HỆ ĐIỀU HÀNH

Project 1:

Exceptions và các system calls đơn giản

Danh sách các thành viên:

1. Huỳnh Ngô Trung Trực 19120040
2. Nguyễn Lê Bảo Thi 19120376
3. Lê Trung Hiếu 19120507

fit@hcmus

MỤC LỤC

1 Bảng đánh giá thành viên

MSSV	Họ Tên	Các công việc thực hiện
19120040	Huỳnh Ngô Trung Trực	Câu d, i, j, demo, viết báo cáo
19120376	Nguyễn Lê Bảo Thi	Câu a, b, c, k, viết báo cáo
19120507	Lê Trung Hiếu	Câu d, e, f, g, h, viết báo cáo

2 Thiết kế và cài đặt chung

2.1 Cài đặt Nachos

Bước 1: Cài đặt hệ điều hành Ubuntu 18.04

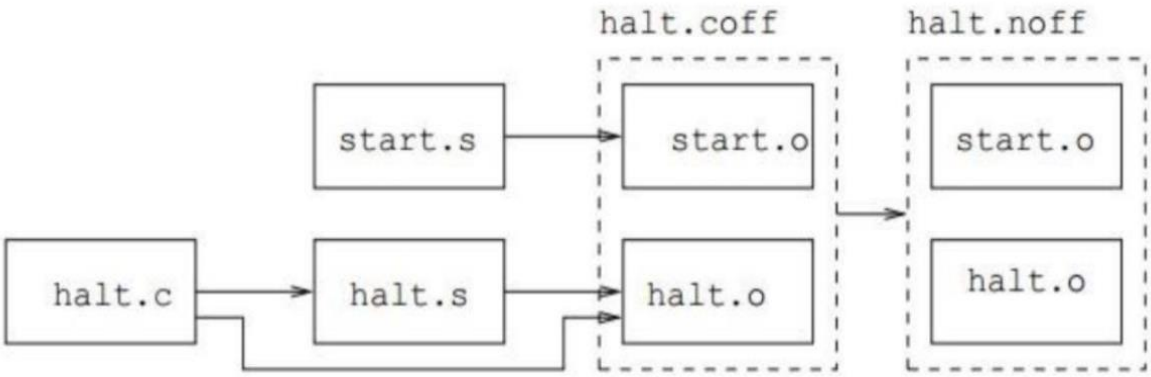
Bước 2: Cài đặt NachOS 4.0 theo hướng dẫn trong link ở moodle

2.3 Cơ chế thực thi và các bước cài đặt Chương trình

2.3.1 Cơ chế thực thi Chương trình

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:

- Chương trình halt.c được cross-compiler biên dịch thành tập tin halt.s là mã hợp ngữ chạy trên kiến trúc MIPS.
- Tập tin halt.s được liên kết với start.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.
- Tập tin halt.coff được tiện ích coff2noff được chuyển thành tập tin halt.noff dạng file thực thi trên NachOS kiến trúc MIPS.



2.3.2 Các bước cài đặt Chương trình

Bước 1: Viết chương trình người dùng .c trong thư mục ./code/test

Bước 2: Vào ./code/test/Makefile thêm tên chương trình vào sau dòng dưới comment "# change this if you create a new test program!"

PROGRAMS = add halt shell matmult sort segments <tên chương trình>

Bước 3: Cũng tại Makefile này, thêm đoạn này phía trước đoạn clean:

<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 4: Biên dịch và chạy chương trình NachOS bằng cách tới thư mục ./nachos/NachOS-4.0/code/ mở Terminal và chạy lệnh make all.

2.2 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 Cách cài đặt System Calls

Bước 1: Định nghĩa một mã số cho mỗi system call. Vào thư mục ./nachos/NachOS-4.0/code/userprog/syscall.h thêm dòng:

#define <tên syscall> <mã số>

Bước 2: Cũng tại file syscall.h, thêm phần khai báo hàm cho system call muốn tạo (nếu chưa được khai báo)

Bước 3: Mở file ./code/test/start.c và ./code/test/start.s thêm đoạn mã:

.globl <tên hàm>

.ent <tên hàm>

<tên hàm>:

addiu $2,$0,<tên system call>

syscall

j $31

.end <tên hàm>

Bước 4: Viết chương trình xử lý các case cho các system call trong ./code/userprog/exception.cc, viết cài đặt hàm ở ksyscall.h

Bước 5: Viết chương trình người dùng .c để kiểm tra các system call trong thư mục ./code/test (Sử dụng hàm như đã khai báo trong prototype ở ./code/userprog/syscall.h

Bước 6: Vào ./code/test/Makefile thêm tên chương trình vào sau dòng dưới comment "# change this if you create a new test program!"

PROGRAMS = add halt shell matmult sort segments <tên chương trình>

Bước 7: Cũng tại Makefile này, thêm đoạn này phía trước đoạn clean:

<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 8: Biên dịch và chạy chương trình NachOS bằng cách tới thư mục ./nachos/NachOS-4.0/code/ mở Terminal và chạy lệnh make all.

3 Thiết kế và cài đặt Exceptions và System Calls

3.1 Viết lại file exceptions.cc (Câu a)

File exception.cc để xử ly' tất cả các exceptions được liệt kê trong *machine/machine.h*. Hầu hết các exception trong này là run-time errors, khi các exception này xảy ra thì user program không thể được phục hồi. Trường hợp đặc biệt duy nhất là *no exception* sẽ trả quyền điều khiển về HĐH ta sẽ sử dụng lệnh return, còn *syscall exceptions* sẽ được xử ly' bởi các hàm chúng ta viết cho user system calls. Với tất cả exceptions khác, HĐH hiển thị ra một thông báo lỗi bằng lệnh DEBUG (<loại lỗi>, < thông tin lỗi>) và dùng cerr để in ra màn hình thông tin lỗi và Halt hệ thống bằng SysHalt() (SysHalt() thực ra là gọi kernel->interrupt->halt()). Ta sẽ cấu trúc các loại lỗi trong exceptions cũng như trong syscall exceptions bằng switch...case.

3.2 Cài đặt hàm IncreasePC() (Câu c)

Tất cả các system calls (ko 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 ly' dựa trên giá trị của program counter, vì vậy ta phải viết mã để tăng gia trị biến program counter. Hàm IncreasePC() sẽ lưu giá trị của thanh ghi hiện tại vào thanh ghi trước, lưu giá trị của thanh ghi kế tiếp vào thanh ghi hiện tại và tăng giá trị của thanh ghi kế tiếp lên 4.

3.3 Cài đặt system call int ReadNum()

Gọi hàm kernel->synchConsoleIn->GetChar() sau đó kiểm tra ký tự không hợp lệ và trả về 0 nếu phát hiện. Kiểm tra dấu của ký tự đầu tiên và chuyển các ký tự thành số tương ứng để tính toán kết quả trả về, nếu kết quả quá lớn hoặc quá nhỏ, trả về giá trị lớn nhất hoặc nhỏ nhất trong kiểu dữ liệu integer.

Tại exception.cc, sau khi nhận giá trị trả về từ hàm SysReadChar(), ta dùng lệnh kernel->machine->WriteRegister(2, (int)result); để lưu kết quả vừa được trả về vào thanh ghi số 2. Sau đó gọi hàm IncreasePC() để tăng giá trị cho program counter, tránh bị loop mãi mãi.

3.4 Cài đặt system call void PrintNum(int number)

Tại exception.cc dùng câu lệnh kernel->machine->ReadRegister(4) để đọc dữ liệu từ thanh ghi, như đã quy ước thì đọc tham số thứ nhất tại thanh ghi số 4, sau đó ép kiểu về int và truyền vào hàm SysPrintNum, sau đó gọi hàm IncreasePC() để tăng giá trị cho program counter, tránh bị loop mãi mãi.

Tại kyscall.h, từ đầu vào int number, ta xử lý int thành 1 mảng các char, sau đó dùng system call kernel->synchConsoleOut->PutChar() để in ra từng char để hiện thị số đó, nếu là số âm thì in trước 1 dấu '-'

3.5 Cài đặt system call char ReadChar()

Ở exception.cc, sau khi nhận giá trị trả về từ hàm SysReadChar(), ta dùng lệnh kernel->machine->WriteRegister(2, (char)result); để lưu kết quả vừa được trả về vào thanh ghi số 2, như đã quy ước, là thanh ghi của giá trị trả về. Sau đó gọi hàm IncreasePC() với cùng lý do đã nêu ở trên.

3.6 Cài đặt system call void PrintChar(char character)

Đơn giản chỉ là gọi hàm kernel->synchConsoleOut->PutChar() để in 1 ký tự character nhận vào ra màn hình. .

Trước đó ở exception.cc, cần dùng câu lệnh kernel->machine->ReadRegister(4) để đọc tham số thứ nhất được truyền vào hàm, sau đó dùng nó để gọi hàm SysPrintChar để thực sự in nó ra màn hình.

3.7 Cài đặt system call int RandomNum()

Dùng thư viện stdlib.h và time.h để tạo số random bằng câu lệnh srand(time(NULL)), vì là random một số nguyên dương nên ta sẽ mod kết quả vừa nhận được với 2147483647 và cộng thêm 1 để số random ra có giá trị từ [1-2147483647].

Ở exception.cc, sau khi nhận giá trị trả về từ hàm SysRandomNumber(), ta dùng lệnh kernel->machine->WriteRegister(2, (int)result); để lưu kết quả vừa được trả về vào thanh ghi số 2, như đã quy ước, là thanh ghi của giá trị trả về. Sau đó gọi hàm IncreasePC() với cùng lý do đã nêu ở trên.

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

Bổ sung hàm System2User(int virtAddr, int len, char* buffer) để chuyển 1 buffer từ user space sang system space.

Tạo một buffer để lưu các ký tự nhập từ bàn phím và sử dụng kernel->machine-> ReadRegistry(4) để lấy địa chỉ để lưu buffer. Sau đó truyền vào hàm System2User() để tiến hành sao lưu.

Tại exception.cc, đơn giản chỉ gọi hàm SysReadString() để đọc chuỗi từ bàn phím và hàm IncreasePC() để tăng giá trị cho program counter, tránh bị loop mãi mãi.

3.9 Cài đặt system call void PrintString (char buffer[])

Bổ sung hàm User2System(int virtAddr, int limit) để chuyển 1 buffer từ system space sang user space.

Sử dụng kernel->machine->ReadRegistry(4) để lấy địa chỉ buffet và truyền vào hàm System2User(). Sau khi nhận

Tại exception.cc, đơn giản chỉ gọi hàm SysPrintString() để đọc chuỗi từ bàn phím và hàm IncreasePC() để tăng giá trị cho program counter, tránh bị loop mãi mãi.

3.10 Viết chương trình help

Mục đích: Dùng để in ra các dòng giới thiệu cơ bản về nhóm và mô tả vắn tắt về chương trình sort và ascii.

Dùng system call PrintString(char[]) để in nội dung cần thiết ra màn hình console.

3.11 Viết chương trình ascii

Mục đích: Dùng để in ra bảng mã ascii (bắt buộc các kí tự đọc được, các mã ký tự không đọc được không cần phải in ra)

Dùng system call PrintNum(int) để in ra các giá trị DEC (cơ số 10) ra màn hình và system call PrintChar(char) để in ra ký tự tương ứng với mã ascii. Riêng với kí tự ứng với mã số là 32 là khoảng trắng sẽ được mô tả bằng [SPACE] và kí tự ứng với mã số là 127 sẽ được mô tả là [DEL]. Ngoài ra còn dùng system call PrintString(char[]) để hiển thị thêm các chuỗi để trình bày bảng mã.

3.12 Viết chương trình sort

Mục đích: Sử dụng thuật toán bubble sort để sắp xếp mảng theo chiều tăng dần hoặc giảm dần tùy vào người dung lựa chọn (Cho phép người dùng nhập vào một mảng n số nguyên với n là số do người dùng nhập vào $n \leq 100$)

Sử dụng system call PrintString(char[]) để in ra các yêu cầu cũng như thông báo cho người dùng như nhập số lượng phần tử, nhập các phần tử, nhập kiểu sắp xếp cũng như thông báo lỗi về việc nhập ngoài phạm vi số lượng phần tử, nhập sai kiểu sắp xếp,...(Mỗi lần lỗi không đúng yêu cầu sẽ dùng system call Halt() để thoát ra khỏi chương trình). Sử dụng system call ReadNum() để nhận dữ liệu số được nhập từ bàn phím của người dùng. Dùng system call PrintNum(int) và PrintChar(char) để in kết quả ra màn hình.

4 Demo

4.1 RandomNum, ReadNum và PrintNum

```
trucj@DESKTOP-II7RAF9:~/nachos/NachOS-4.0/code/build.linux$ ./nachos -x ../test/number
So nguyen ngau nhien la:
1304347313
Nhap 1 so nguyen:
2021
So nguyen vua nhap la:
2021
```

```
trucj@DESKTOP-II7RAF9:~/nachos/NachOS-4.0/code/build.linux$ ./nachos -x ../test/number
So nguyen ngau nhien la:
1149437846
Nhap 1 so nguyen:
12.34
So nguyen vua nhap la:
0
```

```
trucj@DESKTOP-II7RAF9:~/nachos/NachOS-4.0/code/build.linux$ ./nachos -x ../test/number
So nguyen ngau nhien la:
1096735288
Nhap 1 so nguyen:
12345678910
So nguyen vua nhap la:
2147483647
```

```
trucj@DESKTOP-II7RAF9:~/nachos/NachOS-4.0/code/build.linux$ ./nachos -x ../test/number
So nguyen ngau nhien la:
2081359140
Nhap 1 so nguyen:
-10987654321
So nguyen vua nhap la:
-2147483648
```

4.2 ReadChar và PrintChar

```
trucj@DESKTOP-II7RAF9:~/nachos/NachOS-4.0/code/build.linux$ ./nachos -x ../test/char
Nhap 1 ky tu:
z
Ky tu vua nhap la:
z
```

```
trucj@DESKTOP-II7RAF9:~/nachos/NachOS-4.0/code/build.linux$ ./nachos -x ../test/char
Nhap 1 ky tu:
hcmus
Ky tu vua nhap la:
h
```

4.3 ReadString và PrintString

```
trucj@DESKTOP-II7RAF9:~/nachos/NachOS-4.0/code/build.linux$ ./nachos -x ../test/string
Nhap 1 chuoi ky tu:
Toi yeu khoa hoc
Chuoi ky tu vua nhap la:
Toi yeu khoa hoc
```

4.4 Chương trình help

```
trucj@DESKTOP-II7RAF9:~/nachos/NachOS-4.0/code/build.linux$ ./nachos -x ../test/help
---------- Mo ta co ban ve nhom ----------
1. Huynh Ngo Trung Truc - 19120040
2. Nguyen Le Bao Thi - 19120376
3. Le Trung Hieu - 19120507

---------- Mo ta van tat ve chuong trinh sort ----------
Tu nhung thong tin nguoi dung cung cap gom so luong, danh sach phan tu va loai sap xep
Chuong trinh se thuc hien sap xep mang bang Bubble Sort va in ket qua ra man hinh
---------- Mo ta van tat ve chuong trinh ascii ----------
In ra cac ki tu co the nhin thay (Tu 32 den 127) cua bang ma ASCII
```

4.5 Chương trình ascii

```
trucj@DESKTOP-II7RAF9:~/nachos/NachOS-4.0/code/build.linux$ ./nachos -x ../test/ascii
Dec     |     Char
--------|----------
32      |    [SPACE]
33      |       !
34      |       "
35      |       #
36      |       $
37      |       %
38      |       &
39      |       '
40      |       (
41      |       )
42      |       *
43      |       +
44      |       ,
45      |       -
46      |       .
47      |       /
48      |       0
49      |       1
50      |       2
```

4.6 Chương trình sort

```
trucj@DESKTOP-II7RAF9:~/nachos/NachOS-4.0/code/build.linux$ ./nachos -x ../test/sort
Moi ban nhap so luong phan tu n (1 <= n <= 100):
6
Nhap cac phan tu cua mang:
19120040 19120507 19120376 14 10 2021
Chon cach sap xep mang tang dan hay giam dan
  Tang dan: 0
  Giam dan: 1
0
Dang sap xep...
Mang sau khi sap xep la:
10 14 2021 19120040 19120376 19120507
```

5 Tài liệu tham khảo

- Tài liệu hướng dẫn đồ án 1, Install Nachos on Ubuntu 18.04 trên Moodle.
- Loạt video hướng dẫn đồ án trên youtube: Lập trình Nachos HCMUS