

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN ĐHQG-HCM

KHOA CÔNG NGHỆ THÔNG TIN



PROJECT 2 - Tìm hiểu system call - Các thao tác với files

Môn học: Hệ điều hành

Giảng viên:

Nguyễn Văn Giang

Lê Quốc Hòa

Sinh viên thực hiện:

Lê Ngọc Tường 20127383

Nguyễn Tấn Phát 20127588

Huỳnh Vĩ Khang 20127663

1. Thông tin nhóm:

STT	MSSV	HỌ VÀ TÊN	ĐÁNH GIÁ
1	20127383	Lê Ngọc Tường	Tốt
2	20127588	Nguyễn Tấn Phát	Tốt
3	20127663	Huỳnh Vĩ Khang	Tốt

2. Mức độ hoàn thiện:

⇒ **100 %**

3. Môi trường làm việc:

- VMware Workstation 12 Player.
- Github
- CentOS (Linux)

4. Bảng phân công công việc:

Create	Đã hoàn thành	Tấn Phát
Open	Đã hoàn thành	Tấn Phát
Close	Đã hoàn thành	Ngọc Tường
Read	Đã hoàn thành	Vĩ Khang
Write	Đã hoàn thành	Tấn Phát
Seak	Đã hoàn thành	Ngọc Tường
Remove	Đã hoàn thành	Vĩ Khang
Sao chép vùng nhớ kernel-user	Đã hoàn thành	Cả nhóm

createfile	Đã hoàn thành	Tấn Phát
cat	Đã hoàn thành	Ngọc Tường
copy	Đã hoàn thành	Vĩ Khang
delete	Đã hoàn thành	Tấn Phát
concatenate	Đã hoàn thành	Ngọc Tường
Không để user làm sập hdh	Đã hoàn thành	Cả Nhóm
Phân công, lên lịch họp nhóm, kiểm tra tiến độ (ở github)		Tấn Phát
Tổng hợp và viết báo cáo		Vĩ Khang

5. Nội dung tìm hiểu:

5.1 Cài đặt system call thao tác với file:

- Chỉnh sửa các class có sẵn trong file `openFile.h`, `openFile.cc`, `fileSys.h`, `fileSys.cc` thành các class tối ưu hơn cho đề án.

1. Create: Tham khảo ở file [2] GIAO TIẾP GIỮA HDH NACHOS VÀ CHƯƠNG TRÌNH NGƯỜI DÙNG.PDF

`int Create(char * name)`. Trả về (0) nếu tạo file thành công và trả về (-1) nếu tạo file thất bại.

Ở file `exception.cc`

Kiểm tra `*name == NULL` hoặc `len(name) == 0`. Nếu có thì báo lỗi (-1) rồi return.

Kiểm tra hàm `fileSystem->Create(name, 0)` có chạy thành công hay không. Nếu không thì báo lỗi (-1) rồi return. Ngược lại, báo thành công (-1) và return.

2. Open:

`OpenFileID Open(char *name)`. Trả về (-1) nếu mở file thất bại. Trả về ID File hay thứ tự mở file (từ 2 đến 9) nếu mở file thành công.

Ở file `fileSys.h`, thêm thuộc tính:

OpenFile* openf[10]: Là mảng gồm 10 con trỏ OpenFile. Với openf[0] và openf[1] là stdin và stdout

Ở file *filesys.cc*, thêm các hàm:

Int FindFreeSlot(): Tìm slot còn trống từ 2-9 rồi trả về vị trí phần tử còn trống đầu tiên trong openf[10]. Vị trí thứ i tương đương với OpenFileID = i.

OpenFile* Open(char* name) trả về openf thỏa mãn.

Ở file *exception.cc*:

Dùng hàm FindFreeSlot để tìm slot đang trống.

Gán openf[freeSlot] = Open(filename). Nếu thành công thì trả về thanh ghi freeSlot. Ngược lại trả về giá trị (-1)

3. Close: int Close(OpenFileID id): kiểm tra id hợp lệ và file đang mở, nếu đúng thì đóng file và trả về 0, ngược lại trả về -1.

- Các thay đổi trong cho việc viết syscall Close: thay đổi kiểu trả về của Close (nacho để kiểu trả về của Close là void) thành int bằng cách sửa prototype hàm Close trong syscall.h; sửa prototype và thân hàm Close trong sysdep.h và sysdep.cc (thuộc thư mục machine) trả về kiểu int và thêm lệnh return 0 cuối hàm.

4. Read: là system call đọc file từ 1 file có sẵn nếu file này không tồn tại hoặc nằm ngoài bảng mô tả file (tức nằm ngoài phạm vi hoạt động của biến openf) hoặc số tí tự đọc được < 0 hoặc file cần đọc là stdin và stdout thì system call sẽ trả về lỗi cho hệ điều hành). Ngược lại nếu đọc thành công thì sẽ có 2 trường hợp xảy ra.

Đầu tiên ta sẽ copy chuỗi từ vùng nhớ userspace sang systemspace vào 1 chuỗi đệm là buf dài bằng với độ dài của nội dung trong file.

- TH1 là đọc file rỗng -> trả về -2.

- Th2 là đọc file đã có nội dung -> đọc nội dung file từ chuỗi buf ở trên đồng thời copy chuỗi đó trả về lại cho vùng nhớ userspace số byte thật sự.

- Các thay đổi trong thư mục filesys cho việc viết syscall Read là: thêm hàm GetCurrentPos() (hàm trả về vị trí con trỏ hiện tại trong file), biến openf được khai báo thêm trong class Filesystem để kiểm tra tình trạng hiện tại của file

5. Write:

Int Write(char *buffer, int size, OpenFileID id). Trả về (-1) nếu việc writeFile thất bại. Trả về (0) nếu thành công.

Kiểm tra các id có thỏa mãn hay không. Nếu không thì trả về (-1) rồi return.

Sử dụng hàm Write(buf, charcount) để ghi file.

6. Seek: int Seek(int position, OpenFileID id): kiểm tra id hợp lệ và file đang mở, không thực hiện tìm kiếm trên file input và output; tiếp tục kiểm tra pos đầu vào (nếu -1 nghĩa là nhảy đến cuối file, nếu nằm ngoài 0 đến độ dài file thì trả về lỗi); nếu các điều kiện trên đúng thì tìm đến vị trí pos và trả về vị trí con trỏ của file, ngược lại trả về -1.

- Các thay đổi trong thư mục filesys cho việc viết syscall Seek: viết thêm hàm int GetCurrentPos() (hàm trả về vị trí con trỏ hiện tại trong file) cho hai class OpenFile trong file openfile.h; viết thêm hàm int Seek(int pos) (nhảy đến vị trí pos) cho class OpenFile nằm trên trong file openfile.h.

7. Remove: ngược lại với Create ở trên, Remove cũng kiểm tra sự tồn tại của file trước khi xóa, hoặc sẽ không thể xóa nếu file đang mở, những trường hợp còn lại sẽ xóa file bình thường và thông báo cho người.

5.2 Viết chương trình người dùng:

Chương trình CreateFile:

- Người dùng nhập tên file = name. Sử dụng hàm Create(name) để tạo. Nếu kết quả trả về -1 thì thất bại. Ngược lại thì thành công.

Chương trình Cat.c:

- Người dùng nhập tên file, thực hiện mở file nếu mở không thành công thì dừng chương trình, nếu thành công thì đọc từng kí tự từ file và in ra, khi đến cuối file (hàm Read() trả về -2) thì break khỏi vòng lặp, cuối cùng đóng.

Chương trình copy.c:

- Người dùng sẽ được nhập vào 2 tên file của file nguồn và file đích, chương trình sẽ thực hiện read tất cả các thông tin có từ file nguồn, Seek tới cuối file nguồn để có được độ dài của chuỗi đó. Cuối cùng là tạo 1 file mới có tên của file đích và write vào nó.

Chương trình delete.c:

- Người dùng nhập tên file = name. Sử dụng hàm Remove(name, -1) để xóa. Nếu kết quả trả về -1 thì thất bại. Ngược lại thì thành công.

Chương trình concatenate.c:

- Người dùng nhập tên 2 file, thực hiện mở 2 file nếu mở không thành công thì dừng chương trình, nếu thành công thì đọc từng kí tự từ file thứ 2 và ghi vào file thứ 1, khi đến cuối file thứ 2 (hàm Read() trả về -2) thì break khỏi vòng lặp, cuối cùng đóng.

6. Tài liệu tham khảo:

[1] Bien dich va Cai dat Nachos.pdf

[2] Giao tiep giua HDH Nachos va chuong trinh nguoi dung.pdf

[3] Cach Viet Mot SystemCall.pdf

[4] Cach Them 1 Lop Vao Nachos.pdf

[5] https://mystudyhcmus.files.wordpress.com/2017/09/nachos_canban.doc

[6] <https://github.com/nguyenthanchungfit/Nachos-Programing-HCMUS>