

BÀI 2: LÀM QUEN VỚI MÔI TRƯỜNG LINUX

1. MỤC TIÊU BÀI TN

Trong bài TN này SV sẽ thực hành những phần sau:

- Thực hành những câu lệnh thông dụng của Linux
- Viết một chương trình trên linux bằng ngôn ngữ C
- Biên dịch chương trình sử dụng câu lệnh và makefile
- Chạy một chương trình trên linux
- Thiết lập để chương trình tự động chạy khi khởi động Raspberry

2. CÁC CÂU LỆNH THÔNG DỤNG TRÊN LINUX

SV thực hành những câu lệnh bên dưới trước khi làm bài tập ví dụ.

2.1 Nhóm các câu lệnh cơ bản

- **apt-get update**: cập nhật phiên bản Raspbian.
- **apt-get upgrade**: Nâng cấp toàn bộ các gói phần mềm đã cài đặt trên Raspbian.
- **uname -r**: kiểm tra phiên bản linux của Raspbian
- **man <câu lệnh>**: xem hướng dẫn của câu lệnh
- **clear**: Xóa màn hình terminal và các lệnh bạn đã thực thi trước đó
- **date**: Hiển thị ngày giờ hiện tại
- **raspi-config**: Mở menu thiết lập cấu hình Raspberry Pi hay Raspbian
- **reboot**: Khởi động lại Raspberry Pi
- **poweroff**: Tắt Raspberry Pi
- **shutdown -h now**: Tắt Raspberry Pi ngay lập tức
- **shutdown -h 01:22**: Tắt Raspberry Pi vào lúc 1:22 AM.

2.2 Nhóm các câu lệnh liên quan đến file, folder

- **pwd**: vị trí của thư mục hiện tại
- **ls -l**: hiển thị danh sách các file trong folder với các thông tin file size, ngày sửa, quyền hạn của file
- **cd [địa chỉ thư mục]**: chuyển đến thư mục mới

- **mkdir example_directory**: Tạo folder tên example_directory bên trong folder hiện tại
- **touch**: Tạo file trắng mới trong folder hiện tại.
- **cp xxx**: Copy file hoặc thư mục xxx và paste vào vị trí mới.
- **mv xxx**: Move file xxx sang vị trí mới
- **rm xxx**: xóa file tên xxx
- **rmdir example_directory**: Xóa thư mục example_directory
- **chmod [quyền truy cập] [file]**: cấp quyền truy cập cho file

2.3 Nhóm các câu lệnh liên quan mạng

- **ifconfig**: Kiểm tra tình trạng mạng hiện tại trên Raspberry Pi và địa chỉ IP hiện tại của Raspberry Pi.
- **ping**: Kiểm tra kết nối giữa các thiết bị trong mạng.
- **wget http://www.website.com/example.txt**: Tải file example.txt từ trên mạng về và lưu vào folder hiện tại.

2.4 Nhóm các câu lệnh hệ thống

df -h: Hiển thị thông tin dung lượng lưu trữ còn trống.

free: Hiển thị lượng RAM còn trống.

lsusb: Liệt kê các thiết bị USB đang cắm vào Raspberry Pi

2.5 Bài tập (thực hiện trên terminal)

- Tạo folder mới có tên “Hello” trong thư mục /home/pi.
- Tạo một file mới có tên “new.c” trong thư mục /home/pi.
- Move file này vào trong thư mục hello.
- Cấp quyền write cho file new.c
- Mở file new.c bằng Geany
- Nhập đoạn code sau vào file new.c sau đó lưu lại
- Đổi tên file new.c thành hello.c

```
#include <stdio.h>

int main(int argc, char **argv)
{
    printf("Hello World\n");
    return 0;
}
```

3. LẬP TRÌNH C TRÊN RASPBERRY

3.1 Xây dựng chương trình với gcc/g++

Các bước xây dựng một chương trình C trên Linux/Raspberry:

- Viết code: Code chương trình có thể viết trực tiếp trên Raspberry. Tuy nhiên để cho thuận tiện thường các lập trình viên sẽ viết code trên máy tính và đổ xuống raspberry để kiểm tra.
- Biên dịch chương trình: Raspbian tích hợp sẵn các trình biên dịch cho ngôn ngữ C và C++. Để biên dịch một chương trình C/C++ ta dùng các câu lệnh sau:

```
gcc file.c -o tên_chương_trình      // biên dịch code C
g++ file.cpp -o tên_chương_trình    // biên dịch code C++
```

ví dụ: để biên dịch file hello.c bên trên thành một chương trình có tên HELLO ta gõ lệnh như sau: gcc hello.c -o HELLO

- Chạy chương trình: để chạy một chương trình trên linux, ta dùng cấu trúc lệnh “./tên_chương_trình”. Ví dụ: ./HELLO Lưu ý: - có dấu ./ trước tên chương trình.
- - một số chương trình cần được cấp quyền bằng lệnh chmod trước khi chạy

```
pi@raspberrypi:~$ gcc hello.c -o HELLO
pi@raspberrypi:~$ chmod a+x HELLO
pi@raspberrypi:~$ ./HELLO
Hello, World!
pi@raspberrypi:~$ █
```

Bài tập ví dụ:

Biên dịch code bên dưới thành chương trình tên demo.exe

```
// hello.cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!" << endl;
    return 0;
}
```