

# HỆ ĐIỀU HÀNH

## ĐỒ ÁN 1: LẬP TRÌNH NACHOS

**Giảng viên:** Trần Trung Dũng

**Hướng dẫn:** Lê Giang Thanh

**Lớp:** 19TN

Người thực hiện:

| Họ và tên     | MSSV     |
|---------------|----------|
| Mai Duy Nam   | 19120298 |
| Đặng Thái Duy | 19120491 |
| Võ Văn Toàn   | 19120690 |

## MỤC LỤC

|           |  |           |
|-----------|--|-----------|
| <b>1.</b> | <b>CÀI ĐẶT TỔNG QUAN</b>   | <b>3</b>  |
| 1.1       | Cấu hình   | 3         |
| 1.2       | Cấu trúc tổng quát của đồ án   | 3         |
| 1.3       | Các bước cài đặt và chạy thử nghiệm  | 3         |
| 1.4       | Các bước cài đặt một system call   | 4         |
| <b>2.</b> | <b>CÀI ĐẶT SYSTEM CALL VÀ EXCEPTION</b>                                    | <b>6</b>  |
| 2.1       | Cài đặt lại các exception  | 6         |
| 2.2       | Cài đặt hàm IncreasePC()   | 6         |
| 2.3       | Cài đặt system call ReadNum: int ReadNum()                                 | 6         |
| 2.4       | Cài đặt system call PrintNum: void PrintNum(int number)                    | 7         |
| 2.5       | Cài đặt system call ReadChar: char ReadChar()                              | 7         |
| 2.6       | Cài đặt system call PrintChar: void PrintChar(char character)              | 7         |
| 2.7       | Cài đặt system call RandomNum(): int RandomNum()                           | 8         |
| 2.8       | Cài đặt system call ReadString: void ReadString(char[] buffer, int length) | 8         |
| 2.9       | Cài đặt system call PrintString: void PrintString(char[] buffer)           | 8         |
| 2.10      | Cài đặt chương trình help  | 8         |
| 2.11      | Cài đặt chương trình ascii   | 8         |
| 2.12      | Cài đặt chương trình sort  | 9         |
| <b>3.</b> | <b>DEMO CHƯƠNG TRÌNH</b>   | <b>10</b> |
| 3.1       | Halt   | 10        |
| 3.2       | ReadNum  | 10        |
| 3.3       | PrintNum   | 11        |
| 3.4       | RandomInt  | 11        |
| 3.5       | ReadChar   | 11        |
| 3.6       | PrintChar  | 11        |
| 3.7       | ReadString   | 12        |
| 3.8       | PrintString  | 12        |
| 3.9       | Help   | 12        |
| 3.10      | Sort   | 13        |

# 1. CÀI ĐẶT TỔNG QUAN

## 1.1 Cấu hình

- Phiên bản NachOS: 4.0
- Hệ điều hành: Ubuntu

## 1.2 Cấu trúc tổng quát của đồ án

Toàn bộ mã nguồn của đồ án nằm trong thư mục `nachos/`. Bên trong đó, thư mục `NachOS-4.0/` chứa mã nguồn của NachOS, thư mục `usr/` chứa cross-compiler cho kiến trúc MIPS. Thư mục `code/` bên trong NachOS-4.0 là nơi đồ án tập trung xây dựng và chỉnh sửa chủ yếu. Cụ thể, đồ án tập trung chỉnh sửa các file và thư mục sau:

- `userprog/syscall.h`: định nghĩa macro và prototype hàm của các system call
- `userprog/exception.cc`: cài đặt các exception handler, bao gồm `SyscallException` - exception được raise bất kỳ khi nào chương trình người dùng gọi một system call, và các exception khác.
- `test/`: chứa các đoạn code kiểm tra cài đặt system call

## 1.3 Các bước cài đặt và chạy thử nghiệm

Bước 1: Tiến hành build NachOS

```
$ cd nachos/NachOS-4.0/code/build.linux
$ make depend
$ make
```

Bước 2: Tiến hành build các chương trình test

```
$ cd ../test
$ make
```

Bước 3: Chạy các chương trình test

```
$ ../../build.linux/nachos -rs <integer> -x <program>
```

## 1.4 Các bước cài đặt một system call

Sau đây là các bước cài đặt một system call mẫu - Add - dùng để cộng hai số nguyên.

Bước 1: Định nghĩa macro cho system call cần cài đặt trong `userprog/syscall.h`. Ví dụ:

```
#define SC_ADD 21
```

Bước 2: Khai báo interface cho system call trong `userprog/syscall.h`. Ví dụ:

```
int Add(int a, int b);
```

Bước 3: Trong cấu trúc `switch-case` định nghĩa trong `userprog/exception.cc`, tìm đến `case SyscallException`. Trong `case` này cài đặt một cấu trúc `switch-case` khác để xử lý các system call dựa trên macro đã định sẵn. Thêm một `case` mới cho `SC_ADD`:

```
case SC_ADD:
```

Bước 4: Cài đặt các thao tác xử lý system call này, có thể sử dụng các dịch vụ của kernel có sẵn.

Bước 5: Viết chương trình kiểm thử system call vừa cài đặt trong `test/`. Ví dụ: `test/add.c`:

```
#include "syscall.h"

int main() {
    int a = 1;
    int b = 2;
    int c = Add(a, b);
    Halt();
}
```

Bước 6: Thêm `add` vào biến `PROGRAM` và target `all` trong `test/Makefile`

```
PROGRAM = [...] add
all: [...] add
```

Bước 7: Thêm target để biên dịch `add.c` như sau:

```
add.o: add.c
    $(CC) $(CFLAGS) -c add.c
add: add.o start.o
    $(LD) $(LDFLAGS) start.o add.o -o add.coff
    $(COFF2NOFF) add.coff add
```

Bước 8: Thêm đoạn lệnh sau vào file `test/start.S`

```
.globl Add
.ent Add
Add:
    addiu $2,$0,SC_Add
    syscall
    j $31
.end Add
```

Bước 9: Build NachOS để thêm system call mới định nghĩa vào hệ thống, build chương trình kiểm thử và thực thi như mục 1.2.

## 2. CÀI ĐẶT SYSTEM CALL VÀ EXCEPTION

### 2.1 Cài đặt lại các exception

Vào file `machine/machine.h` lấy được danh sách các exception, sau đó qua file `userprog/exception.cc` viết lại từng case cho từng `ExceptionType`. Trước mắt, với mỗi exception ta cài đặt thêm lệnh `kernel->interrupt->Halt()` để tắt hệ điều hành.

### 2.2 Cài đặt hàm `IncreasePC()`

Tất cả các system calls (không phải `Halt`) sẽ yêu cầu NachOS tăng program counter trước khi system call trả kết quả về. Nếu không lập trình đúng phần này thì NachOS sẽ bị vòng lặp gọi thực hiện system call này mãi mãi. Cũng như các hệ thống khác, MIPS xử lý dựa trên giá trị của program counter, vậy nên ta cần xây dựng hàm làm tăng program counter để nạp lệnh tiếp theo để thực hiện.

Ta tiến hành cài đặt hàm `IncreasePC()` như sau: thực hiện lưu giá trị của PC hiện tại cho PC trước → nạp giá trị kế tiếp cho PC hiện tại → nạp giá trị kế tiếp nữa cho PC kế. Khi xây dựng xong hàm, ta sẽ đặt `IncreasePC()` dưới mỗi exception được thực hiện, khi đó ta sẽ không cần dùng lệnh `Halt()` cho các case đó nữa.

### 2.3 Cài đặt system call `ReadNum: int ReadNum()`

- `ReadNum` system call sẽ sử dụng lớp `SynchConsoleIn` để đọc một số nguyên do người dùng nhập vào. Nếu giá trị người dùng nhập không phải là số nguyên thì trả về 0.
- Mô tả cài đặt `SC_ReadNum`:
  - Input: Không có
  - Output: Trả về 1 số nguyên Integer
  - Chức năng: Đọc 1 số nguyên từ màn hình console
- Cài đặt: Ta cấp phát một mảng `char` là `num_buffer` dùng để lưu từng ký tự được đọc từ console. Output của hàm là giá trị `res`. Ta tiến hành duyệt từng ký tự đọc được từ console nhờ vào hàm `GetChar()` của lớp `SynchConsoleIn`, nếu hợp lệ là số thì lưu tuần tự vào `num_buffer`, còn nếu không thì ta in ra lỗi “invalid integer number” sau đó lưu kết quả là 0 vào thanh ghi `r2` và đồng thời kết thúc hàm. Lúc này, nếu `num_buffer` đều toàn các ký tự là số thì tiến hành tính toán giá trị `res` từ `num_buffer` đó. Kiểm tra xem `res` có nằm ngoài phạm vi của Integer không, nếu có thì ta thông báo lỗi “out of range”, lưu lại kết quả 0 vào thanh ghi `r2` và kết thúc hàm. Nếu không, thì kiểm tra xem `res` là số dương hay số âm dựa theo biến

`isPositive` để cập nhật lại giá trị chính xác cho `res`, sau đó lưu kết quả `res` vào thanh ghi `r2` và kết thúc.

## 2.4 Cài đặt system call `PrintNum: void PrintNum(int number)`

- `PrintNum` system call sẽ sử dụng lớp `SynchConsoleOut` để xuất một số nguyên ra màn hình.
- Mô tả cài đặt `SC_PrintNum`:
  - Input: 1 số nguyên Integer
  - Output: Không có
  - Chức năng: In số nguyên ra màn hình console
- Cài đặt: Ta đọc tham số `number` từ thanh ghi `r4`. Cấp phát 1 mảng `char num_buffer` với mục đích để lưu từng ký tự số trong `number`. Kiểm tra xem số đó là số dương hay số âm để cập nhật cho biến `isPositive`. Sau đó ta lưu từng chữ số trong `number` vào trong 1 mảng `temp` phụ, và từ đó cập nhật lại từng ký tự trong `num_buffer` theo mảng `temp` đó. Lúc này `num_buffer` đã lưu lần lượt các ký tự số của `number` (nếu là số âm thì ký tự bắt đầu là "-"), ta tiến hành in lần lượt chúng lên console nhờ vào hàm `PutChar()` của lớp `SynchConsoleOut`.

## 2.5 Cài đặt system call `ReadChar: char ReadChar()`

- `ReadChar` systemcall call sẽ sử dụng lớp `SynchConsoleIn` cùng hàm `SysReadString` (file `userprog/ksyscall.h`) để đọc một chuỗi ký tự do người dùng nhập vào rồi trả về giá trị của ký tự đầu tiên trong chuỗi.
- Mô tả cài đặt `SC_ReadChar`:
  - Input: 1 ký tự/chuỗi
  - Output: Trả về 1 ký tự
  - Chức năng: Tạo ra và trả về ký tự (đầu chuỗi) được nhập vào
- Cài đặt: Dùng hàm `SysReadString` để đọc một chuỗi do người dùng nhập vào biến `s`. Sau đó gán `s[0]` vào biến `c`, xóa `s`, lưu `c` vào thanh ghi `r2` và kết thúc.

## 2.6 Cài đặt system call `PrintChar: void PrintChar(char character)`

- `PrintChar` system call sẽ sử dụng lớp `SynchConsoleOut` để xuất một ký tự ra màn hình.
- Mô tả cài đặt `SC_PrintChar`:
  - Input: 1 ký tự
  - Output: Không có
  - Chức năng: In ký tự đó ra màn hình console
- Cài đặt: Ta đọc tham số `c` từ thanh ghi `r4`, ta tiến hành in `c` lên console nhờ vào hàm `PutChar()` của lớp `SynchConsoleOut`.

## 2.7 Cài đặt system call RandomNum(): int RandomNum()

- RandomNum system call sẽ trả về một số nguyên dương ngẫu nhiên.
- Mô tả cài đặt SC\_RandomNum:
  - o Input: Không có
  - o Output: Trả về một số nguyên dương
  - o Chức năng: Tạo ra và trả về một số nguyên dương ngẫu nhiên
- Cài đặt: Ta dùng hàm rand() có sẵn của C để khởi tạo ra 1 số nguyên dương ngẫu nhiên là number. Sau đó lưu kết quả number vào thanh ghi r2 và kết thúc.

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

- ReadString system call sử dụng lớp SynchConsoleIn để đọc một chuỗi ký tự do người dùng nhập vào.
- Mô tả cài đặt SC\_ReadString:
  - o Input: địa chỉ của buffer, số ký tự tối đa
  - o Output: một chuỗi
  - o Chức năng: Đọc một chuỗi do người dùng nhập vào
- Cài đặt: Ta đọc địa chỉ buffer từ thanh ghi r4 và độ dài chuỗi từ thanh ghi r5, sau đó sử dụng SynchConsoleIn để đọc từ console vào một mảng ở kernel space, sau đó chép mảng đó từ kernel space vào buffer bằng System2User.

## 2.9 Cài đặt system call PrintString: void PrintString(char[] buffer)

- ReadString system call sẽ sử dụng lớp SynchConsoleOut để in một chuỗi ký tự do người dùng nhập vào.
- Mô tả cài đặt SC\_PrintString:
  - o Input: 1 chuỗi ký tự, kết thúc bằng \0
  - o Output: Không có
  - o Chức năng: in ra console một chuỗi ký tự
- Cài đặt: Ta đọc địa chỉ chuỗi buffer trong user space từ thanh ghi r4, sao chép buffer từ userspace vào kernel space, tìm độ dài của chuỗi, sau đó sử dụng SynchConsoleOut để in chuỗi ra console.

## 2.10 Cài đặt chương trình help

Chương trình help sử dụng system call PrintString để in ra các chuỗi thể hiện các thông tin về nhóm, cùng với thông tin hai chương trình ascii và sort ra console.

## 2.11 Cài đặt chương trình ascii



Chương trình `ascii` sử dụng system call `PrintNum` để in mã ASCII, và sử dụng system call `PrintChar` để in các ký tự.

## 2.12 Cài đặt chương trình sort

Chương trình `sort`:

- Sử dụng system call `ReadNum` để nhận vào kích thước mảng và các phần tử của mảng do người dùng nhập.
- Sử dụng system call `ReadString` để nhận chiều sắp xếp (tăng dần hoặc giảm dần).
- Sau khi sắp xếp bằng bubble sort, sử dụng system call `PrintChar`, `PrintNum` và `PrintString` để in mảng ra console.

### 3. DEMO CHƯƠNG TRÌNH

Video:

[https://drive.google.com/file/d/1YVYfBrVFk2TaI64sgE\\_duRLI1J8fYUoY/view?usp=sharing](https://drive.google.com/file/d/1YVYfBrVFk2TaI64sgE_duRLI1J8fYUoY/view?usp=sharing)

#### 3.1 Halt

```
mml@mml-VirtualBox:~/hcmus-os-nachos/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -rs 1023 -x halt

Shutdown, initiated by user program. Machine halting!

Ticks: total 62, idle 0, system 50, user 12
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
```

#### 3.2 ReadNum

```
mml@mml-VirtualBox:~/hcmus-os-nachos/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -rs 1023 -x readnum -123

Shutdown, initiated by user program. Machine halting!

Ticks: total 388162688, idle 388162438, system 220, user 30
Disk I/O: reads 0, writes 0
Console I/O: reads 5, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0
```

```
mml@mml-VirtualBox:~/hcmus-os-nachos/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -rs 1023 -x readnum 1234567890123

Read: 0The integer number is out of range
Shutdown, initiated by user program. Machine halting!

Ticks: total 194838342, idle 194837528, system 770, user 44
Disk I/O: reads 0, writes 0
Console I/O: reads 14, writes 7
Paging: faults 0
Network I/O: packets received 0, sent 0
```

```
mml@mml-VirtualBox:~/hcmus-os-nachos/nachos/NachOS-4.0/code/test$
../build.linux/nachos -rs 1023 -x readnum
abc
I

Read: 0The integer number is not valid
Shutdown, initiated by user program. Machine halting!

Ticks: total 586184612, idle 586184158, system 410, user 44
Disk I/O: reads 0, writes 0
Console I/O: reads 4, writes 7
Paging: faults 0
Network I/O: packets received 0, sent 0
```

### 3.3 PrintNum

```
mml@mml-VirtualBox:~/hcmus-os-nachos/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -rs 1023 -x printnum
-12345
Shutdown, initiated by user program. Machine halting!

Ticks: total 912, idle 600, system 280, user 32
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 6
Paging: faults 0
Network I/O: packets received 0, sent 0
```

### 3.4 RandomInt

```
mml@mml-VirtualBox:~/hcmus-os-nachos/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -rs 1023 -x randomnum
1553723043
Shutdown, initiated by user program. Machine halting!

Ticks: total 1417, idle 1000, system 380, user 37
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 10
Paging: faults 0
Network I/O: packets received 0, sent 0
```

### 3.5 ReadChar

```
mml@mml-VirtualBox:~/hcmus-os-nachos/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -rs 1023 -x readchar
abc
Read: a
Shutdown, initiated by user program. Machine halting!

Ticks: total 125523933, idle 125523468, system 420, user 45
Disk I/O: reads 0, writes 0
Console I/O: reads 4, writes 7
Paging: faults 0
Network I/O: packets received 0, sent 0
```

### 3.6 PrintChar

```
mml@mml-VirtualBox:~/hcmus-os-nachos/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -rs 1023 -x printchar
a
Shutdown, initiated by user program. Machine halting!

Ticks: total 219, idle 100, system 90, user 29
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 1
Paging: faults 0
Network I/O: packets received 0, sent 0
```

### 3.7 ReadString

```
mml@mml-VirtualBox:~/hcmus-os-nachos/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -rs 1023 -x readstring
NachOS
Read: NachOS

Shutdown, initiated by user program. Machine halting!

Ticks: total 487140040, idle 487139307, system 690, user 43
Disk I/O: reads 0, writes 0
Console I/O: reads 7, writes 13
Paging: faults 0
Network I/O: packets received 0, sent 0
```

### 3.8 PrintString

```
mml@mml-VirtualBox:~/hcmus-os-nachos/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -rs 1023 -x printstring
Hello, world!

Shutdown, initiated by user program. Machine halting!

Ticks: total 1950, idle 1400, system 510, user 40
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 14
Paging: faults 0
Network I/O: packets received 0, sent 0
```

### 3.9 Help

```
mml@mml-VirtualBox:~/hcmus-os-nachos/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -rs 1023 -x help
Group members
ID      | Name
----- | -----
19120298 | Mai Duy Nam
19120491 | Dang Thai Duy
19120690 | Vo Van Toan

Program descriptions
ascii: Print a list of printable ASCII characters
usage: ../build.linux/nachos -x ascii

sort: Sort a list of integers using bubble sort algorithm
usage: ../build.linux/nachos -x sort

Shutdown, initiated by user program. Machine halting!

Ticks: total 45631, idle 33900, system 11300, user 431
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 339
Paging: faults 0
Network I/O: packets received 0, sent 0
```

### 3.10 Sort

```
mml@mml-VirtualBox:~/hcmus-os-nachos/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -rs 1023 -x sort
Enter array size: 5
Sort by (ascending: <, descending: >): <
Enter element: 4
Enter element: 2
Enter element: 1
Enter element: 5
Enter element: 3
You've entered: 4 2 1 5 3
Sorted: 1 2 3 4 5

Shutdown, initiated by user program. Machine halting!

Ticks: total 1098333208, idle 1098325106, system 6620, user 1482
Disk I/O: reads 0, writes 0
Console I/O: reads 14, writes 178
Paging: faults 0
Network I/O: packets received 0, sent 0
```

```
mml@mml-VirtualBox:~/hcmus-os-nachos/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -rs 1023 -x sort
Enter array size: 5
Sort by (ascending: <, descending: >): >
Enter element: 4
Enter element: 2
Enter element: 1
Enter element: 5
Enter element: 3
You've entered: 4 2 1 5 3
Sorted: 5 4 3 2 1

Shutdown, initiated by user program. Machine halting!

Ticks: total 271198952, idle 271190926, system 6540, user 1486
Disk I/O: reads 0, writes 0
Console I/O: reads 14, writes 178
Paging: faults 0
Network I/O: packets received 0, sent 0
```