**Report**

# 1Thông tin nhóm

|  |  |  |
| --- | --- | --- |
| **Nhóm 13STT** | **MSSV** | **Họ Tên** |
| 1 | 20120342 | Trần Kỳ Nhật |
| 2 | 20120390 | Võ Hữu Trọng |
| 3 | 20120435 | Lê Thị Ngọc Bích |

# 2Nội dung đồ án

## 2.1. Cài đặt NachOS trên nền tảng linux

Hướng dẫn tại:

<https://www.fit.hcmus.edu.vn/~ntquan/os/setup_nachos.html>

## 2.2. Cài đặt System call để chương trình có thể chạy trên NachOS

* **Tạo các macro:** Để tránh xung đột, nhầm lẫn trong việc gọi hàm cần gán system call với một số cụ thể. Chúng ta có thể thực hiện bước này ở địa chỉ NachOS-4.0/code/userprog/syscall.h

**Quy tắc gán:** **#define < tên system call> <số cụ thể>**

**Ví dụ:** #define SC\_ReadNum 43

* **Định nghĩa hàm trong** **start.S:**

.globl <tên hàm>

.ent <tên hàm>

<tên hàm>:

addiu $2,$0,<tên System call>

syscall

j $31

.end <tên hàm>

**Ví dụ:** hàm **ReadChar**

.globl ReadChar

.ent ReadChar

ReadChar:

addiu $2,$0,SC\_ReadChar

syscall

j $31

.end ReadChar

* Cài đặt exeption.cc ở địa chỉ NachOS-4.0/code/userprog/exception.cc

Sử dụng “switch…..case” để gọi các Syscall theo từng trường hợp.

* Cài đặt Makefile ở địa chỉ NachOS-4.0/code/test/Makefile

Ở PROGRAMS = add halt shell matmult sort segments thêm các file muốn tạo và cấu hình ở dưới theo quy tắc.

**Quy tắc:** PROGRAMS = add halt shell matmult sort segments **<tên file> <tên file 2>**

<tên file>.o: <tên file>..c

$(CC) $(CFLAGS) -c num.c

<tên file>.: <tên file>.o start.o

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

$(COFF2NOFF) <tên file>.coff <tên file>.

**Ví dụ:** PROGRAMS = add halt shell matmult sort segments num character random string ascii sortNumber help

character.o: character.c

$(CC) $(CFLAGS) -c character.c

character: character.o start.o

$(LD) $(LDFLAGS) start.o character.o -o character.coff

$(COFF2NOFF) character.coff character

### Viết lại file exception.cc để xử lý tất cả các exceptions được liệt kê trong *machine/machine.h*.

NoException : Trả lại quyền điều khiển cho hệ điều hành

Các Exception khác: HĐH hiển thị ra một thông báo lỗi và **SysHalt()** hệ thống.

### Viết mã để tăng gia trị biến program counter.

Hàm **move\_program\_counter()** để tăng program counter trước khi system call trả kết quả về.

Sau mỗi **user program** ở các phần sau đề trả về **move\_program\_counter()** để tăng program counter.

### Cài đặt system call int ReadNum(). ReadNum, void PrintNum(int number). PrintNum

Chạy chương trình bằng file num.

Vì giới hạn số nguyên nằm trong khoảng -2^31 đến 2^31-1 nên khi nhập vào hàm sẽ trả về 0.

Các số ngoài số nguyên hoặc kí tự không phải là số nguyên được chương trình kiểm tra và trả về 0

### Cài đặt system call char ReadChar(). ReadChar ,void PrintChar(char character). PrintChar

Chạy chương trình bằng file character.

System call in ra kí tự được nhập đầu tiên, các kí tự sau chương trình sẽ bỏ qua.

### Cài đặt system call int RandomNum(). RandomNum

Chạy chương trình bằng file **random**.

System call sẽ in ra một số nguyên dương ngẫu nhiên.

### Cài đặt đặt system call void ReadString (char[] buffer, int length), void PrintString (char[] buffer),  PrintString

Chạy chương trình bằng file **string**.

Yêu cầu nhập độ dài của chuỗi với độ dài không quá 255 kí tự.

System call sẽ in ra chuỗi được nhập, nếu chuỗi được nhập nhiều hơn độ dài khai báo ở trên system call sẽ in tới độ dài khai báo, các kí tự sau được bỏ qua.

### Viết chương trình **sort**

Chạy chương trình bằng file sortNumber.

Do **sort** đã có sẵn khi tạo Nachos nên nhóm sử dụng **sortNumber** sử dụng phương pháp bubblesort thay thế.

Nhập số nguyên n với 0<=n<=100.

Kiểm tra n có phải số nguyên và 0<=n<=100 nếu không thỏa điều kiện thì yêu cầu nhập lại.

Nhập mảng số nguyên sau đó chọn kiểu sort 1 hoặc 2

1 . Tăng dần

2. Giảm dần

Nếu không thỏa điều kiện yêu cầu nhập lại.

System call cuối cùng in ra dãy đã được sắp xếp.

### Viết chương trình **ascii** để in ra bảng mã ascii

Sử dụng hàm **printChar()** để in ra các ký tự ascii có thể đọc được và giá trị của nó