VNU - University of Science

Faculty of Information Technology

BÁO CÁO ĐỒ ÁN

Đồ án 1 : Syscall for File System

Instructors: Phạm Tuấn Sơn

Lê Viết Long

Group Members: Võ Phạm Thanh Phương - 21127677

Nguyễn Quốc Huy - 21127411

Lê Hoàng Sang - 21127158

Hệ điều hành

Second Semester - 2022 - 2023

Table of content

1 Thông tin thành viên

MSSV	Họ tên	Đóng góp
21127677	Võ Phạm Thanh Phương	100%
21127411	Nguyễn Quốc Huy	100%
21127158	Lê Hoàng Sang	100%

2 Những thứ cần chuẩn bị trước khi cài đặt System call và exception

2.1 Cài đặt trước

- Vào nachos/nachos-3.4/code/userprog, cụ thể là file exception.cc thực hiện khai báo hàm **char* User2System(int virtAddr,int limit)** và hàm **int System2User(int virtAddr,int len,char* buffer)** nhằm mục đích là thực hiện sao chép vùng nhớ giữa User space và System space

2.2 Thông tin cơ bản về các thanh ghi

- Mã syscall và kết quả trả về của mỗi syscall sẽ được đưa vào thanh ghi R2
- Tham số thứ nhất sẽ được đưa vào thanh ghi R4
- Tham số thứ hai sẽ được đưa vào thanh ghi R5
- Tham số thứ ba sẽ được đưa vào thanh ghi R6
- Tham số thứ tư sẽ được đưa vào thanh ghi R7

2.3 Các bước cài đặt system call

- Bước 1: Trong tập tin /code/userprog/syscall.h thêm dòng khai báo syscall mới

```
#define SC_Create 4
int Create(char* name);
```

- Bước 2: Trong tập tin /code/test/start.c và /code/test/start.s thêm dòng

```
    .globl Create
    .ent Create
Create:
    addiu $2,$0,SC_Create
    syscall
    j $31
    .end Create
```

- Bước 3: Trong tập tin code/userprog/exception.cc sửa điều kiện **if** ... thành **switch ... case** (chỉ sửa một lần duy nhất)

- Bước 4: Viết chương trình ở mức người dùng để kiểm tra và có sử dụng hàm đã khai báo ở trong code/userprog/syscall.h

- Bước 5: Thêm đoạn vào **Makefile** trong /code/test/
  Thêm <**tên file**> vào dòng all:

```
    all: halt shell matmult sort <file name>
```

  Thêm đoạn sau phía sau **matmult**:

```
    <file name>.o: <file name>.c
    $(CC) $(CFLAGS) -c <file name>.c
    <file name>: <file name>.o start.o
    $(LD) $(LDFLAGS) start.o <file name>.o -o <file name>.coff
    ../bin/coff2noff <file name>.coff <file name>
```

- Bước 6: Biên dịch lại Nachos, cd tới .../nachos/code viết lệnh "gmake all"

- Bước 7: Thực thi chương trình
  ./userprog/nachos -rs 1023 -x ./test/<tên file>

3 Cài đặt System call và exception

3.1 Cài đặt lại các exception

Vào file ...code/machine/machine.h có liệt kê các exception, từ đó qua file exception.cc và viết lại các case. Sau mỗi exception, thêm lệnh interrupt->Halt() để tắt hệ điều hành

3.2 Cài đặt hàm inc_program_counter()

Thực hiện tăng giá trị Programming Counter để nạp lệnh tiếp theo để thực hiện. Dùng giá trị của PC hiện tại để lưu giá trị của PC trước, nạp giá trị kế cho PC hiện tại và nạp giá trị kế tiếp nữa cho PC kế.

3.3 Cài đặt system call int CreateFile(char *name)

- . CreateFile system call sẽ sử dụng Nachos FileSystem Object để tạo một file rỗng. Chú y' rằng filename đang ở trong user space, có nghĩa là buffer mà con trỏ trong user space trỏ tới phải được chuyển từ vùng nhớ user space tới vùng nhớ system space. System call CreateFile trả về 0 nếu thành công và -1 nếu có lỗi

- Chức năng của **SC_CreateFile**:
  Input: Địa chỉ tên file ở User space
  Output: -1 nếu lỗi và 0 nếu thành công
  Mục đích: tạo một file rỗng với argument là tên file

- Đầu tiên, ta lấy tham số tên tập tin từ thanh ghi **r4**, sau đó sao chép giá trị lưu ở **r4** từ User space sang System space bằng hàm **User2System()**. Sau đó ta thực hiện kiểm tra coi tên file có phải NULL hay không. Nếu không phải thì trả về 0, ngược lại thì trả về -1 vào thanh ghi **r2**

3.4 Cài đặt system call OpenFileID Open(char *name, int type) và int Close(OpenFileID id)

3.4.1 SC_Open

- User program có thể mở 2 loại file, file chỉ đọc và file đọc và ghi. Mỗi tiến trình sẽ được cấp một bảng mô tả file với kích thước cố định. Đồ án này, kích thước của bảng mô tả

file là có thể lưu được đặc tả của 10 files. Trong đó, 2 phần tử đầu, ô 0 và ô 1 để dành cho console input và console output.

- System call mở file phải làm nhiệm vụ chuyển đổi địa chỉ buffer trong user space khi cần thiết và viết hàm xử ly phù hợp trong kernel. Sử dụng class **FileSystem** trong file **filesys**, System call Open sẽ trả về id của file (OpenFileID = một số nguyên), hoặc là -1 nếu bị lỗi.

- Mở file có thể bị lỗi như trường hợp là không tồn tại tên file hay không đủ ô nhớ trong bảng mô tả file. Tham số **type** = 0 cho mở file đọc và ghi, = 1 cho file chỉ đọc. Nếu tham số truyền bị sai thì system call phải báo lỗi. System call sẽ trả về -1 nếu bị lỗi và 0 nếu thành công.

- Chức năng của **SC_Open**:
  Input: Địa chỉ của tên file, chế độ mở
  Output: trả về Id của file nếu thành công, -1 nếu lỗi
  Mục đích: thực hiện mở file với tham số truyền vào là tên và chế độ mở

- Đầu tiên, ta đọc địa chỉ tên file từ thanh ghi **r4** và tham số type từ thanh ghi **r5**, sau đó ta kiểm tra xem vị trí của file trong fileSystem có nằm trong bảng mô tả file có kích thước là 10 file không (0<=pos<=9). Nếu không thì trả về -1. Tiếp theo, ta sao chép địa chỉ tên file từ User space sang System space bằng hàm **User2System()** và tiếp tục kiểm tra xem nếu tên file trùng với "stdin" thì trả về thanh ghi **r2** id của file là 0