ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO ĐỒ ÁN 1**

**EXCEPTION VÀ CÁC SYSCALL ĐƠN GIẢN**

Giảng viên lý thuyết: Lê Giang Thanh

Giảng viên hướng dẫn thực hành: Nguyễn Thanh Quân

Thành viên nhóm:

21120418 – Nguyễn Chí Công

21120449 – Nguyễn Văn Hậu

21120464 – Nguyễn Quốc Hưng

**Thành phố Hồ Chí Minh – 2023**

**MỤC LỤC**

[1. THÔNG TIN NHÓM 4](#_Toc150676717)

[2. TỔNG QUAN ĐỒ ÁN 6](#_Toc150676718)

[2.1. Phiên bản và công cụ làm việc 6](#_Toc150676719)

[2.2. Các giá trị thanh ghi 6](#_Toc150676720)

[2.3. Các hàm hỗ trợ 6](#_Toc150676721)

[3. SYSTEMCALL THAO TÁC VỚI FILE 6](#_Toc150676722)

[3.1: System call Create 6](#_Toc150676723)

[3.2. System call Open 7](#_Toc150676724)

[3.3. System call Close 8](#_Toc150676725)

[3.4 System call Read 8](#_Toc150676726)

[3.5. System call Write 8](#_Toc150676727)

[3.6. System call Seek 8](#_Toc150676728)

[3.7. System call Remove 9](#_Toc150676729)

[4. SYSYEM CALL THAO TÁC VỚI NETWORK TCP 10](#_Toc150676730)

[4.1. System call SocketTCP 10](#_Toc150676731)

[4.2. System call Connect 10](#_Toc150676732)

[4.3. System call Send 11](#_Toc150676733)

[4.4. System call Receive 11](#_Toc150676734)

[4.5. System call Close 11](#_Toc150676735)

[5. CHƯƠNG TRÌNH TEST 12](#_Toc150676736)

[5.1. Chương trình createfile 12](#_Toc150676737)

[5.2. Chương trình cat 12](#_Toc150676738)

[5.3. Chương trình copy 12](#_Toc150676739)

[5.4. Chương trình delete 12](#_Toc150676740)

[5.5 Chương trình concatenate 12](#_Toc150676741)

[5.6 Chương trình socket echo 12](#_Toc150676742)

[5.7. Chương trình socket truyền file 13](#_Toc150676743)

[6. TÀI LIỆU THAM KHẢO 13](#_Toc150676744)

# 1. THÔNG TIN NHÓM

|  |  |  |  |
| --- | --- | --- | --- |
| Họ và tên | MSSV | Công việc | Mức độ hoàn thành |
| Nguyễn Chí Công | 21120418 | Viết báo cáo | 100% |
| Nguyễn Văn Hậu | 21120449 | Syscall Read, Write, Seek, Remove, viết các file .c test syscall file | 100% |
| Nguyễn Quốc Hưng | 21120464 | Syscall socket, syscall Create, Open, Close, các file .c test sysccall socket | 100% |

**Mức độ hoàn thành yêu cầu**

|  |  |  |  |
| --- | --- | --- | --- |
| Phần | Câu | Ghi chú | Mức độ hoàn thành |
| 1 | 1 | Syscall Create | 100% |
|  | 2 | Syscall Open | 100% |
|  | 3 | Syscall Read | 100% |
|  | 4 | Syscall Write | 100% |
|  | 5 | Syscall Remove | 100% |
| 2 | 1 | Syscall socketTCP | 100% |
|  | 2 | Syscall Connect | 100% |
|  | 3 | Syscall Send | 100% |
|  |  | Syscall Receive | 100% |
| 3 |  | Nâng cao | 0% |
| 4 |  | Chương trình test |  |
|  | 1,2,3,4,5 | create, copy, cat, delete, concatenate | 100% |
|  | 6 | Echo | 100% |
|  | 7 | Truyền file | 100% |
|  |  | Báo cáo | 100% |

# 2. TỔNG QUAN ĐỒ ÁN

## 2.1. Phiên bản và công cụ làm việc

* Hệ điều hành sử dụng: Ubuntu 18.04.4 LTS (x86)
* Compiler: gcc, g++  v4.8.4
* IDE: Visual studio code

## 2.2. Các giá trị thanh ghi

* R2: Lưu mã system call và giá trị trả về của nó
* R4: Lưu tham số đầu tiên
* R5: Lưu tham số thứ 2
* R6: Lưu tham số thứ 3
* R7: Lưu tham số thứ 4

## 2.3. Các hàm hỗ trợ

* IncreasePC: tăng program counter
* User2System: copy vùng nhớ từ User space sang System space
* System2User: copy vùng nhớ từ System space sang User space
* PrintString: in chuỗi ra console
* ReadString: đọc chuỗi từ console

# 3. SYSTEMCALL THAO TÁC VỚI FILE

## 3.1: System call Create

- Input: tên file (char\* name)

- Output: 0 nếu thành công hoặc -1 nếu thất bại

- Chi tiết cài đặt:

* Tạo 2 biến fileCreateAddr: lưu địa chỉ ảo và filenameCreate: lưu tên tập tin
* Ta lấy tên của tập tin bằng cách dùng hàm ReadRegister để đọc giá trị của thanh ghi số 4 và dùng hàm User2System để lấy tên tập tin.
* Nếu trường hợp tên rỗng ta xuất lỗi ra màn hình và dùng WriteRegister trả về -1
* Dùng hàm Create của lớp filesystem để tạo file, nếu file trả về false thì xuất lỗi ra màn hình và dùng WriteRegister trả về -1, ngược lại dùng WriteRegister để trả về 0
* Sau khi xử lý xong dùng hàm IncreasePC để tăng giá trị của Program Counter

## 3.2. System call Open

- Input: tên file cần mở (char \* name) và kiểu file (int type)

- Ouput: trả về vị trí của file trong bảng 20 file discriptor nếu mở thành công, hoặc -1 nếu thất baị

- Chi tiết cài đặt:

* Tạo thêm hàm Open với file type trong openfile.h. Quy ước các giá trị 0, 1, 2, 3 lần lượt chỉ đạng của file là “Read and Write”, “Read Only”, “Console Input” và “Console Output”.
* Để quản lí mảng 20 file descriptors, khai báo thêm 2 biến OpenFile\*\* of và int type. Default lại constructor và destructor của filesys.h và filesys.cc.
* Tại hàm constructor của FileSystem ta thêm phần tạo mảng of có 20 phần tử để quan lý đặc tả của 20 file descriptors sau đó ta tạo 2 file đầu tiên lần lượt cho console input và console output. Tại destructor ta xóa đi mảng of.
* Tạo thêm constructor OpenFile với file type và method GetCurrenPos để lấy vị trí hiện tại của file trong mảng 20 file tại class OpenFile
* Lấy tên của file tương tự như trường hợp CreateFile, sau đó ta lấy giá trị type của file từ thanh ghi số 5.
* Tìm vị trí còn trống của filesystem, nếu không tìm được thì ta không xử lý mà trả về lỗi cho hệ điều hành. Ngược lại ta xét 3 trường hợp: nếu type của file system là 0 hoặc 1 thì ta mở file, thêm tên file vào fileNames và trả về file id, nếu type của file system là 2 thì ta trả về file id là 0 (vị trí trong bảng of dành cho console in), nếu là 3 thì ta trả về file id là 1 (vị trí trong bảng of dành cho console out).
* Trước khi trả về ta phải dùng hàm IncreasePC để tăng program counter.

## 3.3. System call Close

**-** Input: ID file đang mở

- Output: 0 nếu thành công, hoặc -1 nếu lỗi

- Chi tiết cài đặt:

* Lấy ID của file từ thanh ghi r4
* Nếu ID file nằm trong khoảng 20 file discriptors thì tiến hành các bước: xóa file tại vị trí ID file trong bảng of -> set vị trí vừa xóa = NULL -> trả về giá trị 0
* Ngược lại thì trả về -1

## 3.4 System call Read

- Input: một buffer chứa nội dung cần đọc, số kí tự (int size) và id của file (OpenFileID id)

- Output: số byte đọc được nếu thành công, hoặc -1 nếu lỗi

+ Chi tiết cài đặt:

* Đọc từ thanh ghi r4 địa chỉ buffer, thanh ghi r5 chiều dài chuỗi kí tự, thanh ghi r6 ID file
* Xử lý các trường hợp không đọc được 🡪 trả về -1:
* ID file không nằm trong bảng 20 file discriptors
* File đó có giá trị = NULL
* Type của file là 3 (quy ước là console output)
* Lấy vị trí hiện tại của con trỏ file bằng hàm GetCurrentPos(), dùng User2System để copy
* Xử lý trường hợp console input: dùng hàm GetChar để lấy từng kí tự trong file. Mỗi lần lấy từng kí tự đưa vào buffer sẽ tăng biến size lên 1. Trả về r2 số byte đọc được là size nếu size > 0, ngược lại trả về -1
* Xóa vùng nhớ buffer, sau khi xử lý từng trường hợp thì tăng Program Counter

## 3.5. System call Write

- Input: một buffer chứa nội dung cần ghi, số kí tự (int size) và id của file (OpenFileID id)

- Output: số byte ghi được nếu thành công, hoặc -1 nếu lỗi

- Chi tiết cài đặt

* Đọc từ thanh ghi r4 địa chỉ buffer, thanh ghi r5 chiều dài chuỗi kí tự, thanh ghi r6 ID file
* Xử lý các trường hợp không ghi được:
* ID file không nằm trong bảng 20 file discriptors
* file đó có giá trị = NULL
* type của file là 1: read only hoặc 2: console input
* Lấy vị trí hiện tại của con trỏ file bằng hàm GetCurrentPos(), dùng User2System để copy
* Xử lý trường hợp 0: Read và Write: dùng fileSystem -> Write để ghi, sau đó lấy vị trí con trỏ mới bằng GetCurrentPos, số byte đọc được là vị trí mới - vị trí cũ. Trả về số byte đọc được
* Xử lý trường hợp 3: console output: dùng hàm PutChar của synchConsoleOut để đưa từng kí tự lên trong buffer lên, mỗi lần đưa lên thì số byte đọc được tăng thêm 1. Trả về số byte ghi được
* Xóa vùng nhớ buffer và tăng Program Counter

## 3.6. System call Seek

- Input:

* posSeek: Vị trí cần chuyển tới hoặc -1 nếu muốn di chuyển đến cuối file.
* idSeek: File descriptor của file cần thực hiện Seek.

- Output:

* Nếu thành công, trả về vị trí mới trong file.
* Nếu lỗi, trả về -1.

- Chi tiết cài đặt:

* Đọc posSeek và idSeek từ thanh ghi của máy.
* Kiểm tra tính hợp lệ của idSeek và xác định xem file có tồn tại không.
* Kiểm tra xem có thể thực hiện Seek trên file không (không thể trên console).
* Nếu posSeek bằng -1, di chuyển con trỏ đến cuối file, ngược lại, di chuyển đến vị trí được chỉ định.
* Kiểm tra xem vị trí mới có hợp lệ không (không vượt quá kích thước của file).
* Nếu thành công, trả về vị trí mới, ngược lại, trả về -1.

## 3.7. System call Remove

- Input:

* addrRemove: Địa chỉ của chuỗi tên file cần xóa.

- Output:

* Nếu thành công, trả về 0.
* Nếu lỗi (file không tồn tại hoặc đang mở), trả về -1.

- Thuật toán:

* Đọc addrRemove từ thanh ghi của máy.
* Chuyển đổi địa chỉ của chuỗi tên file từ user space sang system space.
* Kiểm tra tính hợp lệ của địa chỉ và xác định xem file có tồn tại không.
* Kiểm tra xem file có đang mở không. Nếu có, trả về -1.
* Nếu không có lỗi, thực hiện Remove và trả về kết quả.

# 4. SYSYEM CALL THAO TÁC VỚI NETWORK TCP

- Định nghĩa thêm các hàm với socket trong sysdep.h và sysdep.cc

**-** Tạo file ksocket.h để default class Socket với các method cần thiết

- Các syscall socket sẽ ghi ở file ksyscall.h để gọi các method trong class Socket

- Exception.cc sẽ xử lí các syscall

## 4.1. System call SocketTCP

- Input:

- Ouput: trả về id file descriptor, hoặc -1 nếu lỗi

- Chi tiết cài đặt:

* Method Insert của Socket sẽ tìm vị trí trống trong mảng file descriptor. Nếu vị trí khoog nằm trong mảng thì trả về -1, ngược lại sẽ OpenSocket và trả về ID
* Ghi vào r2 giá trị vừa được trả về

## 4.2. System call Connect

- Input: id của socket, IP và PORT

- Output: 0 nếu thành công, hoặc -1 nếu thất bại

- Chi tiết cài đặt:

* Lấy ID từ r4, Addr từ r5 và PORT từ r6
* Tạo một địa chỉ IP
* Method Connect sẽ connect tới socket , trả về -1 nếu thất bại, 0 nếu kết nối thành công
* Ghi vào r2 kết quả

## 4.3. System call Send

- Input: id của socket, chuỗi buffer và độ dài

- Output: trả về số lượng byte gửi đi nếu thành công, 0 nếu kết nối bị đóng hoặc -1 nếu thất bại

- Chi tiết cài đặt:

* Lấy ID từ r4, Addr từ r5 và độ dài buffer từ r6
* Tạo một địa chỉ IP
* Method Send tìm vị trí trong mảng descriptor trùng id, sau đó gọi SendMessageToSocket để gửi
* Trả về 0 nếu thành công, -1 nếu thất bại
* Ghi kết quả vào r2

## 4.4. System call Receive

- Input: id của socket, chuỗi buffer và độ dài

- Output: trả về số lượng byte nhận được nếu thành công, 0 nếu kết nối bị đóng hoặc -1 nếu thất bại

- Chi tiết cài đặt:

* Lấy ID từ r4, Addr từ r5 và độ dài buffer từ r6
* Tạo một địa chỉ IP
* Method Receive tìm vị trí trong mảng descriptor trùng id, sau đó gọi ReceiveMessageToSocket để nhận
* Trả về 0 nếu thành công, -1 nếu thất bại
* Ghi kết quả vào r2

## 4.5. System call Close

- Input: id của socket

- Output: 0 nếu thành công, hoặc -1 nếu thất bại

- Chi tiết cài đặt:

* Lấy ID từ r4
* Method Close tìm vị trí trong mảng descriptor trùng id, sau đó gọi CloseSocket để đóng. Trả về 0 nếu thành công, -1 nếu thất bại
* Ghi vào r2 kết quả

# 5. CHƯƠNG TRÌNH TEST

## 5.1. Chương trình createfile

Nội dung: người dùng nhập tên file từ console

Kết quả: tạo file mới với tên được nhập từ console

## 5.2. Chương trình cat

Nội dung: yêu cầu người dùng nhập tên file, sau đó hiển thị nội dung file đó

Kết quả:hiển thị nội dung của file trên console

## 5.3. Chương trình copy

Nội dung: người dùng nhập tên file nguồn và file đích, sau đó thực hiện copy file

Kết quả: file đích chứa nội dung được copy từ file nguồn

## 5.4. Chương trình delete

Nội dung: kiểm tra system call Delete

Kết quả: xóa file theo tên mà người dùng nhập từ console

## 5.5 Chương trình concatenate

Nội dung: yêu cầu nhập tên file nguồn 1 và file nguồn 2, sau đó nối nội dung 2 file

Kết quả:tạo file mới có nội dung được nối từ 2 file nguồn

## 5.6 Chương trình socket echo

Nội dung: client gửi một message, server nhận massage và biến chuỗi này thành chữ hoa, sau đó gửi lại cho client

Kết quả: xuất ra màn hình chuỗi in hoa

## 5.7. Chương trình socket truyền file

Nội dung: client đọc và gửi nội dung file text cho server. Server nhận và biến thành chữ hoa, sau đó gửi lại cho client

Kết quả: ghi chuỗi chữ hoa vào file

# 6. TÀI LIỆU THAM KHẢO

1. [Socket TCP](https://web.ics.purdue.edu/~cs354/Nachos/network_8cc-source.html)

2. [Video demo syscall](https://www.youtube.com/watch?v=t0jtY1C129s&list=PLRgTVtca98hUgCN2_2vzsAAXPiTFbvHpO)