first command "echo linux" and its output "Linux" are highlighted with a red box. The second command "echo \"linux raspberry pi\"" and its output "Linux raspberry pi" are also highlighted with a red box.

## in ra lệnh của Bash Shell

<b>printf</b>	Chức năng	In ra chuỗi đã được định dạng
	Định nghĩa	printf Format [factor...]
	Option	Gán định dạng hàm printf của ngôn ngữ C như %d, \n, v.v.
	Ví dụ	printf text printf "text\n" printf "%d\n" 100



A screenshot of a terminal window titled "pi@raspberrypi: ~". The window shows the following command-line interactions:

```
pi@raspberrypi:~ $ printf abc
abc
pi@raspberrypi:~ $ printf "abc\n"
abc
pi@raspberrypi:~ $ printf "%d add %d is %d\n" 10 20 30
10 add 20 is 30
pi@raspberrypi:~ $
```

The first two commands, "printf abc" and "printf "abc\n"" are highlighted with red boxes. The third command, "printf "%d add %d is %d\n" 10 20 30" is also highlighted with a red box.

## Sử dụng ký tự đặc biệt

- | Cung cấp nhiều ký tự đặc biệt để người dùng có thể nhập và thực thi lệnh một cách thuận tiện.
- | Những ký tự đặc biệt cơ bản là \*, ?, |, ;, [], ~, ‘’, “”, ^~, v.v.
- | Sau khi nhập một lệnh, Shell sẽ kiểm tra liệu có ký tự đặc biệt nào không, đổi chúng sang định dạng hợp lệ và thực thi lệnh.
- | \*(dấu sao) → Một ký tự đặc biệt thể hiện một chuỗi ngẫu nhiên. Được thay thế bằng một ký tự lớn hơn số không.
- | Dưới đây là bảng ví dụ về sử dụng \*.

Ví dụ	Giải thích
ls *	Liệt kê toàn bộ tệp tin và thư mục con trong thư mục hiện hành. In ra nội dung của thư mục con.
cp * /tmp	Sao chép toàn bộ tệp tin trong thư mục hiện hành vào thư mục /tmp.
ls-F t*	Liệt kê tên và kiểu của toàn bộ tệp tin có tên bắt đầu bằng 't' như t. tmp. temp. t có nghĩa là bao gồm 't'.
cp *.txt../ch3	Sao chép toàn bộ tệp tin có phần mở rộng txt sang thư mục ch3 trong thư mục high.
ls-l h*d	Liệt kê thông tin chi tiết về tất cả các tệp bắt đầu bằng 'h' và kết thúc bằng 'd'. Bạn có thể xem thông tin về toàn bộ tệp tin m, như hd. had. hard. H12345d, etc.

## Ký tự đặc biệt ? Và [ ]

- | Được dùng để biểu thị ký tự đơn.
- | ? biểu thị một ký tự ngẫu nhiên của length '1'. [ ] biểu thị một trong các ký tự được cho vào trong dấu ngoặc.

Ví dụ	Giải thích
ls t*.txt	Một ký tự ngẫu nhiên đứng sau 't' và liệt kê tên của toàn bộ tệp tin với phần mở rộng txt. Bao gồm t1.txt, t2.txt, ta.txt, v.v. trừ t.txt
ls-l tmp[135].txt	Một ký tự trong 1, 3, 5 đứng sau tmp và liệt kê tên của toàn bộ tệp tin với phần mở rộng txt. Liệt kê thông tin chi tiết về tmp1.txt, tmp3.txt., tmp5.txt.
ls-l tmp[1-3].txt	[1-3] nghĩa là một khoảng từ 1 đến 3. Do đó, kết quả tương tự như ls -l tmp [123] .txt. Liệt kê thông tin chi tiết về tệp tmp1.txt, tmp2.txt, và tmp3.txt.
ls [0-9]*	Liệt kê toàn bộ tệp tin có tên bắt đầu bằng số.
ls [A-Za-z]*[0-9]	Liệt kê các tệp có tên bắt đầu bằng chữ cái.

## Ký tự đặc biệt ~ và -

- || ~ và - là ký tự đặc biệt thể hiện một thư mục.
  - | Khi chỉ dùng ~, thư mục chính của người dùng trong thao tác hiện tại sẽ được hiển thị.
  - | Khi được dùng với ID đăng nhập, thư mục chính của người dùng sẽ được hiển thị.
- || Dưới đây là bảng ví dụ sử dụng ~ và -

Ví dụ	Giải thích
cp *.txt~/ch3	Sao chép toàn bộ tệp tin có phần mở rộng txt sang thư mục tmp trong thư mục chính của người dùng hiện đang hoạt động.
cp ~user2/linux.txt	Tìm linux.txt trong thư mục chính của user2 và sao chép sang thư mục hiện hành.
Cd -	Thay đổi sang thư mục trước trong thao tác.

## Ký tự đặc biệt ; và |

| ; thực hiện lệnh liên tục theo thứ tự từ bên trái

| | chuyển kết quả của lệnh bên trái sang phần nhập của lệnh bên phải (| cũng tương tự shift + \)

Ví dụ	Giải thích
date; ls; pwd	Thực hiện lệnh theo thứ tự từ bên trái. Có nghĩa là, sau khi xuất ra ngày, nó sẽ xuất ra một danh sách các tệp và thư mục hiện hành. Sau đó, hiển thị đường dẫn tuyệt đối của thư mục hiện hành.
ls -al /   more	Xuất ra thông tin chi tiết của toàn bộ tệp tin trong thư mục gốc theo từng màn hình. Kết quả của lệnh ls -al / được chuyển cho phần nhập vào của lệnh ‘more’ và sau đó được xuất ra theo từng trang.

## Ký tự đặc biệt ‘ ’ và “ ”

- ‘ ’ và “ ” bao quanh ký tự và biến chúng thành một chuỗi. Loại bỏ các hàm của ký tự đặc biệt được dùng trong chuỗi.
- ‘ ’ coi toàn bộ ký tự đặc biệt là ký tự thường và “ ” coi toàn bộ ký tự đặc biệt trừ \$ và \ là ký tự thường.

Ví dụ	Giải thích
echo '\$SHELL'	Prints \$SHELL string
echo "\$SHELL"	Kiểu Shell hiện tại hay chính là giá trị được lưu trữ trong SHELL, biến môi trường của Shell, sẽ được in ra. Ví dụ, /bin/sh.

## Ký tự đặc biệt ''

- Shell thông dịch một chuỗi với hai đầu là '' tương đương một lệnh và biến thành kết quả thực thi lệnh.

Ví dụ	Giải thích
echo "Today is 'date'"	'date' được thông dịch là lệnh và trả thành kết quả của lệnh date. In ra kết quả như sau: Today is 2019.12.25(Wed) 19:19:31 UTC
ls /platform/`uname -m`	Biến kết quả của lệnh uname –m thành một chuỗi và liệt kê dưới tên của một thư mục.

## Ký tự đặc biệt \, >, <, >>

- █ \ được dùng ngay đằng trước một ký tự đặc biệt. Loại bỏ chức năng của ký tự đặc biệt và xử lý như ký tự thường.
- █ Ký tự đặc biệt \, >, <, >> chuyển hướng in ra của lệnh

Ví dụ	Giải thích
ls-lt\*	In ra thông tin chi tiết về các tệp tin có tên t*. Khi dùng t* mà không có \, in ra thông tin chi tiết về mọi tệp tin bắt đầu bằng 't'
echo \\$SHELL	Prints \$SHELL. Cho ra kết quả tương tự echo '\$SHELL'.

Ví dụ	Giải thích
ls-l > res	Liệt kê kết quả của lệnh ls -l trong tệp res (Không in ra màn hình)

## Thiết bị nhập/xuất tiêu chuẩn

- | Thiết bị nhập tiêu chuẩn: Một thiết bị nhận thông tin cần thiết cho Shell để thực hiện tác vụ → Bàn phím
- | Thiết bị xuất tiêu chuẩn: Một thiết bị in ra một kết quả → Màn hình
- | Thiết bị báo lỗi tiêu chuẩn: Một thiết bị in ra thông báo lỗi → Màn hình

## Đặc tả tệp tin

- | Một dãy số quản lý tệp tin.
- | Đặc tả tệp tin được dùng khi thay đổi thiết bị nhập/xuất.
- | ‘Chuyển hướng’ có nghĩa là đổi thiết bị nhập/xuất tiêu chuẩn như tệp tin.

File Descriptor	Tên khác của file descriptor (đặc tả tệp tin).	Định nghĩa
0	stdin	Nhập tiêu chuẩn một lệnh
1	stdout	Xuất theo tiêu chuẩn một lệnh
2	stderr	Báo lỗi tiêu chuẩn của một lệnh

## chuyển hướng xuất

| > : Xóa nội dung của tệp tin hiện có và lưu kết quả mới

| >> : Bổ sung kết quả sau nội dung của tệp tin hiện có

### > : Ghi đè tệp tin

| 1: Đặc tả tệp tin số 1 (thiết bị xuất tiêu chuẩn, màn hình)

| Shell tạo một tệp tin để lưu kết quả của lệnh nếu không xác định tên tệp tin theo chuyển hướng bằng ">"

| Nếu tệp tồn tại, nội dung trước sẽ bị mất và thay thế bằng lệnh



Chức năng Chuyển hướng tệp tin (ghi đè)

Định dạng Lệnh 1> tên tệp tin  
Lệnh 1>> tên tệp tin

## Ví dụ ghi đè một tệp tin bằng >

```
pi@raspberrypi:~ $ ls -i
5295 access.txt      5289 data          259098 Pictures
5114 a.txt.cp       259086 Desktop      259095 Public
3320 a.txt.ln       259096 Documents    388997 temp
5125 a.txt.sl       259093 Downloads    259094 Templates
1340 Bookshelf      259097 Music        259099 Videos
3720 b.txt         5297 new access.txt

pi@raspberrypi:~ $ ls -i > output_tmp
pi@raspberrypi:~ $ ls
access.txt  Bookshelf  Documents  output_tmp  Templates
a.txt.cp    b.txt     Downloads  Pictures   Public
a.txt.ln    data      Music     Public
a.txt.sl    Desktop   new access.txt temp

pi@raspberrypi:~ $ cat output_tmp
5295 access.txt
5114 a.txt.cp
3320 a.txt.ln
5125 a.txt.sl
1340 Bookshelf
3720 b.txt
5289 data
259086 Desktop
259096 Documents
259093 Downloads
259097 Music
5297 new access.txt
5110 output_tmp
259098 Pictures
259095 Public
388997 temp
259094 Templates
259099 Videos
pi@raspberrypi:~ $
```

## >> : Thêm nội dung vào tệp tin

- > Chức năng    Thêm nội dung vào tệp tin.
- Định dạng    Lệnh >> tên tệp tin

| Nếu không có tệp tin xác định, một tệp tin mới sẽ được tạo. Nếu tồn tại một tệp tin xác định, nó sẽ thực thi lệnh ở cuối tệp tin hiện có.

The screenshot shows a terminal window titled "pi@raspberrypi: ~". The window has a menu bar with "File", "Edit", "Tabs", and "Help". The terminal content is as follows:

```
pi@raspberrypi:~ $ cat a.txt.ln
a.txt.ln is edited!
pi@raspberrypi:~ $ date >> a.txt.ln
pi@raspberrypi:~ $ cat a.txt.ln
a.txt.ln is edited!
Sun 22 Aug 2021 03:13:20 AM KST
pi@raspberrypi:~ $
```

The command `cat a.txt.ln` was run first, followed by `date >> a.txt.ln`, and then another `cat a.txt.ln`. The output shows that the date was appended to the end of the file, indicating successful use of the redirection operator.

## Chuyển hướng báo lỗi

- █ Báo lỗi tiêu chuẩn cũng được in ra trên màn hình theo mặc định và có thể được chuyển hướng như xuất tiêu chuẩn.
- █ Ví dụ đầu ra tiêu chuẩn và báo lỗi tiêu chuẩn
- █ Chuyển hướng xuất tiêu chuẩn: thông báo lỗi không được chuyển hướng.

The screenshot shows a terminal window titled "pi@raspberrypi: ~". The terminal displays the following sequence of commands and their outputs:

```
pi@raspberrypi:~ $ ls -i
5295 access.txt      3720 b.txt      259097 Music      388997 temp
5114 a.txt.cp       5289 data       5297 new_access.txt 259094 Templates
3320 a.txt.ln       259086 Desktop   5110 output_tmp   259099 Videos
5125 a.txt.sl       259096 Documents 259098 Pictures
1340 Bookshelf      259093 Downloads 259095 Public

pi@raspberrypi:~ $ ls -i xxxxx
ls: cannot access 'xxxxx': No such file or directory
pi@raspberrypi:~ $ ls -i > ls.out
pi@raspberrypi:~ $ ls -i xxxxx > ls.err
ls: cannot access 'xxxxx': No such file or directory
pi@raspberrypi:~ $ cat ls.out
5295 access.txt
5114 a.txt.cp
3320 a.txt.ln
5125 a.txt.sl
1340 Bookshelf
3720 b.txt
5289 data
259086 Desktop
259096 Documents
259093 Downloads
5335 ls.out
259097 Music
5297 new_access.txt
5110 output_tmp
259098 Pictures
259095 Public
388997 temp
259094 Templates
259099 Videos

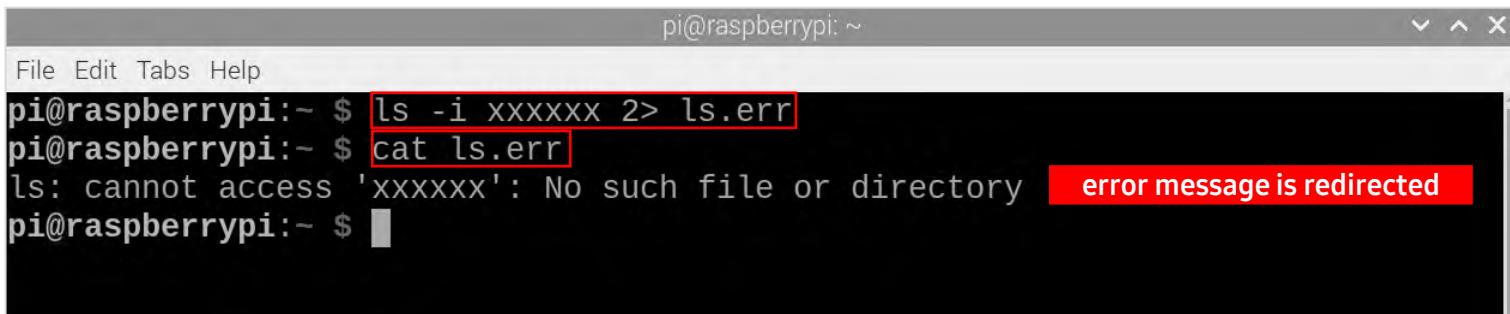
pi@raspberrypi:~ $ cat ls.err
error message is not redirected
```

## Chuyển hướng báo lỗi

Khi chuyển hướng báo lỗi, không thể bỏ qua số đặc tả tệp tin.

2>

Chức năng    Lưu thông báo lỗi tiêu chuẩn vào một tệp tin  
Định dạng    Lệnh 2> tên tệp tin



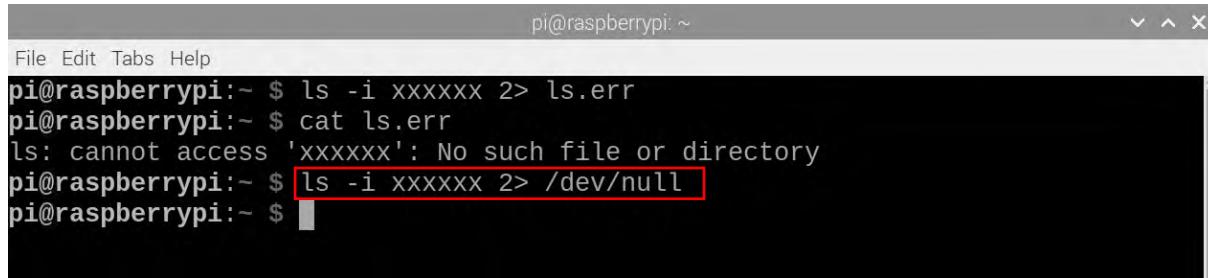
A screenshot of a terminal window titled "pi@raspberrypi: ~". The window has a menu bar with "File", "Edit", "Tabs", "Help", and icons for "v", "^", and "x". The terminal content shows the following command sequence:

```
pi@raspberrypi:~ $ ls -i xxxxxx 2> ls.err
pi@raspberrypi:~ $ cat ls.err
ls: cannot access 'xxxxxx': No such file or directory
error message is redirected
pi@raspberrypi:~ $
```

The command `ls -i xxxxxx 2> ls.err` is highlighted with a red box. The output line `ls: cannot access 'xxxxxx': No such file or directory` is also highlighted with a red box, and the text "error message is redirected" is overlaid in red to its right.

## Chuyển hướng báo lỗi

| Hủy thông báo lỗi.



A screenshot of a terminal window titled "pi@raspberrypi: ~". The window has a menu bar with "File", "Edit", "Tabs", and "Help". The terminal content shows the following commands and their output:

```
pi@raspberrypi:~ $ ls -i xxxxxx 2> ls.err
pi@raspberrypi:~ $ cat ls.err
ls: cannot access 'xxxxxx': No such file or directory
pi@raspberrypi:~ $ ls -i xxxxxx 2> /dev/null
pi@raspberrypi:~ $
```

The command `ls -i xxxxxx 2> /dev/null` is highlighted with a red rectangle.

- ▶ Nếu truy cập vào /dev/null, chúng hoàn toàn không tồn tại.

# Chuyển hướng báo lỗi



Chức năng

Thay đổi đầu vào tiêu chuẩn

Định dạng

Lệnh  $\odot <$ tên tệpLệnh  $\odot <$ tên tệp

## Ví dụ về câu lệnh chuyển hướng bằng lệnh cat

```
pi@raspberrypi:~ $ cat Desktop/test.txt
This text file will be copied to Raspberry Pi

It include some empty rows
Let's check this out
pi@raspberrypi:~ $ cat < Desktop/test.txt
This text file will be copied to Raspberry Pi

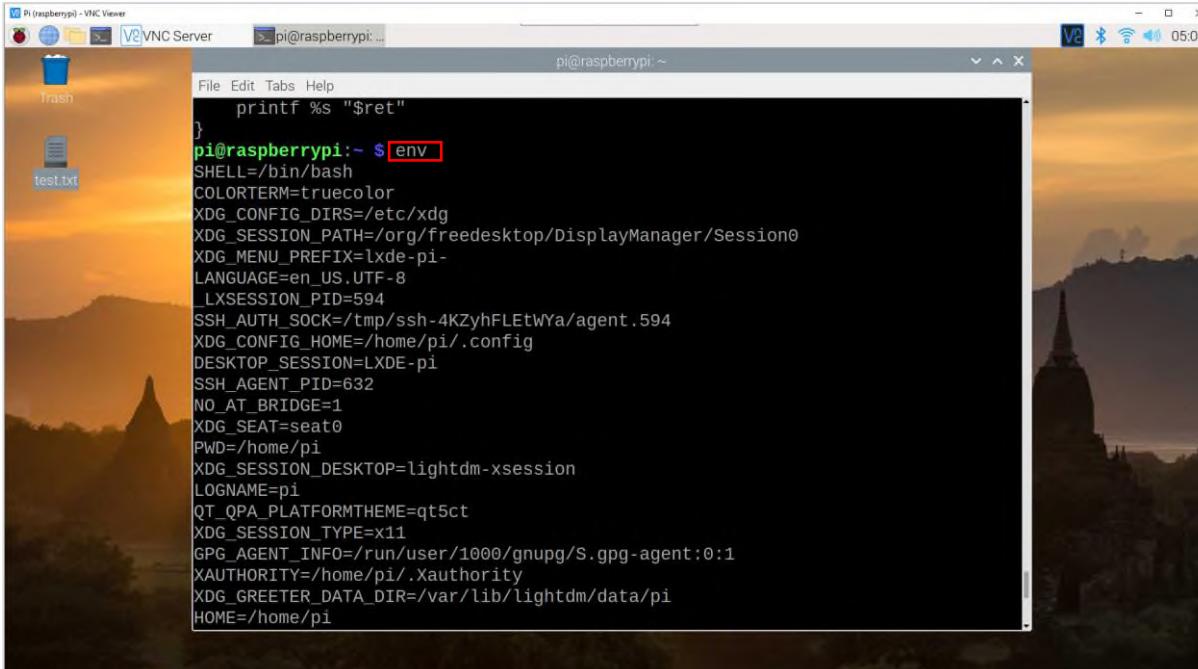
It include some empty rows
Let's check this out
pi@raspberrypi:~ $ cat @< Desktop/test.txt
This text file will be copied to Raspberry Pi

It include some empty rows
Let's check this out
pi@raspberrypi:~ $
```

## In toàn bộ biến bằng lệnh set

| set: in cả biến shell và biến môi trường.

| env: in các biến môi trường



```
pi@raspberrypi:~ $ env
SHELL=/bin/bash
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_MENU_PREFIX=lxde-pi-
LANGUAGE=en_US.UTF-8
_LXSESSION_PID=594
SSH_AUTH_SOCK=/tmp/ssh-4KZyhFLEtWYa/agent.594
XDG_CONFIG_HOME=/home/pi/.config
DESKTOP_SESSION=LXDE-pi
SSH_AGENT_PID=632
NO_AT_BRIDGE=1
XDG_SEAT=seat0
PWD=/home/pi
XDG_SESSION_DESKTOP=lightdm-xsession
LOGNAME=pi
QT_QPA_PLATFORMTHEME=qt5ct
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/home/pi/.Xauthority
XDG_GREETER_DATA_DIR=/var/lib/lightdm/data/pi
HOME=/home/pi
```

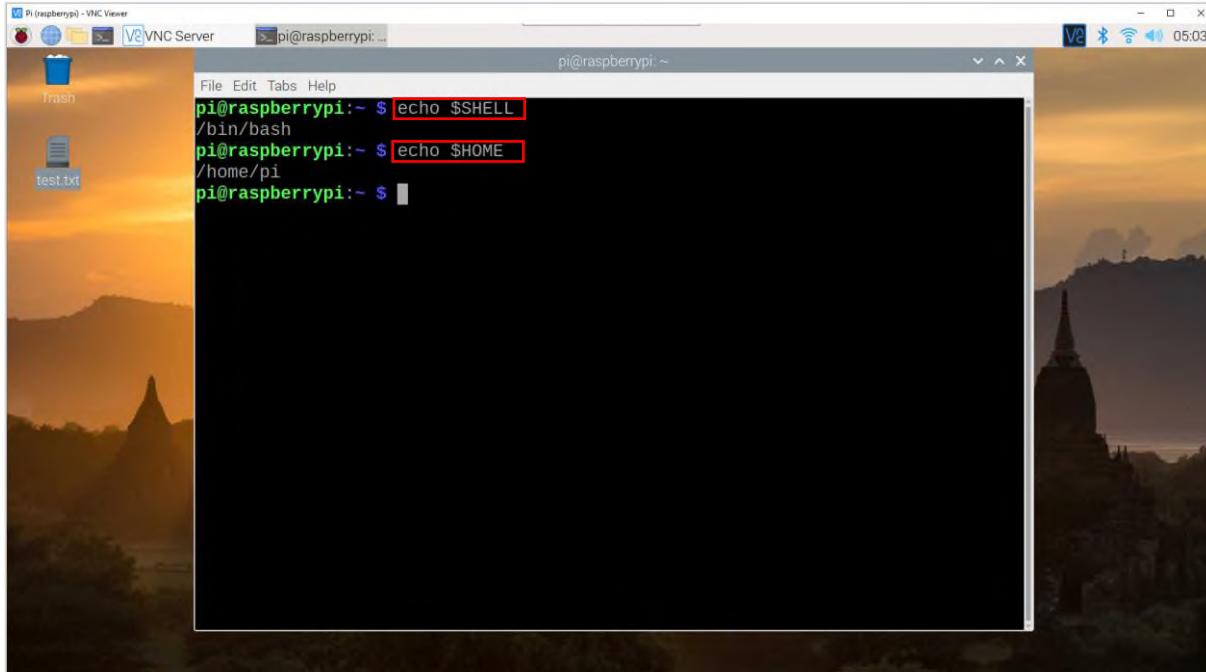
## Danh sách các biến môi trường chính

| set: in cả biến Shell và biến môi trường

biến môi trường	Giải thích	biến môi trường	Giải thích
HISTSIZE	Kích thước kho lưu trữ lịch sử	PATH	Đường dẫn để tìm kiếm lệnh
HOME	Đường dẫn tuyệt đối tới thư mục chính người dùng	PWD	đường dẫn tuyệt đối tới thư mục đang làm việc
LANG	Ngôn ngữ được sử dụng	SHELL	Shell đăng nhập
LOGNAME	Tên tài khoản người dùng		

## In biến cụ thể bằng lệnh echo

Khi in giá trị của một biến, thêm ký tự đặc biệt \$ vào đầu trước tên biến.



## Thiết lập biến Shell

### Thiết lập biến Shell

Định dạng	Tên biến = Chuỗi
Ví dụ	SOME = test

```
pi@raspberrypi:~ $ SOME=test
pi@raspberrypi:~ $ echo $SOME
test
pi@raspberrypi:~ $
```

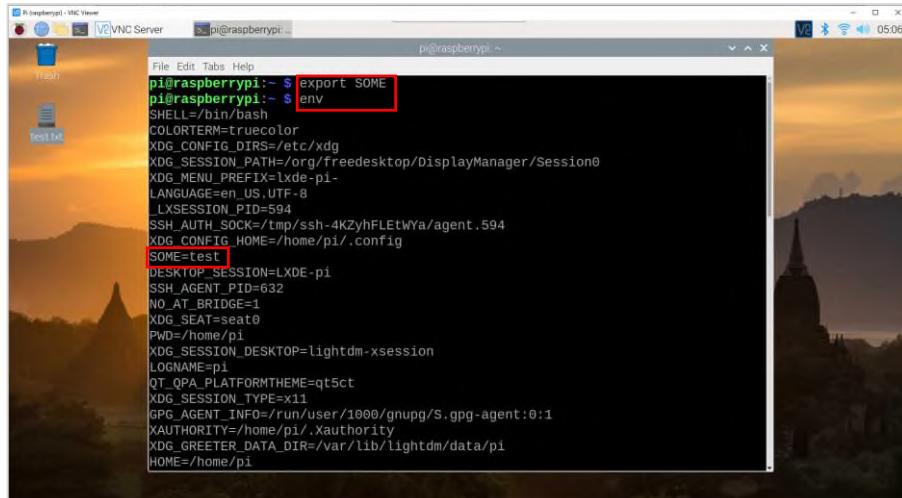
There should be no space between variable name and string.

## Thiết lập biến Shell

### export

Chức năng	Thay thế biến cell cụ thể bằng một biến môi trường.
Định dạng	Export [-n] [Biến cell]
Option	-n: Chuyển đổi biến môi trường thành biến cell.
Ví dụ	export export SOME export SOME=test

Đầu tiên, xác định biến Shell và dùng lệnh export để chuyển thành biến môi trường

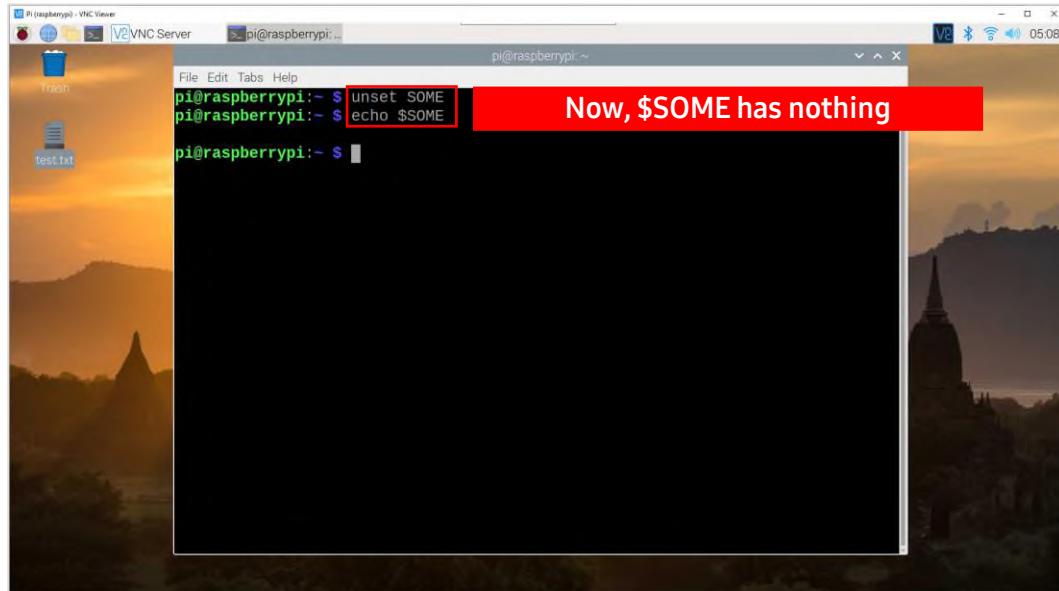


```
pi@raspberrypi:~ $ export SOME
pi@raspberrypi:~ $ env
SHELL=/bin/bash
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_MENU_PREFIX=lxde-pi-
LANGUAGE=en_US.UTF-8
_LXSESSION_PID=594
SSH_AUTH_SOCK=/tmp/ssh-4KZhFLEtWYa/agent.594
XDG_CONFIG_HOME=/home/pi/.config
SOME=test
DESKTOP_SESSION=lxde-pi
SSH_AGENT_PID=632
NO_AT_BRIDGE=1
XDG_SEAT=seat0
PWD=/home/pi
XDG_SESSION_DESKTOP=lightdm-xsession
LOGNAME=pi
QT_QPA_PLATFORMTHEME=qt5ct
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/home/pi/.Xauthority
XDG_GREETER_DATA_DIR=/var/lib/lightdm/lightdm-data/pi
HOME=/home/pi
```

## Hủy đặt biến bằng lệnh unset

### unset

Chức năng	Hoàn tác biến cụ thể.
Định dạng	unset biến
Ví dụ	unset SOME



## alias

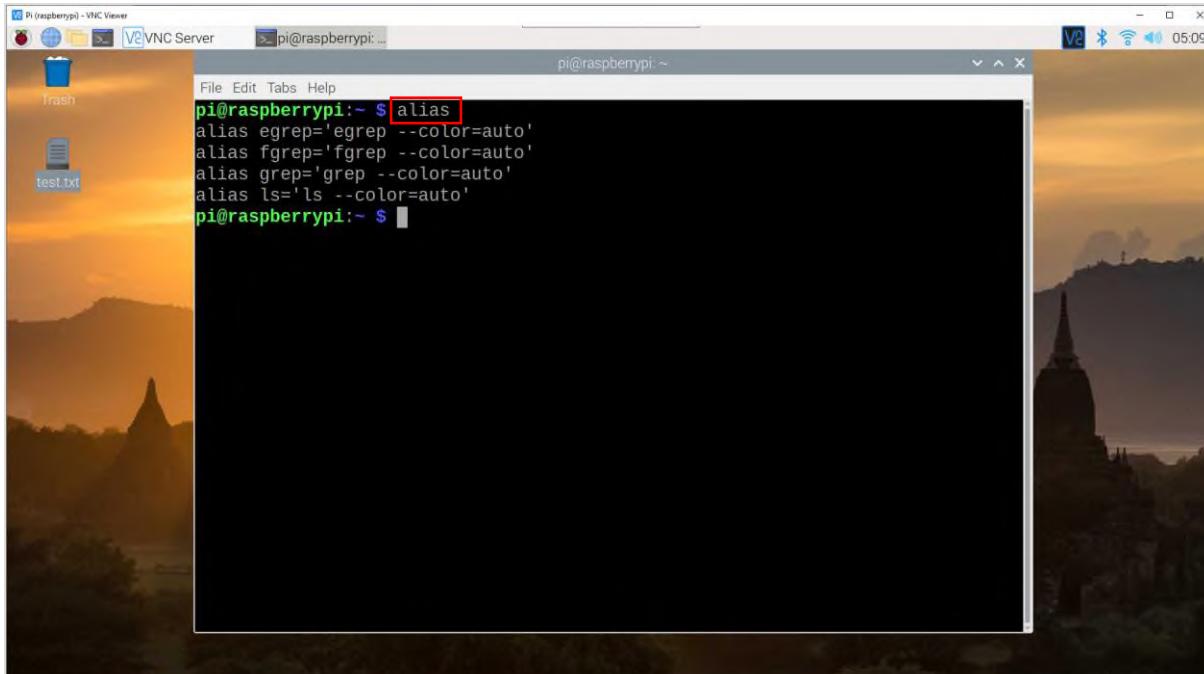
- | Cho phép tạo một tên shortcut (lối tắt) cho một lệnh, tên tệp tin hoặc bất kỳ văn bản Shell nào
- | Giúp tiết kiệm thời gian bằng lệnh ngắn hơn thay vì lệnh dài
- | Có thể liên kết nhiều lệnh thành một
- | Có thể tạo nhiều bí danh cho lệnh bằng tên lối tắt bao gồm những option thường dùng

### alias

Chức năng	Tạo bí danh.
Định dạng	alias name = 'command'
Ví dụ	alias: In một danh sách các bí danh hiện được đặt alias name = 'command': Khi dùng một lệnh được chỉnh sửa alias name = 'command;command2;...': Khi dùng nhiều lệnh với một tên

## alias

| Nếu bạn chạy lệnh alias mà không xác định, bí danh hiện được đặt sẽ được in.



## alias

| Dưới đây là ví dụ đặt bí danh cho lệnh ls

```
pi@raspberrypi:~ $ ls
access.txt  Bookshelf  Documents  Music      Public
a.txt.cp    b.txt      Downloads   new_access.txt temp
a.txt.ln    data       ls.err     output_tmp   Templates
a.txt.sl    Desktop   ls.out     Pictures    Videos
pi@raspberrypi:~ $ ls -F
access.txt  Bookshelf/  Documents/  Music/      Public/
a.txt.cp    b.txt*     Downloads/  new_access.txt temp/
a.txt.ln    data       ls.err     output_tmp   Templates/
a.txt.sl@   Desktop/   ls.out     Pictures/   Videos/
pi@raspberrypi:~ $ alias ls='ls -F'
pi@raspberrypi:~ $ ls
access.txt  Bookshelf/  Documents/  Music/      Public/
a.txt.cp    b.txt*     Downloads/  new_access.txt temp/
a.txt.ln    data       ls.err     output_tmp   Templates/
a.txt.sl@   Desktop/   ls.out     Pictures/   Videos/
pi@raspberrypi:~ $
```

Now, ls command is changed like ls -F

## unalias: Hoàn tác đặt bí danh

### unalias

Chức năng	Xóa bí danh.
Định dạng	unalias bí danh

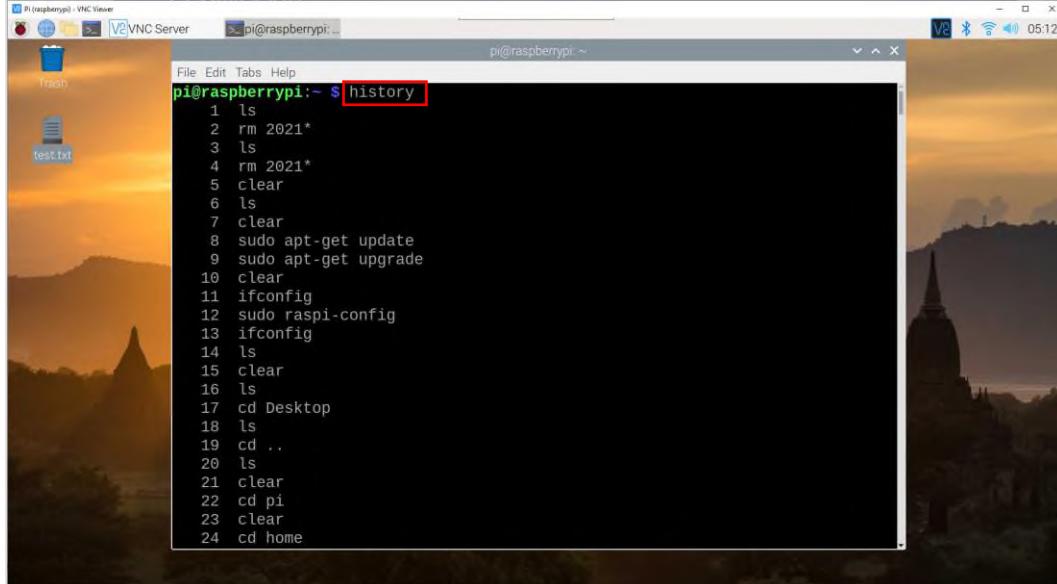
```
pi@raspberrypi:~ $ unalias ls
pi@raspberrypi:~ $ ls
access.txt Bookshelf Documents Music Public
a.txt.cp b.txt Downloads new_access.txt temp
a.txt.ln data ls.err output_tmp Templates
a.txt.sl Desktop ls.out Pictures Videos
pi@raspberrypi:~ $
```

Now, ls command is not like a ls -F

## history: Nhắc lại lệnh do người dùng nhập trước đó

### history

Chức năng	In lịch sử nhập lệnh.
Định dạng	Lịch sử



A screenshot of a VNC session on a Raspberry Pi. The desktop background shows a sunset over mountains. A terminal window is open, displaying the output of the 'history' command. The command was run at the prompt 'pi@raspberrypi:~ \$'. The output lists 24 previous commands, including file operations like 'ls' and 'rm', system updates with 'sudo apt-get update' and 'upgrade', network configuration with 'ifconfig', and directory navigation with 'cd'. The terminal window has a dark theme.

```
pi@raspberrypi:~ $ history
1 ls
2 rm 2021*
3 ls
4 rm 2021*
5 clear
6 ls
7 clear
8 sudo apt-get update
9 sudo apt-get upgrade
10 clear
11 ifconfig
12 sudo raspi-config
13 ifconfig
14 ls
15 clear
16 ls
17 cd Desktop
18 ls
19 cd ..
20 ls
21 clear
22 cd pi
23 clear
24 cd home
```

Bài 4

# Trình chỉnh sửa và IDE

| 4.1. Trình soạn thảo văn bản

| 4.2. Nano

| 4.3. IDE

| 4.4. VSC

## Trình soạn thảo văn bản

### Phần mềm dùng để soạn thảo file tài liệu đơn giản

- Đây là một trong những phần mềm cơ bản nhất trên máy tính có thể đọc các tệp ở định dạng văn bản mà con người có thể đọc được thay vì định dạng nhị phân, đồng thời chỉnh sửa và lưu chúng trong các bước đơn giản.

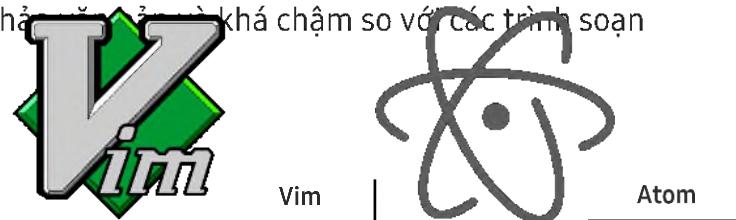
### Tệp văn bản chung so với tệp trình xử lý Word

- Về cơ bản, có một sự khác biệt quan trọng giữa tệp văn bản thuần túy được tạo bằng trình soạn thảo văn bản và tệp tài liệu được tạo bằng trình xử lý văn bản như Microsoft Word.
- Bạn có thể chỉnh sửa một tệp văn bản thuần túy để hiển thị nó chính xác như được thể hiện trong tệp. Văn bản duy nhất không thể hiển thị là ký tự điều khiển của bộ ký tự được sử dụng. Ví dụ thực tế bao gồm thay đổi dòng và tab.
- Các tệp văn bản nói chung được sử dụng nhiều hơn để lập trình và viết hơn là để viết tài liệu trong quá khứ.
- Ngoài việc xác định các bộ ký tự, tài liệu thường chứa các loại tệp chi tiết "ký tự điều khiển", có thể kích hoạt các tính năng như in đậm, in nghiêng, phông chữ, dòng và bảng.

## Trình soạn thảo văn bản

### Các loại soạn thảo văn bản

- ▶ Mục đích của Text Editor là sử dụng nhẹ khi chỉ chỉnh sửa văn bản thuận túy mà không có các biểu mẫu riêng biệt.
- ▶ Trình chỉnh sửa văn bản đại diện bao gồm vim, nano và nguyên tử. Chúng ta hãy xem nhanh các tính năng của các trình soạn thảo này.
- ▶ Vim là trình soạn thảo văn bản đa hệ điều hành so với vi.
  - Vim tương thích với vi đồng thời hỗ trợ sự tiện lợi của người dùng bằng cách bổ sung riêng các chức năng khác nhau.
  - Ưu điểm là bạn có thể tự do thay đổi môi trường chỉnh sửa của mình bằng cách sử dụng các tập lệnh Vim và hơn thế nữa hoặc sử dụng ngữ pháp biểu thức chính quy mở rộng, đánh dấu ngữ pháp mạnh mẽ, hoàn tác nhiều lần và hỗ trợ đa ngôn ngữ với UNICODE và kiểm tra ngữ pháp.
  - Trình soạn thảo văn bản cơ bản của bản phân phối Linux.
  - Mặc dù sự tiện lợi ngày càng tăng, cấu trúc này vẫn khó sử dụng đối với những người lần đầu sử dụng Vim.
- ▶ Atom là một loại mã nguồn mở miễn phí của trình soạn thảo mã nguồn và tài liệu đa hệ điều hành.
  - Atom dựa trên Chromium và được chuẩn bị bằng CoffeeScript. Nó có thể được sử dụng với bất kỳ IDE nào.
  - Nó có rất nhiều tính năng hay nhưng nó hơi nặng khi là một trình soạn thảo văn bản so với các trình soạn thảo khác.



## Trình soạn thảo văn bản

### Các loại Trình soạn thảo Văn bản

- ▶ Nano là trình soạn thảo văn bản cho hệ điều hành hoặc hệ thống máy tính dựa trên UNIX sử dụng giao diện dòng lệnh.
  - Cũng giống như vim, nó là trình soạn thảo văn bản cơ bản của bản phân phối Linux.
  - Tháng 2 năm 2001 nano trở thành một phần của dự án GNU. Nó có một số tính năng không có trong Pico, chẳng hạn như màu văn bản, tìm kiếm và thay thế cụm từ thông dụng, cuộn trớn tru, nhiều bộ đệm, hỗ trợ phím có thể kết hợp lại, hoàn tác và thử lại các thay đổi chỉnh sửa.
  - Đối với những người lần đầu sử dụng, nó dễ sử dụng hơn một chút so với vim.
  - Nó có thể được sử dụng nhẹ hơn một chút so với nguyên tử.
- ▶ Vì lý do này, hãy xem cách sử dụng Nano làm Trình chỉnh sửa.

```
!!!  
iLE88Dj. :jD88888Dj:  
.LGitE888D.f8GjjjL8888E;  
iE :8888Et. .G8888.  
;i E888, ,8888,  
D888, :8888:  
D888, :8888:  
D888, :8888:  
888W, :8888:  
W88W, :8888:  
W88W: :8888:  
DGGD: :8888:  
:8888:  
:W888:  
:8888:  
E888i  
tW88D
```

Nano

Bài 4

## Trình chỉnh sửa và IDE

| 4.1. Trình soạn thảo văn bản

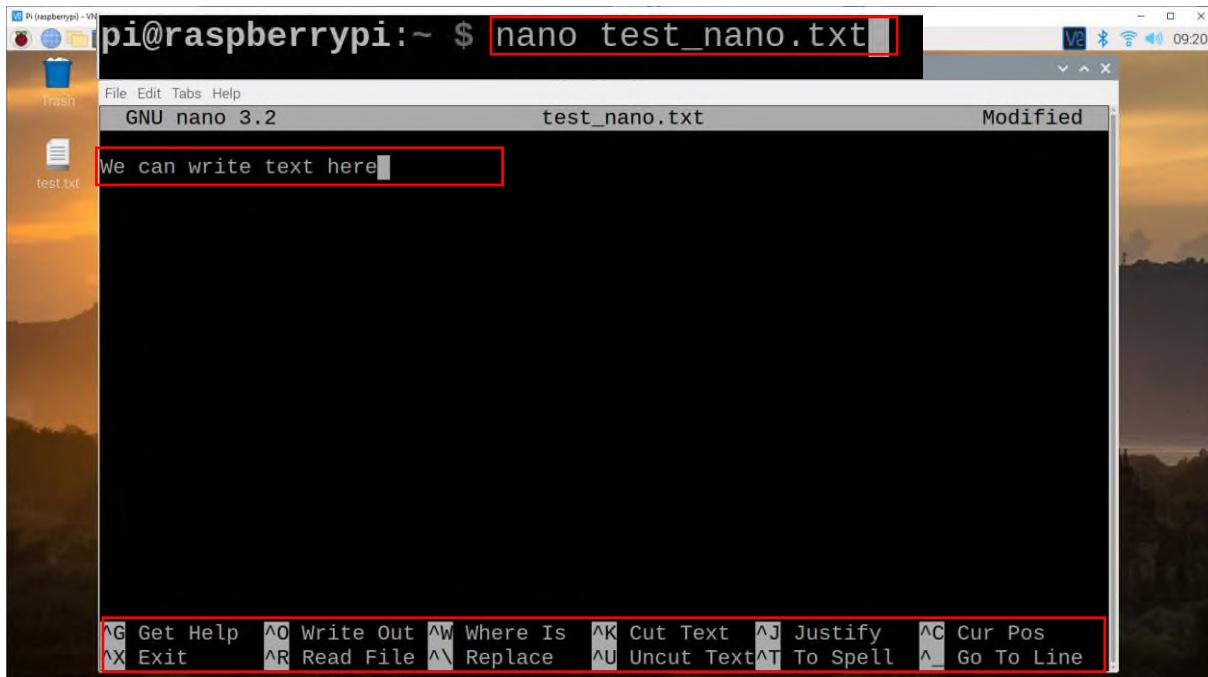
| **4.2. Nano**

| 4.3. IDE

| 4.4. VSC

# Nano

## Cách sử dụng nano



- ▶ Nếu lệnh được thực thi ở định dạng như `nano <tên tệp>`, thì tệp được tạo với tên tệp liên quan và nano được thực thi.
- ▶ Có thể nhập văn bản tự do vào con trỏ và có thể tìm thấy các phím tắt ở phía dưới.
- ▶ ^ có nghĩa là Ctrl +, vì vậy, ví dụ, để Nhận trợ giúp, bạn có thể nhập Ctrl + g trên bàn phím.

# Nano

## | Cách di chuyển con trỏ nano

- ▶ Di chuyển con trỏ trong nano cực kỳ đơn giản.
- ▶ Phím mũi tên của bàn phím có thể được sử dụng để di chuyển con trỏ đến hướng mong muốn từng khoảng một.
- ▶ Có thể dùng phím Home, End để di chuyển về đầu và cuối dòng.
- ▶ Bạn có thể sử dụng Ctrl + phím mũi tên trái, phải để di chuyển một từ về phía trước hoặc một từ về sau.
- ▶ Bạn có thể sử dụng Trang lên, Trang xuống để chuyển đến trang tiếp theo hoặc trang trước đó.

# Nano

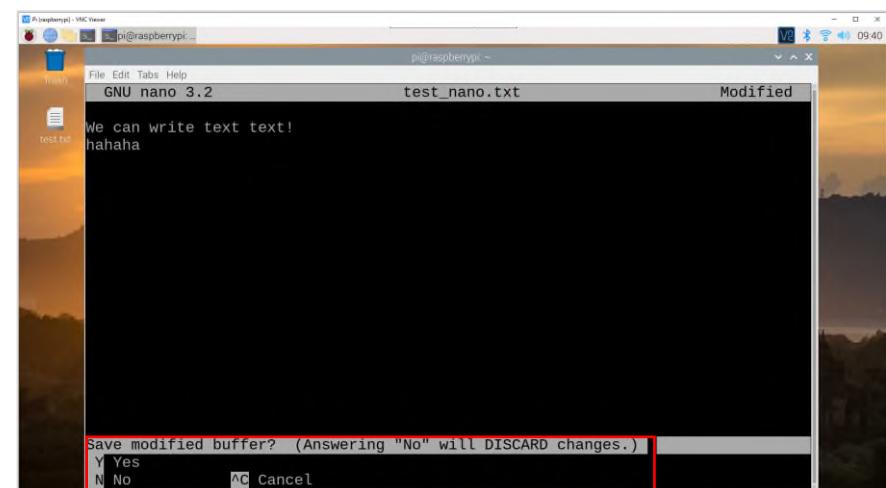
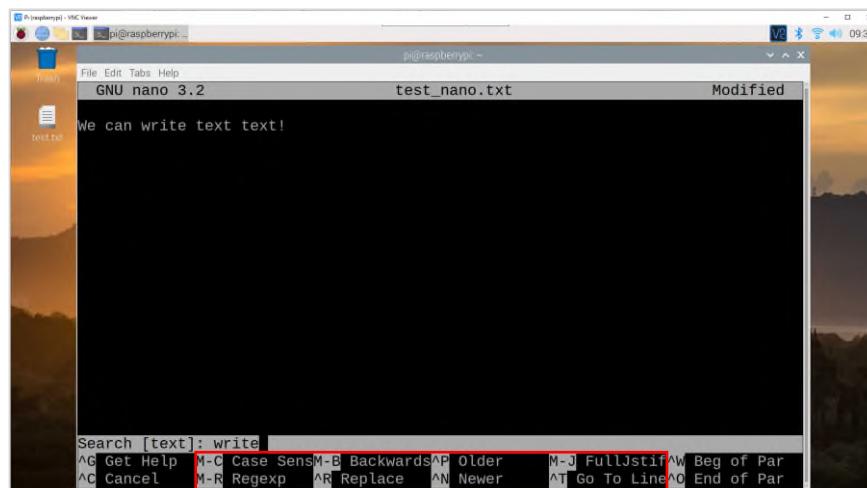
## I Các phím tắt thường dùng

- ▶ Alt + u : Hoàn tác
- ▶ Alt + e : Làm lại
- ▶ Alt + n : In số dòng
- ▶ Alt + a : Gán/tắt điểm bắt đầu của khối đã chọn
- ▶ Alt + 6 : Sao chép khối đã chọn vào Clipboard
- ▶ Ctrl + k : Cắt khối được chọn / Cắt 1 dòng nếu không có khối được chọn
- ▶ Ctrl + u : Dán
- ▶ Ctrl + s : Lưu
- ▶ Ctrl + o : Lưu dưới dạng
- ▶ Ctrl + x : Kết thúc

# Nano

## Cách sử dụng nano

- ▶ Nếu nhập Ctrl + w, là lệnh Where được viết ở dưới cùng, bạn có thể tìm kiếm văn bản với cụm từ Search:
- ▶ Nếu bạn nhìn vào phần dưới cùng ở đây, bạn có thể thấy rằng cấu hình của các phím tắt đã thay đổi. Trong đó M là viết tắt của Meta Key, Meta Key cho Windows là Alt và ESC cho Mac. Luôn có hướng dẫn ở phía dưới, giúp bạn dễ dàng thực hành cách sử dụng nano.
- ▶ Để lưu và thoát, bạn chỉ cần gõ lệnh thoát Ctrl + x. Bạn sẽ được hỏi có muốn lưu các sửa đổi của mình ở đây không. Bạn có thể tiếp tục bằng cách gõ y (có) hoặc n (không).



Bài 4

## Trình chỉnh sửa và IDE

- | 4.1. Trình soạn thảo văn bản
- | 4.2. Nano
- | **4.3. IDE**
- | 4.4. VSC

## IDE

### | IDE (Integrated Development Environment) là gì?

- ▶ IDE là phần mềm để tạo các ứng dụng kết hợp các công cụ phổ biến của nhà phát triển thành một Giao diện người dùng đồ họa (GUI) duy nhất.
- ▶ Nói chung, IDE bao gồm 3 yếu tố sau:
  - Trình chỉnh sửa mã nguồn: Trình soạn thảo văn bản giúp bạn viết mã phần mềm, bao gồm tô sáng cú pháp bằng các ký hiệu trực quan. Nó cung cấp tính năng tự động hoàn thành ngôn ngữ cụ thể và kiểm tra lỗi khi viết mã.
  - Tự động hóa bản dựng cục bộ: Một tiện ích tự động hóa các tác vụ đơn giản và lặp đi lặp lại như biên dịch mã nguồn máy tính thành mã nhị phân, đóng gói mã nhị phân và chạy thử nghiệm tự động khi tạo các bản dựng cục bộ của phần mềm được người dùng sử dụng.
  - Trình gỡ lỗi: Một chương trình cho phép bạn kiểm tra các chương trình khác và hiển thị bằng đồ họa vị trí gỡ lỗi của mã nguồn của bạn.

## IDE

### | Tại sao các lập trình viên sử dụng IDE?

- ▶ Nó cho phép các nhà phát triển nhanh chóng lập trình các ứng dụng mới bằng cách loại bỏ nhu cầu cấu hình thủ công và tích hợp nhiều tiện ích như một phần của quy trình thiết lập.
- ▶ Sẽ rất hữu ích cho các nhà phát triển mới khi sử dụng IDE để làm quen với các công cụ và quy trình làm việc tiêu chuẩn của nhóm khi giới thiệu.
- ▶ Nó tiết kiệm thời gian với các tính năng, chẳng hạn như hoàn thành mã thông minh và tạo mã tự động giúp loại bỏ nhu cầu nhập toàn bộ chuỗi.
- ▶ Nó giúp các nhà phát triển tổ chức quy trình công việc và giải quyết vấn đề.
- ▶ Nó xác định các lỗi do lỗi của con người gây ra trong thời gian thực bằng cách phân tích mã khi nó được viết.
- ▶ Hầu hết các nhóm phát triển doanh nghiệp ngày nay chọn một IDE được cấu hình sẵn được tối ưu hóa cho một trường hợp sử dụng cụ thể.

# IDE

## Các loại IDE

- ▶ Có nhiều loại IDEs, nhưng các IDEs tiêu biểu bao gồm PyCharm, Eclipse và VSC (Visual Studio Code).
- ▶ Nói một cách chính xác, VSC có thể được mở rộng thành trình soạn thảo văn bản hoặc IDE.
- ▶ Chúng ta hãy xem qua các tính năng của từng IDE:
- ▶ Eclipse
  - Nó là một môi trường phát triển tích hợp dựa trên Java. Nó được tạo ra bởi Quỹ Eclipse. Vì Java là ngôn ngữ lập trình được phát triển bởi Sun Microsystems, Eclipse, có nghĩa là một biểu thức liên quan đến mặt trời, được lấy làm tên.
  - Nó có thể được sử dụng, sửa đổi và phân phối lại miễn phí ở mọi nơi trong Windows, macOS hoặc Linux. Đối với macOS, có thể cài đặt Java mà không cần đặt biến môi trường, đây là đặc điểm của các HĐH giống Unix không phụ thuộc vào sổ đăng ký. Ngay cả khi bạn chỉ cài đặt thời gian chạy Java (JRE), trình biên dịch được bao gồm ngay lập tức, vì vậy bạn có thể cài đặt Java và gõ javac trong thiết bị đầu cuối để hoạt động ngay lập tức.
  - Vì nó có thể được sử dụng ở mọi nơi mà không bị giới hạn bởi hệ điều hành, nên nó có thể được sử dụng để phát triển phần mềm trên nhiều nền tảng khác nhau.



# IDE

## Các loại IDE

### ▶ Pycharm

- PyCharm là một môi trường phát triển tích hợp (IDE) được sử dụng trong lập trình máy tính, dành riêng cho ngôn ngữ Python.
- Nó được phát triển bởi công ty Séc JetBrains (trước đây gọi là IntelliJ).
- Nó cung cấp phân tích mã, trình gỡ lỗi đồ họa, trình kiểm tra Bài tích hợp, tích hợp với các hệ thống kiểm soát phiên bản (VCSes) và hỗ trợ phát triển web với Django cũng như khoa học dữ liệu với Anaconda.
- Điều hướng dự án và mã: chế độ xem dự án chuyên biệt, chế độ xem cấu trúc tệp và chuyển nhanh giữa các tệp, lớp, phương thức và tập quán.
- Tái cấu trúc Python: bao gồm đổi tên, trích xuất phương thức, giới thiệu biến, giới thiệu hằng số, kéo lên, đẩy xuống và những thứ khác.
- Hỗ trợ cho các khung web: Django, web2py và Flask [chỉ phiên bản chuyên nghiệp] [8].
- Trình gỡ lỗi Python tích hợp.
- Thủ nghiệm Bài tích hợp, với phạm vi mã từng dòng.
- Phát triển Python của Google App Engine [chỉ dành cho phiên bản chuyên nghiệp].
- Tích hợp kiểm soát phiên bản: giao diện người dùng hợp nhất cho Mercurial, Git, Subversion, Perforce và CVS với danh sách thay đổi và hợp nhất.

# IDE

## Các loại IDE

### ► VSC(Visual Studio Code)

- Nó là một trình soạn thảo văn bản được giới thiệu vào ngày 29 tháng 4 năm 2015 và phiên bản đầy đủ 1.0.0 được phát hành vào ngày 15 tháng 4 năm 2016. Nó được phát triển bởi Microsoft.
- Nó được xây dựng trên khung Electron. Đây là trình chỉnh sửa đầu tiên hỗ trợ đa nền tảng trong số các công cụ phát triển của Microsoft và hỗ trợ Windows, macOS và Linux.
- Ưu điểm lớn nhất là các chức năng mở rộng cho phép nó được mở rộng lên IDE, ngoài hoạt động như một trình soạn thảo đơn giản và các tài liệu chính thức của Microsoft cung cấp các ví dụ về việc sử dụng các chức năng mở rộng đó.
- Vì Visual Studio Code không phải là IDE nên nó không có trình xây dựng tích hợp, vì vậy bạn phải xây dựng một môi trường biên dịch riêng để xây dựng trong Mã VS. Ví dụ: nếu bạn muốn biên dịch mã C/C++ trên Windows, hãy cài đặt hai plug-in C/C++ và C/C++ Compile Run, cài đặt TDM-GCC và bắt đầu từ môi trường có thể biên dịch đơn giản [1][2]. Khi được sử dụng cùng với Công cụ CMake, một phần bổ trợ CMake do Microsoft cung cấp chính thức, nó có thể được sử dụng làm IDE để quản lý phụ thuộc, tự động hoàn thành chặng hạn như Intellisense, gỡ lỗi, kiểm tra và triển khai cùng một lúc.
- Nó là một trình soạn thảo văn bản và IDE phổ biến đã chứng kiến sự gia tăng đáng kể trong việc sử dụng và chia sẻ VSC trong những năm gần đây.



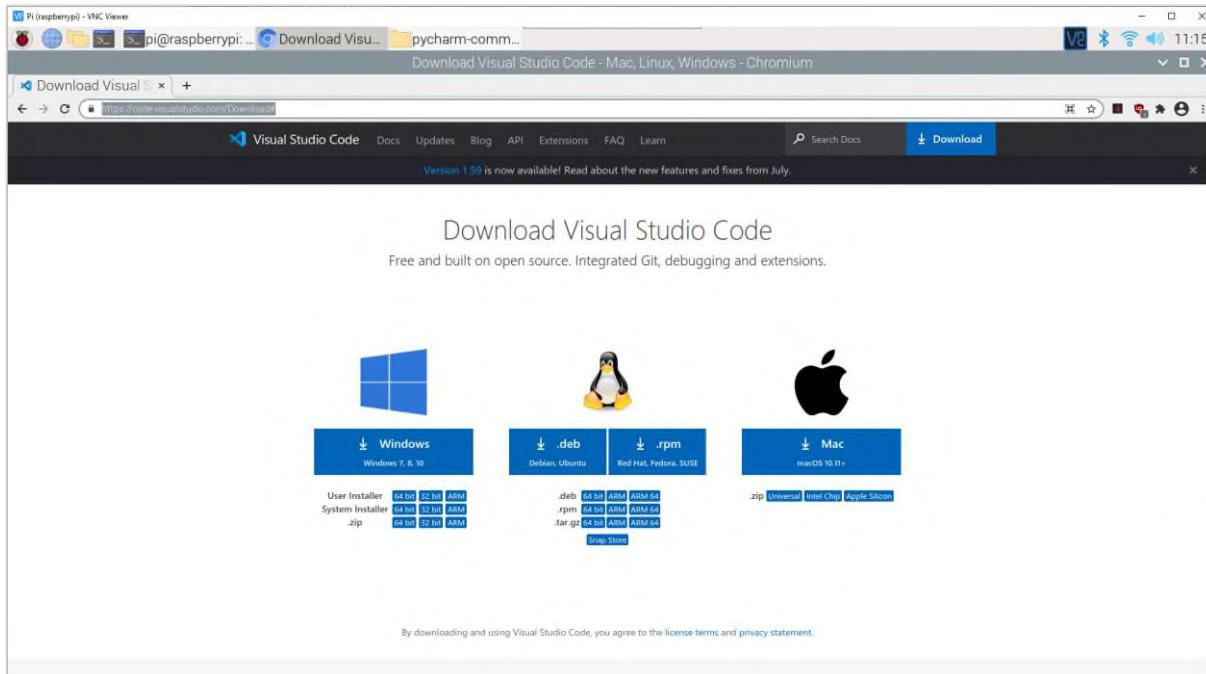
Bài 4

## Trình chỉnh sửa và IDE

- | 4.1. Trình soạn thảo văn bản
- | 4.2. Nano
- | 4.3. IDE
- | 4.4. VSC**

# VSC

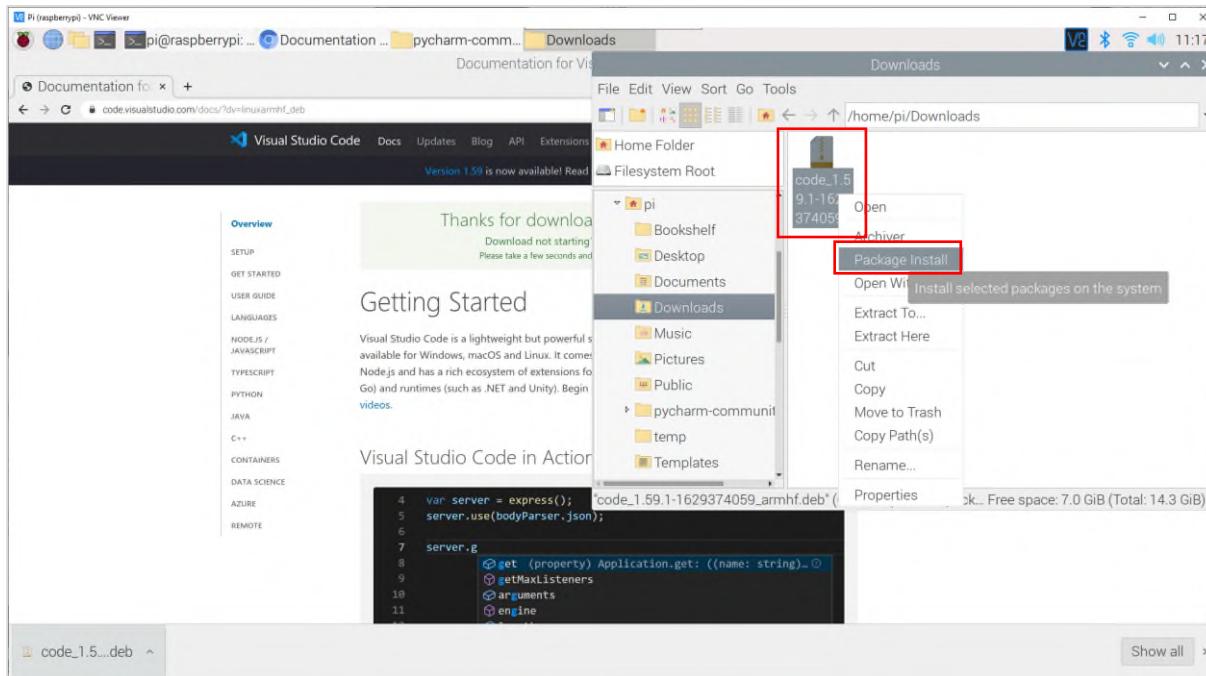
## Cách sử dụng VSC



- ▶ Để cài đặt VSC trong HĐH Raspberry Pi, trước tiên hãy kết nối với [#](https://code.visualstudio.com/Download) và tải xuống phiên bản ARM của .deb.

# VSC

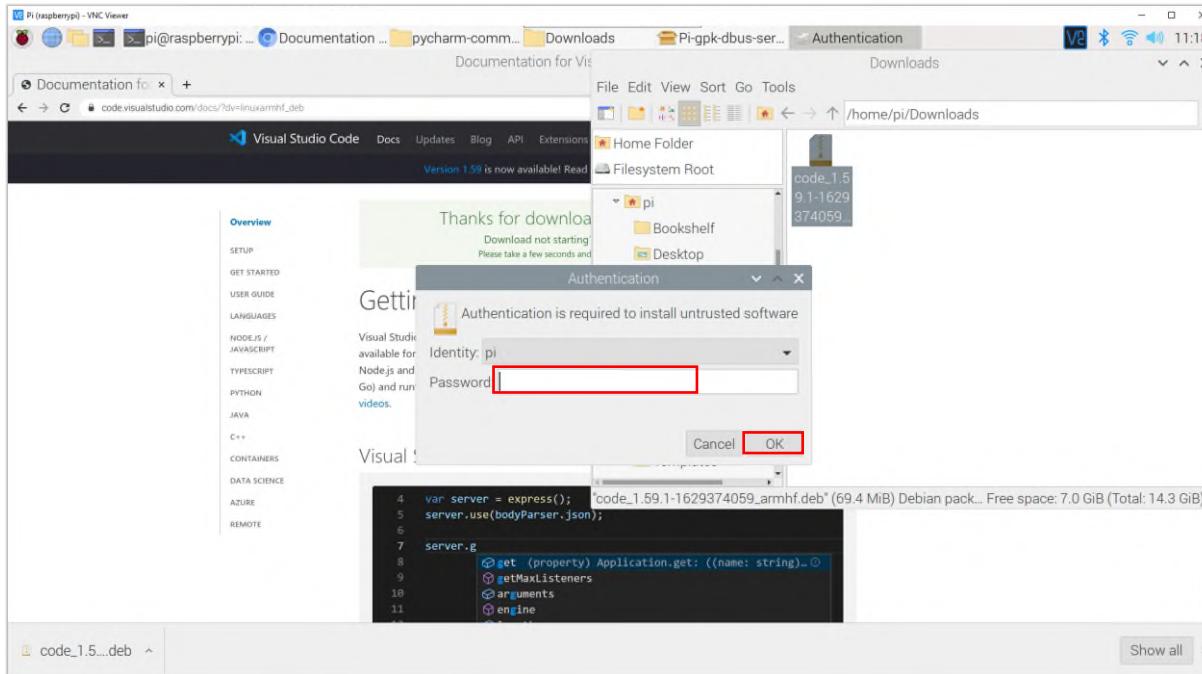
## | Cách sử dụng VSC



- ▶ Nhấp chuột phải vào tệp đã tải xuống trong /home/pi/Tải xuống, sau đó nhấp vào **Package install**.

# VSC

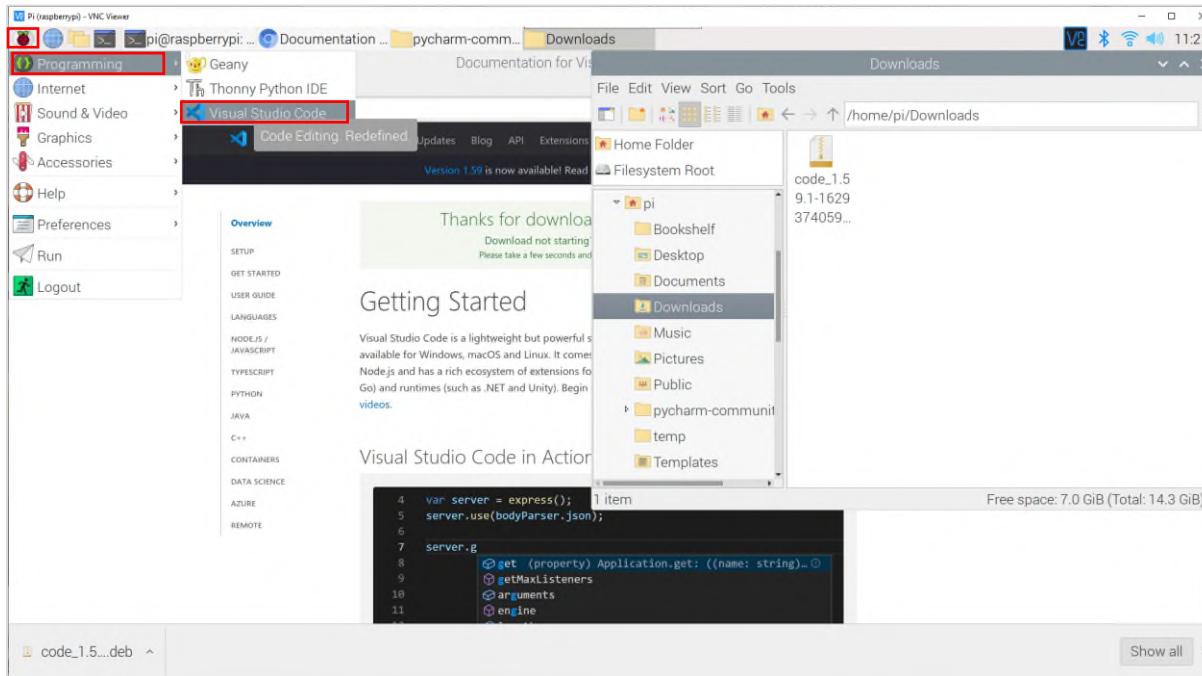
## | Cách sử dụng VSC



- ▶ Nhập mật khẩu của Raspberry Pi rồi nhấn **OK**.

# VSC

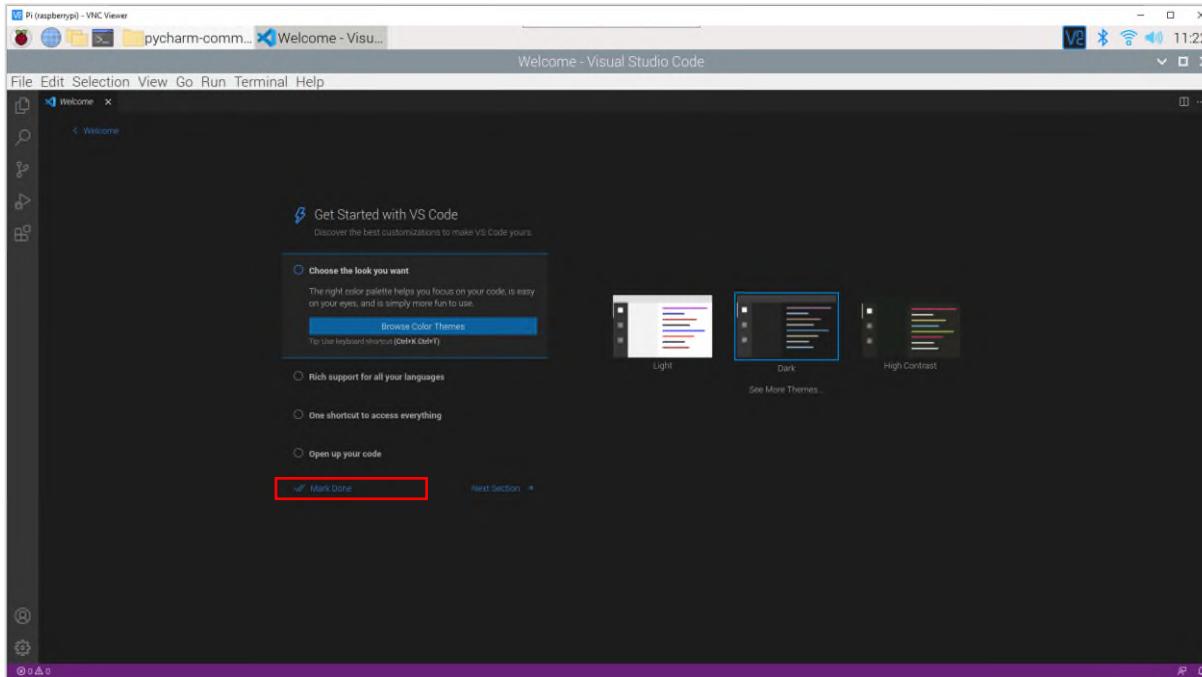
## | Cách sử dụng VSC



- ▶ Khi quá trình cài đặt hoàn tất, Raspberry Pi menu → Lập trình → Visual Studio Code được tạo.
- ▶ Bấm để chạy.

# VSC

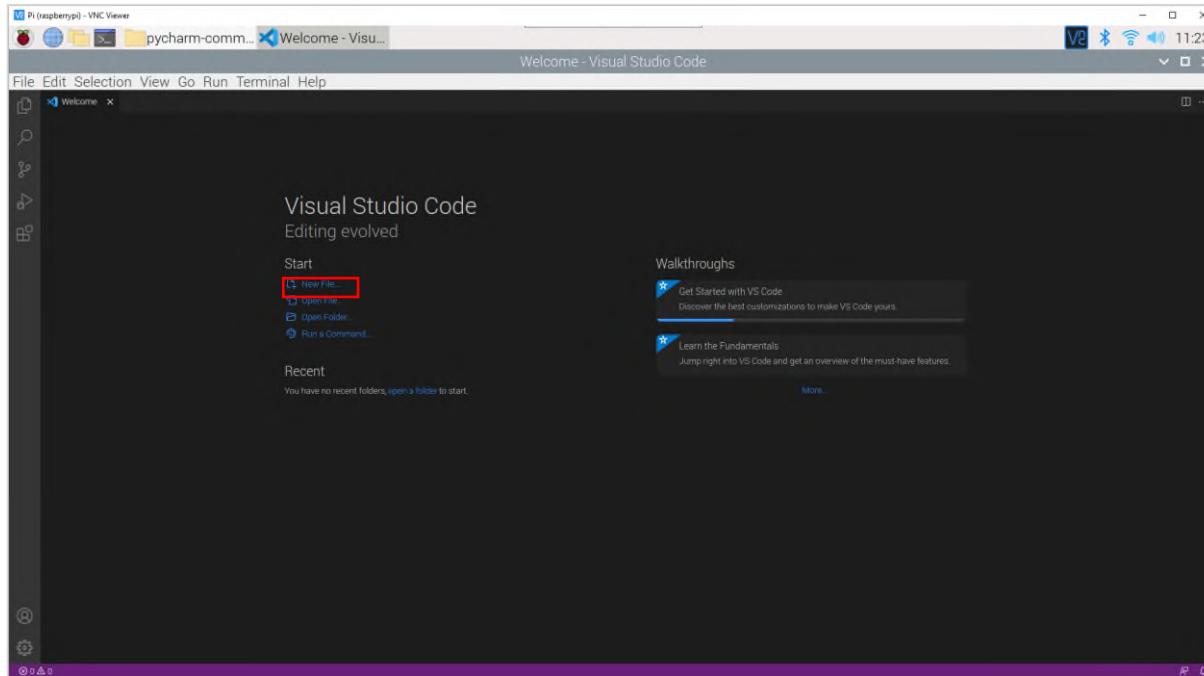
## Cách sử dụng VSC



- ▶ Có các mục cài đặt đơn giản.
- ▶ Nhấp vào **Mark done** rồi Bỏ qua.

# VSC

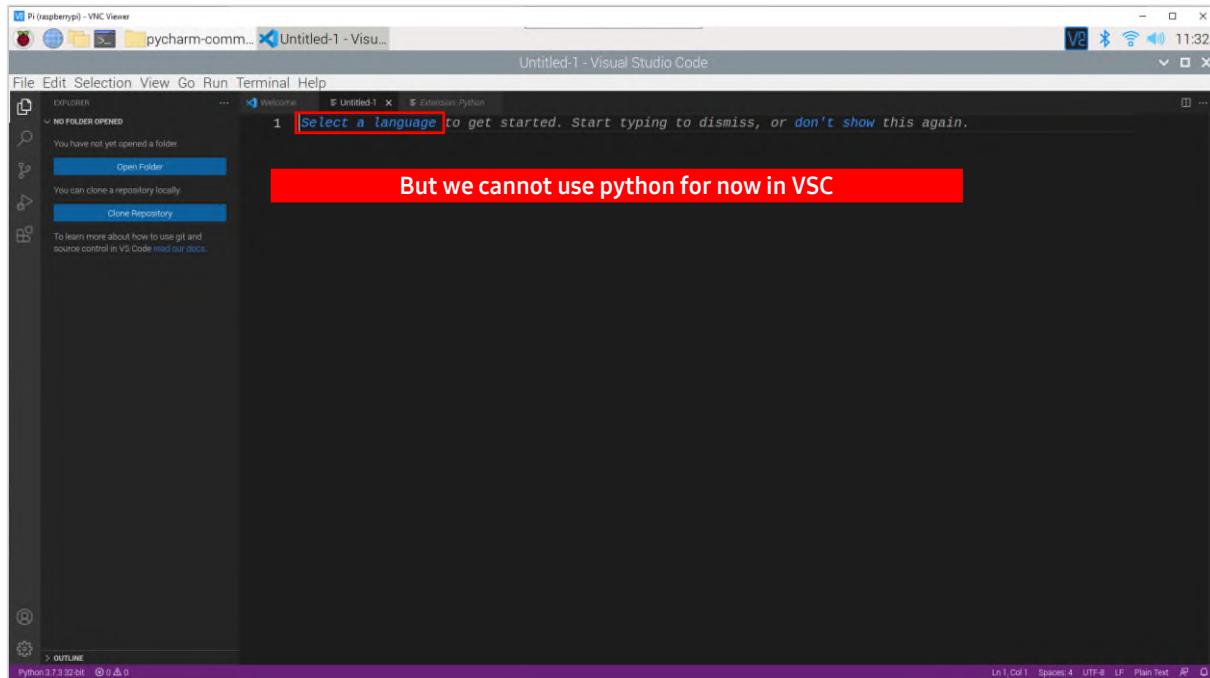
## Cách sử dụng VSC



▶ Nhấp vào **New file** để tạo tệp mới.

# VSC

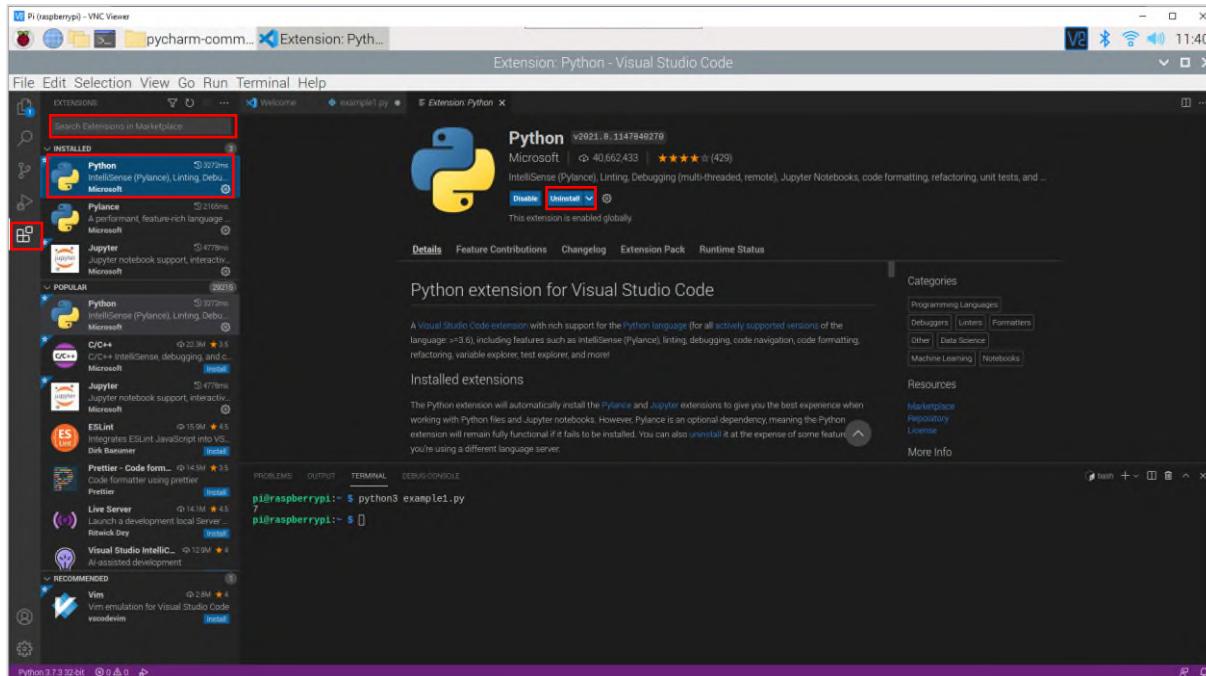
## Cách sử dụng VSC



- ▶ Thông báo Select a language xuất hiện.
- ▶ Tuy nhiên VSC thực chất là một Text editor nên nó chỉ có chức năng soạn thảo văn bản nên phải cài thêm Python.

# VSC

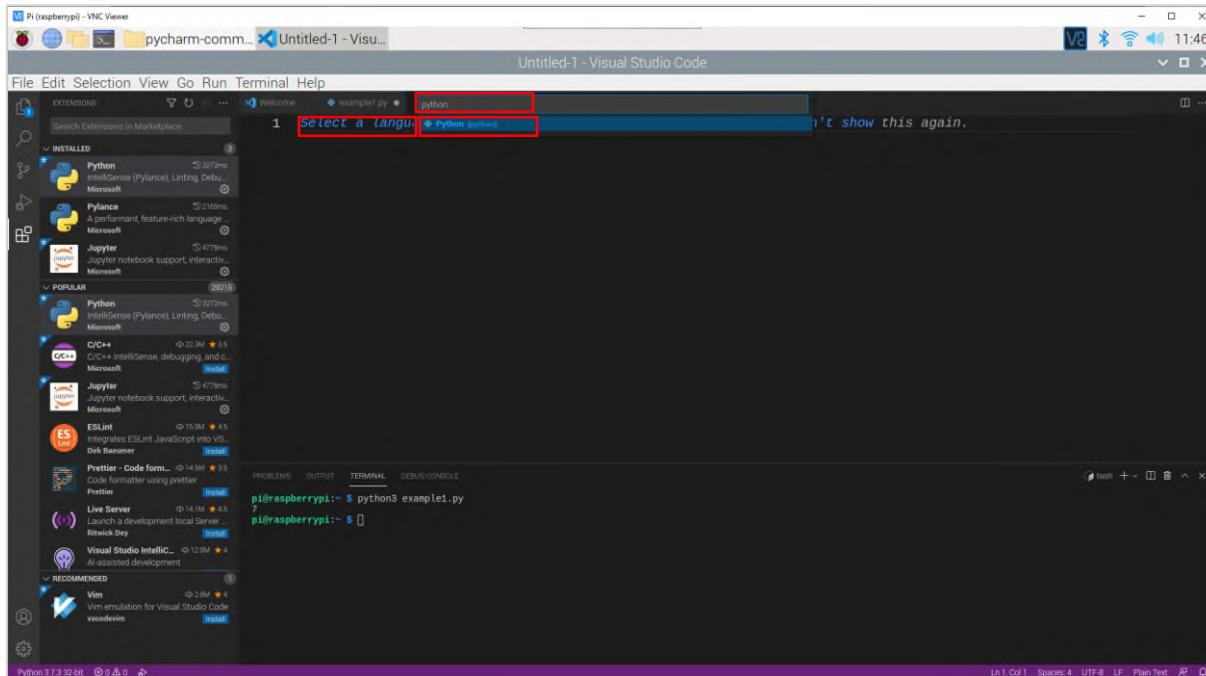
## Cách sử dụng VSC



▶ Nhấp vào **Tiện ích mở rộng tìm kiếm** trong **Marketplace** ở bên trái và nhấp vào **Python** → Nhấp vào **Cài đặt** để cài đặt Python.

# VSC

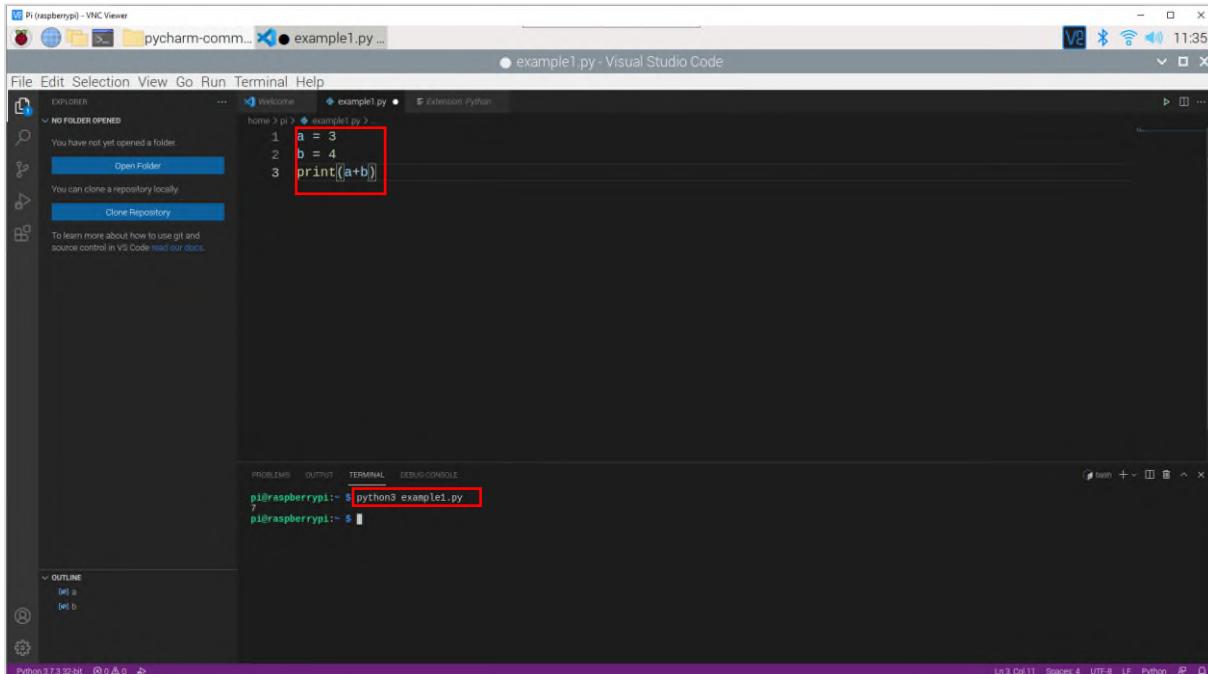
## Cách sử dụng VSC



- ▶ Vì Python đã được cài đặt, hãy đặt ngôn ngữ là Python.
- ▶ Nhấp vào **Select a language** → Nhập python rồi chọn.

# VSC

## Cách sử dụng VSC



- ▶ Tạo một tệp có tên example1.py và viết mã.
- ▶ Viết xong nhấn Ctrl + s để lưu lại và đặt tên là example1.py.
- ▶ Nhấn tab Terminal ở trên cùng để tạo terminal mới và khi bạn nhập lệnh python3 example.py trong terminal, bạn có thể thấy kết quả được hiển thị.

# Kết thúc bài học

A blurred background image shows a person's hands holding a white paper coffee cup with a black lid. In the foreground, a laptop keyboard and a smartphone are visible on a desk. The overall color palette is blue and grey.

# Together for Tomorrow! Enabling People

Education for Future Generations

©2022 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of book.

This book is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this book other than the curriculum of Samsung Innovation Campus or to use the entire or part of this book, you must receive written consent from copyright holder.