Report

NACHOS

EXCEPTIONS VÀ CÁC SYSTEM CALLS ĐƠN GIẢN

Nhóm 13

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

MỤC LỤC

1 Thông tin nhóm

STT	MSSV	Họ Tên
1	1612291	Nguyễn Thị Ngân Khánh
2	1612297	Võ Đăng Khoa
3	1612296	Tạ Ngọc Duy Khoa
4	1612529	Đặng Minh Quân

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 "excutable" 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…

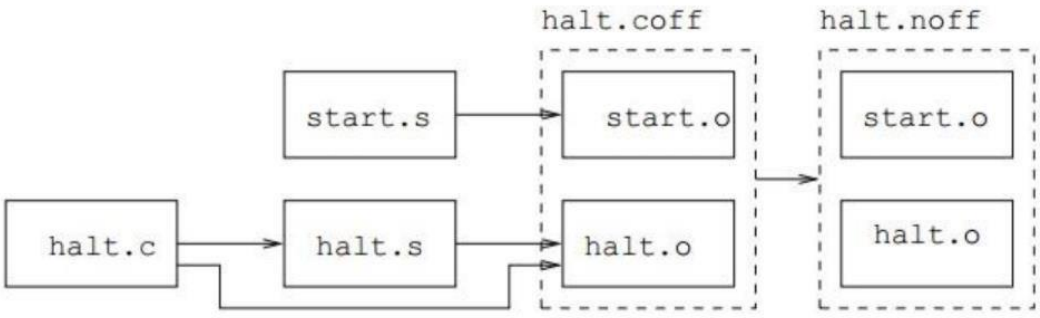
2.2 CÀI ĐẶT TỔNG QUAN

1) Cơ chế thực hiện, quy trình 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**

3) 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:

	VD: System Call **"ReadInt"**
.globl <tên hàm>	.globl ReadInt
.ent <tên hàm>	.ent ReadInt
<tên hàm>:	ReadInt:
addiu $2,$0,<tên system call>	addiu $2, $0, SC_ReadInt
syscall	syscall
j $31	j $31
.end <tên hàm>	.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 lên "make all".

4) 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

5) 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 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 syscall CreateFile: int CreateFile(char * name)

Dùng để tạo 1 file

Giá trị trả về: 1 nếu Tạo thành công, 0 nếu tạo thất bại

Sử dụng hệ thống filesystem để tạo file với hàm filesystem ->Create(filename,0)

3) Cài đặt System Call: OpenFileID Open(char *name, int type) và int Close(OpenFileID id)

- Open

  Hàm mở file, sử dụng hàm OpenfileId Open(char*name, int type) là một hàm hệ thống đã cung cấp

  Giá trị trả về là 1 nếu mở thành công, 0 nếu mở thất bại.

  Theo đề bài, cần có một bảng mô tả file (lưu tối đa đượ c 10 file) nên hệ thống cần có một bảng gồm 10 dòng mà mỗi dòng là một cấu trúc mô tả file.ở đây ta sử dụng cấu trúc Table thêm vào thread.h và thread.cc để phục vụ việc đọc file.

- Close

  Hàm đóng file

Giá trị trả về: 1 nếu thành công, 0 nếu thất bại

Sử dụng hàm currentThread -> gbTalbe.closeFile(idFile)

4) Cài đặt System Call: int Read (char* buffer, int charcount, OpenFileID id) và int Write (char* buffer, int charcount, OpenFileID id)

- Write

  Hàm ghi xuống file

  Giá trị trả về: 1 nếu thành công, 0 nếu thất bại

  Load thông tin vào bảng mô tả file, sau đó thể hiện thông tin lên màn hình

- Read

  Hàm đọc file

  Giá trị trả về: 1 nếu thành công, 0 nếu thất bại

  Sử dụng infoFile->of->Read trong đó infoFile = currentThread →

  gbTalbe.returnFile(idFile)

5) Cài đặt System Call: int Seek (int pos, OpenFileID id)

Dùng để thay đổi vị trí của con trỏ trong 1 file

Sử dụng thông tin trong gTable để tìm vị trí dịch chuyển của con trỏ và di chuyển tới vị trí cần thiết

Sử dụng currentThread->gbTalbe.m_Table[idFile].of->Seek(pos).currentThread->gbTalbe.m_Table[idFile].of->Seek(pos);

6) Chương trình createfile để kiểm tra System Call CreateFile

Dùng tên file cố định hoặc cho người dùng nhập vào từ console.

Ta gọi lại system call Open để mở file stdin với type quy ước bằng 2. Nếu mở file thành công thì gọi system call Read đọc tên file vừa nhập từ stdin và gọi system call CreateFile để tạo file với tham số truyền vào là tên file đọc được. Cuối cùng là đóng file stdin với system call Close.

7) Chương trình echo

Xuất lại chuỗi người dùng nhập từ console

Sử dụng syscall Scan cho người dùng nhập vào chuỗi từ bàn phím, đẩy xuống lưu trong vùng nhớ. Sau đó sử dụng syscall Print để lấy dữ liệu ra khỏi vùng nhớ và xuất lên màn hình console

8) Chương trình cat

Hiện thị nội dung của một file

Dùng syscall ReadString cho người dùng nhập vào tên file cần hiển thị, đẩy tên file xuống lưu vào trong vùng nhớ. Lấy tên file đưa vào syscall Open với type = 1 (chỉ đọc) . Trong khi còn đọc không bị lỗi và chưa đến cuối file thì đọc file với độ dài maxlength và xuất ra console

9) Chương trình copy

Sao chép nội dung từ file nguồn sang file đích

Dùng syscall ReadString cho người dùng nhập vào tên file nguồn và đích, đẩy tên file xuống lưu vào trong vùng nhớ. Lấy tên file đưa vào syscall Open với type của file nguồn là 1 (tránh bị thay đổi nội dung file nguồn) và file đích với type = 0. Trong khi còn đọc

không bị lỗi và chưa đến cuối file thì đọc file nguồn với độ dài maxlength và ghi vào file đích

2.5. DEMO

https://youtu.be/U4yNwN1ezII

2.6. 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.pdf
- **Dang Khoa-Le Tan.** *Loạt bài nachOS.* http://dangkhoahome.blogspot.com/p/nachos.html
- **Nguyễn Thành Chung.** *Loạt video hướng dẫn đồ án nachOS.* https://www.youtube.com/watch?v=t0jtY1C129s&list=PLRgTVtca98hUgCN2_2vzsAAXPiTFbvHpO