



**fit@hcmus**

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG - HCM  
KHOA CÔNG NGHỆ THÔNG TIN

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
ĐHQG – TP.HCM  
KHOA CÔNG NGHỆ THÔNG TIN**

# **BÁO CÁO**

**HỆ ĐIỀU HÀNH  
20CLC06**

**Giảng viên hướng dẫn**

Nguyễn Văn Giang

Lê Quốc Hoà

# MỤC LỤC

<b>1</b>	<b>Thông tin nhóm .....</b>	<b>3</b>
<b>2</b>	<b>NỘI DUNG ĐỒ ÁN .....</b>	<b>4</b>
2.1.	Cài đặt NachOS trên nền tảng linux .....	4
2.2.	Cơ chế thực hiện, quy trình thực thi của một chương trình trên nachos .....	5
1)	Cơ chế thực hiện, quy trình thực thi của một chương trình trên NachOS ..	5
2)	Các bước cập nhật thanh ghi .....	5
2.3.	CÀI ĐẶT TỔNG QUAN .....	6
1)	Các bước tạo một System call .....	6
2)	Lớp SynchronConsole ở ../code/threads/ trong file system.h và system.cc .....	7
3)	System2User và User2System ở ../code/userprog trong file exception.cc .....	7
2.4.	CÀI ĐẶT SYSTEM CALL VÀ EXCEPTION .....	8
1)	Cài đặt hàm IncreasePC() .....	8
2)	Cài đặt System call ReadNum .....	8
3)	Cài đặt System Call PrintNum .....	9
4)	Cài đặt System Call ReadChar .....	9
5)	Cài đặt System Call PrintChar .....	9
6)	Cài đặt System Call RandomNum .....	9
7)	Cài đặt System Call ReadString .....	9
8)	Cài đặt System Call PrintString .....	9
9)	Chương trình help .....	10
10)	Chương trình ascii .....	10
11)	Chương trình sortBubble .....	10
2.5.	TÀI LIỆU THAM KHẢO .....	11

# 1 Thông tin nhóm

STT	MSSV	Họ và tên
1	20127278	Nguyễn Văn Phú
2	20127480	Nguyễn Đức Duy
3	20127493	Nguyễn Văn Hậu (Nhóm trưởng)

## 2 NỘI DUNG ĐỒ ÁN

### 2.1. Cài đặt NachOS trên nền tảng linux

- Bước 1: Cài đặt Linux trên máy ảo
- Bước 2 : Tải và giải nén file **nachos.rar**
- Bước 3 : Vào thư mục “./nachos/cross-compiler/decstation-ultrix/bin/”. Thiết lập thuộc tính “executable” ở mọi người dùng đối với tất cả các file có trong thư mục (as, cc1, cpp0, gcc, ld).
- Bước 4 : Mở **Geany** (phần mềm soạn thảo văn bản đơn giản), mở file **Makefile** ở đường dẫn “./nachos/nachos-3.4/code/test”.
- Bước 5 Sửa dòng “MAKE = gmake” thành “MAKE = make” và lưu lại.

- Bước 6 : Mở **LXTerminal**. Gõ lệnh “**cd ~/nachos/nachos-3.4/code**” rồi Enter.
- Bước 7 :Tiếp tục gõ “**make all**” để tiến hành cài đặt NachOS (Quá trình cài đặt sẽ tiến hành trong ít phút).
- Bước 8: Chạy thử chương trình trên NachOS bằng lệnh : **./userprog/nachos -rs 1023 -x ./test/halt** rồi Enter.

Kết quả chương trình sẽ hiện như sau:

*Machine halting!*

*Ticks: total xx idle xx, system xx, user xx*

*Disk I/O: reads 0, writes 0*

*Console I/O: reads 0, writes 0*

*Paging: faults 0*

*Network I/O: packets received 0, sent 0 Cleaning up...*

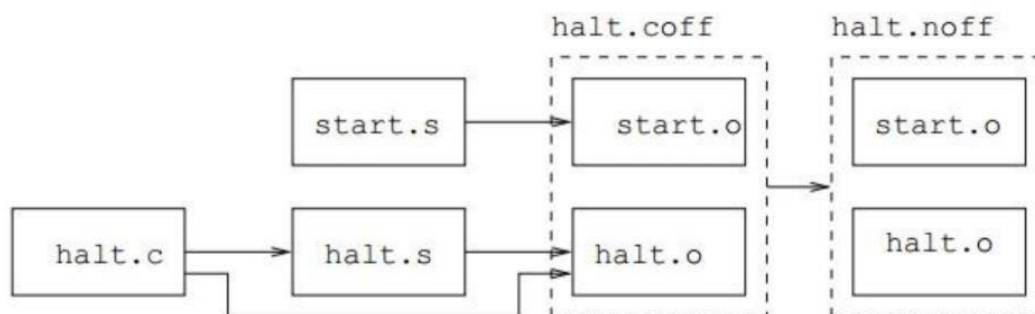
## 2.2 CƠ CHẾ THỰC HIỆN, QUY TRÌNH THỰC THI CỦA MỘT CHƯƠNG TRÌNH TRÊN NACHOS

### 1) Cơ chế thực hiện, quy trình thực thi của một chương trình trên NachOS

Một chương trình (ví dụ: *Halt.c*) muốn được thực thi thì nó cần phải được biên dịch quá trình biên dịch trên NachOS gồm ba bước:

- 1. Chương trình *halt.c* được cross-compiler biên dịch thành tập tin *halt.s* là mã hợp ngữ chạy trên kiến trúc MIPS
- 2. Tập tin *halt.s* được liên kết với *starts.s* để tạo thành tập tin *halt.coff* (bao gồm *halt.o* và *start.o*) là dạng file thực thi trên linux với kiến trúc MIPS
- 3. Tập tin *halt.coff* được tiện ích *coff2noff* được chuyển thành tập tin *halt.noff* dạng file thực thi trên NachOS kiến trúc MIPS

Quá trình này được mô tả bằng hình dưới đây:



### 2) Các bước cập nhật thanh ghi

**Bước 1:** Mã của system call được đưa vào thanh ghi **r2**.

**Bước 2:** Các biến người dùng sử dụng được đưa vào thanh ghi **r4, r5, r6**.

**Bước 3:** Giá trị trả về của system call được đưa vào thanh ghi **r2**

## 2.3 CÀI ĐẶT TỔNG QUAN

### 1) Các bước tạo một System call

#### Bước 1: Tạo macro:

Tránh nhầm lẫn việc gọi hàm, cần define con số cụ thể trong kernel space thành một macro ->

Vào `../code/userprog/syscall.h`: **#define <tên system call> <số cụ thể>**

– VD: System Call “**ReadInt**”:

**#define SC\_ReadInt 11**

#### Bước 2: Định nghĩa hàm trong file assembler

Ta mở 2 files: `code/test/start.c` và **`code/test/start.s`**:

Và chèn đoạn mã vào:

```
.globl <tên hàm>
.ent <tên hàm>
<tên hàm>:
addiu $2,$0,<tên system call>
syscall
j $31
.end <tên hàm>
```

VD: System Call “**ReadInt**”

```
.globl ReadInt
.ent ReadInt
ReadInt:
addiu $2, $0, SC_ReadInt
syscall
j $31
.end ReadInt
```

#### Bước 3: Định nghĩa cụ thể cho một công việc

Mở file `usrprog/exception.cc`

Chuyển đoạn mã “**if.....else**” sang đoạn mã “**switch....case**” với các **case** là các **syscall** cần gọi.

#### Bước 4 : Viết chương trình ở mức người dùng để kiểm tra file `.c` `./code/test`

Sử dụng hàm như đã khai báo prototype ở `/code/userprog/syscall.h`

#### Bước 5 :Thêm đoạn vào Makefile trong `/code/test/` Thêm tên file chạy chương trình vào

**dòng all:**

all: halt shell matmult sort <tên file>

Thêm đoạn sau phía sau matmult:

<tên file>.o: <tên file>.c

\$(CC) \$(CFLAGS) -c <tên file>.c <tên file>:

<tên file>.o start.o

\$(LD) \$(LDFLAGS) start.o <tên file>.o -o <tên file>.coff

../bin/coff2noff <tên file>.coff <tên file>

**Bước 6 :** Biên dịch và chạy Nachos, cd tới ./nachos/code chạy lệnh “make all”.

## 2) Lớp SynchConsole ở ../code/threads/ trong file system.h và system.cc

Chức năng: Hỗ trợ việc nhập xuất từ màn hình console

Gồm 2 hàm chính:

**int Read(char \*into, int numBytes):** Cho phép nhập một chuỗi từ màn hình console và lưu biến thuộc vùng nhớ hệ điều hành NachOS.

**int Write(char \*from, int numBytes):** Cho phép xuất một chuỗi từ biến thuộc vùng nhớ hệ điều hành NachOS ra màn hình console

## 3) System2User và User2System ở ../code/userprog trong file exception.cc

*Phân tích “System2User”:* **int System2User(int virtAddr, int len, char\* buffer)**

**Chức năng:** Hàm thực hiện chuyển một chuỗi được lưu trong hệ điều hành NachOS vào bộ nhớ của chương trình ứng dụng chạy trên NachOS/MIPS

*Phân tích “User2System”:* **char\* User2System(int virtAddr, int limit)**

**Chức năng:** Hàm thực hiện chuyển một chuỗi được lưu trong vùng nhớ của chương trình chạy trên NachOS/MIPS vào vùng nhớ của hệ điều hành NachOS.

## 2.4. CÀI ĐẶT SYSTEM CALL VÀ EXCEPTION

### 1) Cài đặt hàm IncreasePC():

Làm tăng Programming Counter(PC) để nạp lệnh tiếp theo để thực hiện. Ta thực hiện lưu giá trị của PC hiện tại cho PC trước, nạp giá trị kế cho PC hiện tại, nạp giá trị kế tiếp nữa cho PC kế.

### 2) Cài đặt hàm System Call ReadNum:

- Sử dụng lớp synchcons để đọc 1 số nguyên cho người dùng nhập vào
- Nếu giá trị không phải là số nguyên thì sẽ trả về (0)
- Nếu số đó quá lớn vượt quá giá trị -2,147,483,648 đến 2,147,483,647 thì sẽ trả về số 0

### 3) Cài đặt hàm System Call PrintNum:

- Sử dụng lớp SynchConsoleOut để xuất một số nguyên ra màn hình
- Lấy dữ liệu từ thanh ghi R4 tiến hành kiểm tra rồi xuất ra màn hình
- Nếu không có giá trị thì sẽ hiện ra số 0

### 4) Cài đặt hàm System Call ReadChar:

- **ReadChar** system call sẽ sử dụng lớp SynchConsoleIn để đọc một ký tự do người dùng nhập vào. **Nếu mà lớn hơn 1 ký tự thì sẽ không hợp lệ và return 0**
- **Nếu mà không có ký tự nào thì cũng trả về 0**
- Nếu thỏa mãn 1 ký tự thì trả về ký tự đó

### 5) Cài đặt hàm System Call PrintChar:

- **PrintChar** system call sẽ sử dụng lớp SynchConsoleOut để xuất một ký tự ra màn hình
- Lấy dữ liệu từ thanh ghi R4 và xuất ra màn hình ký tự đầu tiên



## 6) Cài đặt hàm System Call RandomNum:

- **RandomNum** system call sẽ trả về một số nguyên dương ngẫu nhiên
- Thông qua hàm random có sẵn trong sysdep.cc ta có thể gọi tới và trả về giá trị random

## 7) Cài đặt hàm System Call ReadString:

- Gọi hàm Read của SynchConsole để đọc chuỗi
- Lấy địa chỉ tham số buffer truyền vào thành ghi thứ 4
- Lấy độ dài tối đa của chuỗi nhập vào thành ghi thứ 5

## 8) Cài đặt hàm System Call PrintString:

- Lấy địa chỉ của tham số buffer từ thanh ghi thứ 4
- Copy chuỗi từ vùng nhớ User Space sang System Space với bộ nhớ đệm buffer dài 255 ký tự
- Đếm độ dài của chuỗi và gọi hàm Write của SynchConsole để in chuỗi

## 9) Chương trình help:

- Gọi hàm PrintString đã viết để in ra thông tin về các thành viên của nhóm và mô tả vắn tắt về chương trình **sort** và **ascii**

```
[sv@localhost code]$ ./userprog/nachos -x ./test/help
Gioi thieu
Do an he dieu hanh so 1
20127278 - Nguyen Van Phu
20127480 - Nguyen Duc Duy
20127493 - Nguyen Van Hau
** Bang ma ASCII gom 256 ki tu, nhưng chi in ra 93 ky tu co the doc duoc
** Sort: Cho phep nguoi dung nhap vao mang so nguyen it hon 100 so nguyen
va sau do su dung thuat toan Bubble Sort de sap xep
Machine halting!

Ticks: total 35308, idle 31900, system 3340, user 68
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 319
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
```

## 10) Chương trình ascii:

- Chạy vòng lặp for để xuất ra các ký tự in được trong bảng mã ascii bằng cách gọi hàm PrintChar

```
[sv@localhost code]$ ./userprog/nachos -x ./test/ascii
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H
I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j k l m n o p
q r s t u v w x y z { | } ~ Machine halting!
```

```
Ticks: total 37166, idle 28200, system 6590, user 2376
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 282
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Cleaning up...

### 11) Chương trình sortBubble:

- Sử dụng hàm ReadNum để cho người dùng nhập vào kích thước array, các phần tử của array và hướng sort rồi sau đó dùng thuật toán bubble sort để sắp xếp các phần tử trong array và dùng vòng lặp for kết hợp với hàm PrintNum để xuất ra mảng số nguyên đó.

```
[sv@localhost code]$ ./userprog/nachos -x ./test/sortBubble
Input the size of array: 4
Input all elements of array
2
3
4
0
1.Increasing order
2.Decreasing order
0.Exit
Your choice: 1
The array after sorted:
0 2 3 4
1.Increasing order
2.Decreasing order
0.Exit
Your choice: 2
The array after sorted:
4 3 2 0
1.Increasing order
2.Decreasing order
0.Exit
Your choice: 0
Machine halting!

Ticks: total 1350138405, idle 1350133014, system 3950, user 1441
Disk I/O: reads 0, writes 0
```

## 2.5. TÀI LIỆU THAM KHẢO

- **GV. Nguyễn Tấn Sơn.** *Bộ tài liệu giáo khoa (2018).*
- **Saman Hadiani, Niklas Dahlbäck, and Uwe Assmann** Linköpings Universitet. *Nachos Beginner's Guid.*  
<https://www.ida.liu.se/~TDDI04/material/begguide/>
- **Căn bản và rất căn bản về hệ điều hành NachOS.**  
[http://read.pudn.com/downloads161/ebook/733633/Nachos\\_CanBan/Nachos\\_CanBan.p  
df](http://read.pudn.com/downloads161/ebook/733633/Nachos_CanBan/Nachos_CanBan.pdf)
- **Dang Khoa-Le Tan.** *Loạt bài nachOS.*  
[http://dangkhoa.com.blogspot.com/p/nachos.html](http://dangkhoa.com/blogspot.com/p/nachos.html)
- **Nguyễn Thành Chung.** *Loạt video hướng dẫn đồ án nachOS.*  
[https://www.youtube.com/watch?v=t0jtY1C129s&list=PLRgTVtca98hUqCN2\\_2vzsAAXPiT  
FbvHpO](https://www.youtube.com/watch?v=t0jtY1C129s&list=PLRgTVtca98hUqCN2_2vzsAAXPiTFbvHpO)