

# **HỆ ĐIỀU HÀNH**

## **BÀI THỰC HÀNH 1**

### **TỔNG QUAN HỆ ĐIỀU HÀNH**

#### **Hướng dẫn báo cáo thực hành:**

1. Tham khảo slide bài giảng về các giải thuật điều phối, giải thuật thay trang nhớ, các đặc trưng quản lý bộ nhớ, cấp phát bộ nhớ.
2. Báo cáo 2 số liệu Input/Output đối với các bài tập về lập trình mô phỏng giải thuật.
3. Các chương trình truy cập bộ nhớ (chính và phụ) cần chạy trong chương trình Turbo C.
4. Ứng với mỗi số liệu Input/Output của chương trình có đối chiếu tính toán bằng tay về độ đúng đắn của việc hiện thực giải thuật.
5. Khi tính toán bằng tay cần vẽ biểu đồ tương ứng (biểu đồ Gantt trong quản lý tiến trình; quá trình nạp trang nhớ trong quản lý bộ nhớ).
6. Bài báo cáo trình bày bằng Microsoft Word.
7. Bài báo cáo theo mỗi cá nhân sinh viên.
8. Thời gian nhận báo cáo: **2 ngày** sau buổi thực hành.

#### **Mục tiêu:**

1. Cài đặt được máy ảo
2. Cài đặt được các hệ điều hành cơ bản.
3. Hiểu được cơ chế quản lý tiến trình của hệ điều hành.
4. Hiểu được cơ chế quản lý file của hệ điều hành.

#### **Nội dung:**

1. Cài phần mềm VMWare (nếu máy chưa có)
2. Cài hệ điều hành Windows XP (Windows 7) trong VMWare. Trình bày các bước cài đặt. Tìm hiểu chức năng 10 services của hệ điều hành vừa cài.
3. Cài hệ điều hành Linux trong VMWare.

#### **Hướng dẫn thực hành:**

##### **A. Thực hành các lệnh cơ bản của Linux**

##### **I. Quản lý tập tin**

1. **cp file thư\_mục:** chép file vào thư mục “thư\_mục”
2. **rm file:** xóa bỏ tập tin trong thư mục hiện hành

3. **rm -rf thư\_mục**: xóa bỏ thư mục mang tên “thư\_mục” với tất cả các tập tin trong đó (force)
4. **find thư\_mục name\_file**: tìm tập tin mang tên “file” trong thư mục “thư\_mục” kể cả trong các thư mục con
5. **diff file1 file2**: so sánh nội dung của 2 tập tin hoặc của 2 thư mục
6. **rmdir thư\_mục**: xóa bỏ thư mục trống mang tên “thư\_mục”
7. **mkdir thư\_mục**: tạo ra thư mục “thư\_mục”
8. **mv thư\_mục1 thư\_mục2**: chuyển tên “thư\_mục1” thành “thư\_mục2”
9. **cp -r thư\_mục1 thư\_mục2**: chép toàn bộ nội dung của thư mục “thư\_mục1” sang thư mục “thư\_mục2”
10. **rsync -a thư\_mục1 thư\_mục2**: chép toàn bộ nội dung của thư mục “thư\_mục1” sang thư mục “thư\_mục2”
11. **mkdir -p thư\_mục1/thư\_mục2**: tạo ra thư mục cha “thư\_mục1” và thư mục con “thư\_mục2” cùng lúc.

## II. Di chuyển/liệt kê các tập tin

12. **cd**: di chuyển sang thư mục “/home/người\_dùng”
13. **cd ~/Desktop**: di chuyển sang thư mục “/home/người\_dùng/Desktop”
14. **cd ..**: di chuyển sang thư mục cha (ngay trên thư mục hiện hành)
15. **ls -l Thư\_mục**: liệt kê danh mục tập tin trong thư mục “Thư\_mục” một cách chi tiết
16. **dir -l Thư\_mục**: liệt kê danh mục tập tin trong thư mục “Thư\_mục” một cách chi tiết
17. **ls -d**: liệt kê tên các thư mục nằm trong thư mục hiện hành
18. **dir -d**: liệt kê tên các thư mục nằm trong thư mục hiện hành

## III. Quản trị hệ thống

19. **shutdown -h now**: khởi động lại máy tính ngay lập tức
20. **uname -r**: cho biết phiên bản của nhân Linux
21. **lsusb**: liệt kê các thiết bị usb có mặt trong máy tính
22. **lspci**: liệt kê các thiết bị pci có mặt trong máy tính
23. **sudo command**: thực hiện lệnh “command” với tư cách người siêu dùng (root)
24. **gksudo command**: giống với sudo, nhưng dùng cho các ứng dụng đồ họa
25. **time command**: cho biết thời gian cần thiết để thực hiện xong lệnh “command”

26. **clear**: xoá màn hình của cửa sổ Terminal

#### IV. Gói phần mềm

27. **apt-get install soft**: cài phần mềm “soft” đồng thời giải quyết các gói phần mềm phụ thuộc

28. **apt-get remove soft**: loại bỏ phần mềm “soft” cũng như tất cả các gói phần mềm trực thuộc

29. **apt-file list soft**: liệt kê các tập tin có trong gói phần mềm “soft”

30. **alien -di paquet.rpm**: chuyển phần mềm “paquet.rpm” thành gói phần mềm dạng Debian “paquet.deb” (-d) và thực hiện cài đặt luôn (-i)

31. **dpkg -i paquet.deb**: cài đặt phần mềm “paquet.deb” (không giải quyết các gói phụ thuộc)

#### V. Phân vùng ổ cứng

32. **fdisk -l**: hiện thị các phân vùng tích cực

33. **fdisk /dev/hda1**: tạo mới và bỏ phân vùng trên ổ cứng IDE thứ nhất

34. **mkfs.vfat /dev/hda1**: tạo một hệ thống tập tin “fat32” trên phân vùng “/dev/hda1”

#### B. Thực hành các lệnh nâng cao của Linux

##### Các lệnh liên quan đến tiến trình

- Một số thuật ngữ:
  - UID: số uid của người chủ tiến trình
  - PID: số của tiến trình (process identity)
  - PPID: số của tiến trình cha (parent process identity)
  - C: chỉ số sử dụng bộ xử lý (processor utilization for scheduling)
  - STIME: thời điểm bắt đầu tiến trình
  - TTY: terminal điều khiển tiến trình
  - TIME: thời gian tích lũy thực hiện tiến trình (cumulative time)
  - COMMAND: tên lệnh sinh ra tiến trình

Lệnh	Ý nghĩa	Ví dụ
ps	Hiện thị các tiến trình đang chạy	\$ ps
ps -ag	Hiện thị thông tin về tất cả các tiến trình đang chạy	\$ ps -ag
ps aux	Hiện thị các tiến trình và người dùng tạo ra tiến trình	\$ ps aux
kill {PID}	Stop tiến trình PID	\$ kill 1012
killall {Process name}	Stop tiến trình có tên Process name	\$ killall httpd
kill 0	Stop tất cả các tiến trình ngoại trừ shell của người dùng	\$ kill 0
free	Thống kê việc sử dụng bộ nhớ	\$ free
pstree	Hiện thị cây tiến trình	\$ pstree
[ctrl-z]	Làm treo (tạm ngưng) một tiến trình đang chạy	ấn phím [ctrl-z]

bg	Chuyển quá trình đang chạy sang chế độ chạy ngầm (background). Cần phải treo tiến trình trước khi thực hiện nó	\$ bg
fg	Chuyển quá trình đang chạy ngầm (background) sang chế độ chạy bình thường (foreground).	\$ fg
jobs	Liệt kê các tiến trình chạy ngầm	\$ jobs
top	Hiển thị theo thời gian thực các tiến trình hiện đang chạy và các thông tin khác có liên quan: bộ nhớ, sử dụng CPU	\$ top

## 2. Quản lý tiến trình ngầm

- Tạo tiến trình chạy ngầm (background): \$lệnh &
- Quản lý các tiến trình đang chạy ngầm: \$set -m
- Hiển thị trạng thái của các tiến trình ngầm: \$jobs -l
- Tiếp tục thực hiện các tiến trình ngầm sau khi cắt liên lạc với terminal: \$nohup lệnh &  
Các số liệu của tiến trình đưa ra stdout và stderr sẽ được ghi lên tập tin nohup.out
- Đợi kết thúc tiến trình: \$wait PID
- Hủy tiến trình: **\$kill PID** hoặc **\$kill %n**: diệt tiến trình theo số thứ tự của nó trong danh sách các tiến trình chạy ngầm

### Bài tập 1

Thực hiện tất cả các lệnh trên

### Bài tập 2

Sử dụng câu lệnh ps, thực hiện các yêu cầu sau:

1. Hiển thị tất cả các tiến trình của người dùng hiện đang làm việc
2. Hiển thị tất cả các tiến trình hiện đang chạy
3. Đọc và hiểu các output trên

### Bài tập 3

Liệt kê các tiến trình ngầm hiện đang chạy

### Bài tập 4

1. Thực thi lệnh top
2. Treo tiến trình đó

## **C. Thực hành các lệnh cơ bản của Windows**

### **I. Quản lý tập tin**

1. **DIR [drive:][path][filename]**: Xem thông tin tập tin, thư mục
2. **DEL [drive:][path][filename]**: xóa bỏ tập tin, thư mục
3. **Copy file thư mục**: Copy file đến “thư mục”

### **II. Quản trị hệ thống**

4. **tracert ip/host**: Xem đường đi của gói tin
5. **Net send ip/host thông\_điệp\_muốn\_gửi**: Gửi thông điệp đến máy khác
6. **Netstat [/a][/e][/n]**: liệt kê các kết nối với máy tính của mình
  - a. **/a**: **Hiển thị tất cả các kết nối và các cổng đang lắng nghe (listening)**
  - b. **/e**: **hiển thị các thông tin thống kê Ethernet**
  - c. **/n**: **Hiển thị các địa chỉ và các số cổng kết nối**
7. **ipconfig /all**: **hiển thị cấu hình IP của máy tính**
8. **systeminfo**: **thị thông tin của hệ thống từ cấu hình cơ bản, loại CPU, khu vực địa lý, ...**
9. **Shutdown [-m \\ip] [-t xx] [-i] [-l] [-s] [-r] [-a] [-f] [-c “comment”]**: Shutdown máy tính
  - **-m\\ip**: Shutdown máy tính có địa chỉ ip
  - **-t xx** : đặt thời gian cho việc thực hiện lệnh shutdown
  - **-l** : logg off (lưu ý ko thể thực hiện khi remote)
  - **-s** : shutdown
  - **-r** : shutdown và restart
  - **-a** : không cho shutdown
  - **-f** : shutdown mà ko cảnh báo
  - **c “comment”** : lời cảnh báo trước khi shutdown
10. **bootcfg**: Đọc thông số khởi động máy

### **III. Phân vùng ổ cứng**

11. **chkdsk [volume]**: Kiểm tra và sửa lỗi ổ cứng volume
12. **format [volume]**: Format ổ cứng volume

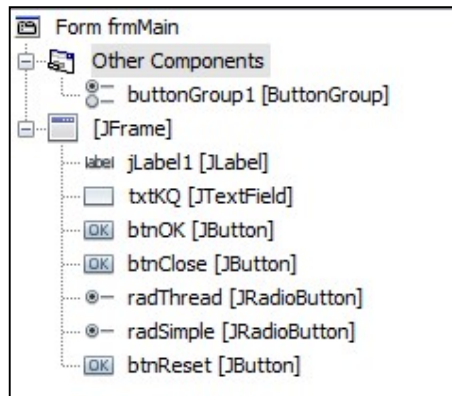
## **D. Tạo tiến trình, tiểu trình và thiết lập độ ưu tiên bằng ngôn ngữ lập trình**

### **I. Tạo tiến trình, tiểu trình và thiết lập độ ưu tiên bằng Java**

Dùng Netbeans thiết kế chương trình như sau:

a) Thiết kế giao diện:

Dùng các component như hình 1 để thiết kế giao diện như hình 2.



Hình 1: Các component dùng trong chương trình

Hình 2: Giao diện chương trình

b) Tạo lớp mở rộng đối tượng Thread

```
1 package threadtest;
2 import static java.lang.Thread.sleep;
3 import javax.swing.JOptionPane;
4 import javax.swing.JTextField;
5 public class ThreadTest extends Thread {
6     JTextField txtKQ;
7     public ThreadTest(JTextField t) {txtKQ=t;}
8     public void draw(int i) {txtKQ.setText(String.valueOf(i));}
9     public void reset() {txtKQ.setText("");}
10    public void run() {
11        for(int i=0;i<100;i++){
12            try{
13                sleep(100);
14            }
15            catch (InterruptedException ex) {
16                JOptionPane.showMessageDialog(null, ex.toString(),
17                    "Thông báo lỗi", JOptionPane.ERROR_MESSAGE);
18            }
19            draw(i);
20        }
21    }
22 }
```

c) Viết lệnh cho phím OK

```
private void btnOKActionPerformed(java.awt.event.ActionEvent evt) {  
  
    if(radThread.isSelected())  
    {  
        ThreadTest b = new ThreadTest(txtKQ);  
        b.setPriority(Thread.MAX_PRIORITY);  
        b.start();  
    }  
    else  
    {  
        for(int i=0;i<100;i++)  
        {  
            try {  
                Thread.sleep(100);  
            } catch (InterruptedException ex) {  
                Logger.getLogger(frmMain.class.getName()).log(Level.SEVERE, null, ex);  
            }  
            txtKQ.setText(String.valueOf(i));  
        }  
    }  
  
}
```

d) Viết lệnh cho phím Reset:

```
private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {  
    txtKQ.setText("");  
}
```

e) Viết lệnh cho phím Close

```
private void btnCloseActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```

## II. Thiết lập độ ưu tiên bằng C++

Dùng Dev C++ viết chương trình như hình dưới:

```
1 #include <cstdlib>  
2 #include <iostream>  
3 #include <Windows.h>  
4 #include <conio.h>  
5 using namespace std;  
6 int main ()  
7 {  
8     using namespace std;  
9     cout<<"Enter length of array"<<endl;  
10    int size;  
11    cin>>size;  
12    int * ps=new int [size];
```

```

13     for (int i=0; i<size; i++)
14         ps[i]=3;
15     int k=4;
16     SetPriorityClass(GetCurrentProcess(), HIGH_PRIORITY_CLASS);
17     while (k>=0)
18     {
19         for (int bi=0; bi<size; bi++)
20             std::cout<<ps[bi];
21         std::cout<<std::endl;
22         int i=size-1;
23         if (ps[i]==3){
24             ps[i]=4;
25             continue;
26         }
27         if (ps[i]==4){
28             while (ps[i]==4){
29                 ps[i]=3;
30                 --i;
31             }
32             ps[i]=4;
33             if (i<k)
34                 k--;
35         }
36     }
37     return 0;
38 }

```

### III. Nhận xét

Thực thi chương trình viết bằng Java và C++. Thay đổi các chế độ khác nhau khi có hoặc không có độ ưu tiên. Dùng chương ProceXP để quan sát và nêu nhận xét.