

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN MÔN HỌC
HỆ ĐIỀU HÀNH**

**SYSTEM CALLS & FILE
NETWORK OPERATIONS**

Giảng viên lý thuyết: Lê Giang Thanh

Giảng viên thực hành: Lê Hà Minh, Nguyễn Thanh Quân

Mục lục

I. Thông tin nhóm:	3
II. Đánh giá mức độ hoàn thành:	4
III. Lập trình:	5
IV. Phân công:	8
V. Tham khảo:	8

I. Thông tin nhóm:

Tên thành viên

MSSV

Nguyễn Trần An Hòa

21127047

Phạm Phú Toàn

21127183

Tô Khánh Linh

21127683

II. Đánh giá mức độ hoàn thành:

Phần		Yêu cầu	Hoàn thành
1	1	syscall Create	100%
	2	syscall OpenFileID, Close	100%
	3	syscall Read, Write	100%
	4	syscall Seek	100%
	5	syscall Remove	100%
2	1	syscall socketTCP	100%
	2	syscall Connect	100%
	3	syscall Send	100%
		syscall Receive	100%
3		Advanced	100%
4		Test program	
	1,2,3,4,5	create, copy, cat, delete, concatenate	100%
	6	echo	100%
	7	file transfer	100%
Mức độ hoàn thành: 100%			

III. Lập trình:

a. Các hàm chính:

i. Syscall Create:

- Đọc địa chỉ của file từ thanh ghi số 4. Lấy tên của file cần tạo chuyển từ bộ nhớ của người dùng sang hệ thống. Sau đó xem xét các trường hợp xảy ra như không đọc được tên file, tên file có độ dài bằng 0 thì sẽ trả về -1 vào thanh ghi số 2 (báo cho người dùng là không thành công).
- Nếu đọc tên file thành công, sử dụng hàm Create(char *) của lớp FileSystem để tạo file, nếu hàm trả về true, tạo file thành công – trả về 0 vào thanh ghi số 2. Ngược lại, trả -1 vào thanh ghi số 2.

ii. Syscall OpenFileID:

- Các tham số đầu vào lần lượt là char * name, int type. Đọc địa chỉ của file từ thanh ghi số 4 và đọc loại muốn nhập vào từ thanh ghi số 5. Sau đó, đưa tên file từ bộ nhớ người dùng về hệ thống.
- Tạo một OpenFileId **openfile gồm 20 phần tử trong lớp FileSystem, trong đó, tiếp tục xét xem mảng 2 chiều đó vẫn còn chỗ trống bằng hàm isFull(). Nếu vẫn còn sẽ trả về vị trí index còn trống, ngược lại trả về người dùng -1 vào thanh ghi số 2 (thông báo đã hết chỗ trống).
- Trong bảng hai chiều, tạo sẵn hai vị trí đầu tiên là hai file stdin và stdout; trong đó stdin có type=1 và stdout có type=2.
- Xét các trường hợp không đọc được tên file và tên file có độ dài bằng 0 để báo trả về cho người dùng.
- Nếu đọc tên file thành công, xét đến trường hợp, người dùng nhập vào type =0 và type =1; nếu khác hai loại trên, sẽ trả về không thành công vì sai type.
- Đối với type=0 là cho phép mở file để đọc và viết bình thường, và type =1 là cho phép mở file chỉ để đọc; sử dụng hàm Open có sẵn của hệ thống để mở file. Nếu thành công báo về vị trí index của file trong bảng hai chiều, ngược lại báo về -1.

- Mỗi khi mở file thành công sẽ lưu lại tên file đã mở trong mảng filename được tạo ở lớp FileSysteME để sau này kiểm tra xem file đã được mở hay chưa.

iii. Syscall Close:

- Đọc id của file cần đóng từ thanh ghi số 4.
- Gọi hàm hủy của lớp OpenFile, và trả về Null tại vị trí index của mảng filename.

iv. Syscall Read:

- Lần lượt đọc địa chỉ của buffer, charcount và id của file từ các thanh ghi 4,5,6.
- Xét các trường hợp nếu id <0 hoặc id >=20 hoặc trong mảng hai chiều openfile[id]=NULL, thì trả về -1 (thông báo không thành công). Copy chuỗi buffer từ vùng nhớ user space sang system space với độ dài charcount.
- Đối với id=0 (quy ước id=0 là vị trí của stdin), Sử dụng hàm Read của lớp SynchConsole để đọc stdin và trả về số byte thực sự đọc được.
- Chỉ xét trường hợp type =0 hoặc type =1 để trả về vị trí số byte thực sự. Nếu không thành công trả về -1.

v. Syscall Write:

- Tương tự giống với syscall Write, tuy nhiên, nếu id=0 hoặc type=1 thì không thể ghi.
- Đối với id=1 (quy ước id=1 là vị trí của stdout), sử dụng hàm cho sẵn của lớp SynchConsole để write. Trả về số Byte thực sự write được.
- Nếu type=0, write như file bình thường và trả về hệ thống số byte write được.

vi. Syscall Seek:

- Xét các trường hợp id=0, id=1 và tại vị trí trong openfile= NULL và type=2, thì sẽ báo trả về không thành công -1.
- pos là vị trí muốn tìm đến, trong đó nếu pos lớn hơn độ dài file thì sẽ không seek được.

- Nếu pos=-1 thì đưa con trỏ về vị trí cuối file.
- Nếu pos là giá trị bình thường, sử dụng hàm seek của hệ thống để đưa con trỏ về vị trí cần thiết. Trả lại vị trí của pos.

vii. Syscall Remove:

- Xét các trường hợp tương tự như các syscall trên, trả về -1 nếu gặp lỗi với tên filename (được nhận địa chỉ từ thanh ghi số 4),
- Nếu file đã được mở, thì sẽ không thể xóa file.
- Sử dụng hàm remove được cho sẵn để xóa file.

viii. Syscall Socket:

- Kiểm tra xem đã có đủ 20 socket tạo ra chưa.
- Sử dụng hàm tạo socket từ thư viện <sys/socket.h>
- Nếu thành công trả về vị trí index của nó trong bảng, ngược lại trả về -1.

ix. Syscall Connect:

- Lấy socketID từ thanh ghi số 4, địa chỉ của tham số ip từ thanh ghi số 5, lấy port từ thanh ghi số 6.
- Sau đó kiểm tra socketID có nằm trong khoảng cho phép hay không, và kiểm tra xem socket tại vị trí id đó đã được tạo hay chưa.
- Sử dụng hàm connect của thư viện, thành công trả về cho thanh ghi số 2 là 0 và -1 nếu không thành công.

x. Syscall Send:

- Lấy địa chỉ của buffer, charcount và id của file từ lần lượt từ các thanh ghi 4, 5, 6.
- Kiểm tra các điều kiện của id, và xem liệu socketid đã được tạo.
- Sử dụng hàm send có sẵn của thư viện, trả về số byte thực sự nếu đã gửi thành công và ngược lại trả về -1 nếu không thành công.

xi. Syscall Receive:

- Tương tự giống hàm syscall Send sử dụng hàm Receive để nhận message từ server gửi về client. Nếu nhận thành công trả về số byte đã nhận được, ngược lại trả về -1.

b. Các hàm phụ:

- i. Syscall PrintString: In ra chuỗi kí tự
- ii. Syscall ReadString: nhận chuỗi kí tự nhập từ bàn phím.
- iii. Syscall PrintInt: in ra số nguyên.

c. Advanced Program:

- Dựa vào tính kế thừa và đa hình, để viết bảng file descriptor của file và network chung (openfile[20]). Và quy định thêm type=2 nếu file tạo ra là Socket.
- ở các syscall Read, syscall Write của file, xét thêm trường hợp type =2, tương ứng với send và receive nếu type bằng 2 thì sẽ chạy các hàm tương ứng với socket.

IV. Phân công:

Thành viên	Công việc
Nguyễn Trần An Hòa	Syscall Open, Close, create, remove, seek, socket
Phạm Phú Toàn	Socket , Advanced
Tô Khánh Linh	Syscall Read, Write, Test Program

V. Tham khảo:

- Tài liệu của thầy Nguyễn Thanh Quân.