

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



Báo cáo đồ án 2

Môn: Hệ điều hành

TÌM HIỂU SYSTEM CALLS - CÁC THAO TÁC VỚI FILE

Giáo viên hướng dẫn:

Trần Trung Dũng

Nguyễn Thanh Quân

Lê Thanh Giang

Sinh viên thực hiện:

Nguyễn Hoàng Huy

Lưu Nguyễn Tiến Anh

Phan Quốc Kỳ

Lớp: 20CTT1TN

TP. Hồ Chí Minh, 2022

Bảng phân công công việc:

MSSV	Họ tên	Công việc
20120011	Nguyễn Hoàng Huy	Create, Seek, cat, concatenate
20120017	Phan Quốc Kỳ	Open, Close, Remove, copy
20120245	Lưu Nguyễn Tiến Anh	Read, Write, createfile, delete

Trình bày các System calls:**I. Phần 1: Cài đặt system calls thao tác với file****1. Cài đặt int Create(char* name)**

Bước 1: Lấy địa chỉ thanh ghi của name rồi lấy file name bằng cách chuyển từ user mode sang system mode.

Bước 2: Xét các trường hợp ngoại lệ, báo lỗi nếu có.

Bước 3: Gọi hàm Create từ kernel và trả kết quả về thanh ghi số 2.

Bước 4: Tăng PC

2. Cài đặt OpenFileID Open(char* name, int type), int Close(OpenFileID id)**- Cài đặt syscall Open:**

Đầu vào tên file name, và type (0: cho phép đọc viết, 1: chỉ đọc). Đầu ra là OpenFileID là một số (index của file đang được mở trong hệ thống file).

Bước 1: Đọc tham số truyền vào thông qua thanh ghi 4,5

Bước 2: Kiểm tra type có thỏa yêu cầu không. Chuyển tên file từ User Space sang System Space

Bước 3: Đọc index từ lớp FileSystem để kiểm tra xem còn index nào còn trống không, nếu hết thì báo lỗi. Tìm index trống

Bước 4: Sử dụng phương thức open của lớp FileSystem để mở file với type tương ứng và gán vào bảng tableDescriptor (là 1 file table descriptor với kích thước 20, lưu index và tên file). Tăng index

Bước 5: Tăng thanh ghi PC

- Cài đặt syscall Close:

Đầu vào là OpenFileID, đầu ra là một số nguyên 0 nếu thành công, -1 nếu thất bại.

Bước 1: Đọc tham số truyền vào từ thanh ghi số 4

Bước 2: Kiểm tra xem OpenFileID có nằm trong vùng cho phép của file table descriptor không, và tại OpenFileID đó có file nào được mở không

Bước 3: Nếu có file đang được mở tại OpenFileID thì xóa file đó khỏi file table descriptor và trả về 0. Nếu không thì trả về -1.

Bước 4: Tăng thanh ghi PC

3. Cài đặt `int Read(char *buffer, int size, OpenFileID id)` và `int Write(char *buffer, int size, OpenFileID id)`

- Cài đặt syscall Read:

Đầu vào: buffer chứa dữ liệu, số ký tự, id của file

Đầu ra: lỗi thì trả về -1, thành công thì trả về số byte đọc được, file null thì trả về -2

Bước 1: Lấy địa chỉ của tham số buffer từ thanh ghi số 4, lấy số lượng ký tự từ thanh ghi số 5 và lấy id file từ thanh ghi số 6.

Bước 2: Kiểm tra tính hợp lệ của dữ liệu đầu vào (id của file có nằm ngoài bảng mô tả file hay không, file có tồn tại hay không, đọc file stdout). Nếu không hợp lệ thì trả về -1.

Bước 3: Nếu file là null thì trả về -2.

Bước 4: Xét nếu là trường hợp đọc file stdin thì dùng phương thức Read của lớp `SynchConsoleInput` để lấy số byte đọc được.

Bước 5: Xét nếu là trường hợp đọc file bình thường thì trả về số byte thực sự đọc được.

- Cài đặt syscall Write:

Đầu vào: buffer chứa dữ liệu, số ký tự, id của file

Bước 1: Lấy địa chỉ của tham số buffer từ thanh ghi số 4, lấy số lượng ký tự từ thanh ghi số 5 và lấy id file từ thanh ghi số 6.

Bước 2: Kiểm tra tính hợp lệ của dữ liệu đầu vào (id của file có nằm ngoài bảng mô tả file hay không, file có tồn tại hay không, đọc file only read). Nếu không hợp lệ thì trả về -1.

Bước 3: Nếu file là null thì trả về -2.

Bước 4: Xét nếu là trường hợp ghi file read & write thì trả về số byte thực sự ghi được.

Bước 5: Xét nếu là trường hợp đọc file stdout thì trả về -1 nếu chuỗi vượt quá 255, còn lại thì trả về số byte thực sự ghi được.

4. Cài đặt `int Seek(int position, OpenFileID id)`

Bước 1: Lấy các tham số từ thanh ghi 4 và 5.

Bước 2: Kiểm tra ngoại lệ và kiểm tra giới hạn bảng mô tả

Bước 3: Nếu position bằng -1 thì nhảy đến vị trí Length(), ngược lại nhảy đến vị trí đã nhập.
Bằng cách gọi kernel->fileSystem->openf[id]->Length() và
kernel->fileSystem->openf[id]->Seek(pos)

Bước 4: Tăng PC

5. Cài đặt int Remove(char* name)

Đầu vào là tên file. Đầu ra là -1 nếu không xóa được file, 0 nếu xóa được file.

Bước 1: Đọc tham số truyền vào từ thanh ghi 4

Bước 2: Chuyển tên file từ User Space sang System Space

Bước 3: Nếu không tồn tại file có tên file như vậy thì trả về -1

Bước 3: Nếu file đang được mở trong file table descriptor thì trả về -1

Bước 4: Xóa file và trả về 0

Bước 5: Tăng thanh ghi PC

II. Phần 2: Viết chương trình người dùng

1. Viết chương trình createfile

Dùng system call Create, đưa vào tên file để tạo file.

2. Viết chương trình copy, yêu cầu nhập file nguồn và file đích và thực hiện copy

Bước 1: Dùng system call PrintString, ReadString để đọc tên file nguồn và đích (filenameSrc, filenameDes)

Bước 2: Dùng system call Open, để mở 2 file (filenameSrc thì mở dạng chỉ đọc, filenameDes thì mở dạng đọc và ghi). Kiểm tra nếu không mở được file

Bước 3: Dùng system call Seek, để seek đến cuối file lấy độ dài của file nguồn

Bước 4: Dùng system call Read, Write để đọc từ file nguồn và ghi vào file đích cho tới khi đủ độ dài của file nguồn

Bước 5: Dùng system call Close để đóng file. Và dùng system call Halt để tắt máy.

3. Viết chương trình delete

Dùng system call Remove, đưa vào tên file để xóa file. System call Remove sẽ kiểm tra file có đang mở hoặc có hợp lệ hay không trước khi thao tác xóa.

4. Viết chương trình cat

Bước 1: Nhập tên file bằng hàm ReadString.

Bước 2: Mở file bằng hàm Open.

Bước 3: Kiểm tra mở file có thành công, nếu không thì báo lỗi và kết thúc.

Bước 4: Tìm file size bằng cách Seek với position = -1. Sau đó Seek lại 0 để quay về đầu.

Bước 5: Đọc từng ký tự và hiện ra màn hình, dùng PrintChar.

5. Viết chương trình concatenate

Bước 1: Mở các file thứ nhất và thứ hai, kiểm tra nếu có lỗi thì kết thúc. Dùng Open

Bước 2: Tạo file thứ 3, kiểm tra nếu có lỗi thì kết thúc. Dùng Create

Bước 3: Đọc các ký tự file 1 và ghi vào file 3, dùng Read

Bước 4: Đọc các ký tự file 2 và ghi vào file 3, dùng Read

Bước 5: Đóng các file, Dùng Close