ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN



HỆ ĐIỀU HÀNH

ĐỒ ÁN 1: SYSTEM CALL

*Giảng viên hướng dẫn:* **thầy Phạm Tuấn Sơn**

*Nhóm thực hiện*: **18120078 – Ngô Phù Hữu Đại Sơn**

**18120211 – Võ Thế Minh**

**18120227 – Phạm Văn Minh Phương**

*Lớp*: **HĐH 18\_4**

*Khóa*: **2018**

MỤC LỤC

[A. Thông tin khái quát 2](#_Toc55116783)

[I. Thông tin nhóm 2](#_Toc55116784)

[II. Bảng phân công công việc 2](#_Toc55116785)

[B. Nội dung 3](#_Toc55116786)

[I. Môi trường phát triển 3](#_Toc55116787)

[II. Mục tiêu của đồ án 3](#_Toc55116788)

[III. Ý tưởng thiết kế 3](#_Toc55116789)

[IV. Triển khai 4](#_Toc55116790)

[V. Kiểm thử 5](#_Toc55116791)

[C. TỔNG KẾT 6](#_Toc55116792)

[VI. Đánh giá đồ án 6](#_Toc55116793)

[1. Mức độ hoàn thành của các thành viên 6](#_Toc55116794)

[2. Mức độ hoàn thành đồ án: 6](#_Toc55116795)

[VII. Nguồn tham khảo 6](#_Toc55116796)

# Thông tin khái quát

## I. Thông tin nhóm

|  |  |  |
| --- | --- | --- |
| MSSV | HỌ TÊN | VAI TRÒ |
| 18120211 | Võ Thế Minh | Trưởng nhóm |
| 18120078 | Ngô Phù Hữu Đại Sơn | Thành viên |
| 18120227 | Phạm Văn Minh Phương | Thành viên |

## II. Bảng phân công công việc

|  |  |
| --- | --- |
| MSSV | CÔNG VIỆC PHỤ TRÁCH |
|  | Cài đặt nachOS |
|  | Tái cấu trúc lớp FileSystem |
|  | Tái cấu trúc lớp OpenFile |
|  | Cài đặt + kiểm thử system call Create |
|  | Cài đặt + kiểm thử system call Open |
|  | Cài đặt + kiểm thử system call Read |
|  | Cài đặt + kiểm thử system call Seek |
|  | Cài đặt + kiểm thử system call Write |
|  | Cài đặt + kiểm thử system call Close |
|  | Cài đặt + kiểm thử chương trình Echo |
|  | Cài đặt + kiểm thử chương trình Cat |
|  | Cài đặt + kiểm thử chương trình Copy |

# Nội dung

## Môi trường phát triển

* *Máy ảo*: nachOS 3.4
* *Hệ điều hành:* Ubuntu 14.04-i386
* *Compiler*: gcc - 3.4.6
* *Cross-compiler*: gcc - 2.95.3
* *Binutils*: 2.11.2

## Mục tiêu của đồ án

1. Tìm hiểu cách cài đặt hệ điều hành giả lập nashOS
2. Tìm hiểu cách thức giao tiếp giữa HĐH nashOS và chương trình người dùng
3. Tìm hiểu cách viết các system call cho HĐH nashOS.

## Ý tưởng thiết kế

* 1. **Chuyển đổi dữ liệu liệu giữa System space và User space**
  2. **Từ System space đến User space**

Sử dụng hàm WriteMem (có chức năng ghi 1, 2 hoặc 4 bytes từ System space sang User space) đã được định nghĩa sẵn của lớp Machine, xây dựng phương thức public System2User cho lớp Machine để chép dữ liệu từ 1 vùng nhớ ở System space sang User space

* 1. **Từ User space đến System space**

Sử dụng hàm ReadMem (có chức năng đọc 1, 2 hoặc 4 bytes từ User space sang System space) đã được định nghĩa sẵn của lớp Machine, xây dựng phương thức public User2System cho lớp Machine để chép dữ liệu từ 1 vùng nhớ ở User space sang System space

* 1. **Chén lớp SynchConsole và nashOS để đọc ghi từ thiết bị xuất nhật chuẩn**

HĐH nashOS không cung cấp cho ta các phương thức xuất nhập từ thiết bị xuất/nhập chuẩn nên ta cần phải chèn lớp SynchConsole vào nachos để xây dựng được các system call có sử dụng dữ liệu từ console. Đồng thời cũng dung để xuất các thông báo cho chương trình người dung.

* 1. **Tái cấu trúc lớp FileSystem**

HĐH nashOS ban đầu chỉ cung cấp cho ta lớp FileSystem quản lý việc nhập/xuất của 1 file tại 1 thời điểm. Vì vậy cần tái cấu trúc lại lớp này để có thể quản lý nhiều file hơn tại 1 thời điểm. Sử dụng 1 mảng để quản lý các file đang mở (quản lý tối đa 10 file trong cùng 1 thời điểm). 2 phần tử đầu tiên của mảng này dung để quản lý thiết bị xuất nhập chuẩn.

* 1. **Tái cấu trúc lớp OpenFile**

Lớp OpenFile của nachos mặc định khi mở file thì luôn có thể dùng để đọc và ghi. Ta cần dùng 1 thuộc tính type để phân biệt các file mở chỉ để đọc, có khả năng đọc và ghi, thiết bị nhập chuẩn, thiết bị xuất chuẩn.

* 1. **Xây dựng system call Create**
  2. **Xây dựng system call Open**
  3. **Xây dựng system call Close**
  4. **Xây dựng system call Read**
  5. **Xây dựng system call Write**
  6. **Xây dựng system call Seek**
  7. **Xây dựng chương trình echo**

Dựa vào các system call đã được xây dựng trước đó (Read, Write) để có thể lấy chuỗi người dùng nhập từ stdin và xuất chuỗi ra stdout.

* 1. **Xây dựng chương trình cat**

Dựa vào các system call đã được xây dựng trước đó (Open, Close, Read, Write, Seek) để đọc tên file từ stdin, mở và đóng file, tính kích thước nội dung file và xuất nội dung file ra stdout.

* 1. **Xây dựng chương trình copy**

Dựa vào các system call đã được xây dựng trước đó (Open, Close, Read, Write, Seek) để đọc tên file từ stdin, mở và đóng file, tính kích thước nội dung file nguồn và chép từng kí tự nội dung từ file nguồn qua file đích.

## Triển khai

* 1. **User2System**
  2. **System2User**
  3. **Tái cấu trúc FileSystem**
  4. **Tái cấu trúc OpenFile**
  5. **Xây dựng system call Create**
  6. **Xây dựng system call Open**
  7. **Xây dựng system call Close**
  8. **Xây dựng system call Read**
  9. **Xây dựng system call Write**
  10. **Xây dựng system call Seek**
  11. **Xây dựng chương trình echo**

Chương trình echo sẽ xuất lại chuỗi kí tự mà người dùng nhập từ console.

Gọi system call Read với 3 tham số là địa chỉ buffer, số kí tự tối đa và OpenFileID (Trong trường hợp này là stdin, được define là 0) để đọc chuỗi người dùng nhập từ console. Nếu hàm Read đọc được thành công thì nó sẽ trả về độ dài chuỗi thực sự đọc được vào biến length. Nếu length khác -1 và khác -2 (Trường hợp file lỗi hoặc EOF), tức là đọc được chuỗi từ stdin thành công, system call Write sẽ được gọi với 3 tham số là địa chỉ buffer, số kí tự của chuỗi đọc được và OpenFileID (Trong trường hợp này là stdout, được define là 1) để xuất chuỗi đọc được ra console.

* 1. **Xây dựng chương trình cat**

Chương trình cat sẽ hiển thị nội dung file với filename được nhập từ console.

Gọi system call Read để đọc tên file người dùng nhập từ console. System call Open được gọi để mở file vừa nhập, nếu mở file thành công thì sẽ dùng system call Seek để dịch con trỏ file về cuối file để lấy kích thước nội dung file và chuyển con trỏ file về đầu file để tiến hành đọc và xuất ra stdout từng kí tự thông qua system call Read và Write (Từ đầu đến hết kích thước file). Sau khi xuất hết nội dung file thì đóng file bằng system call Close. Nếu mở file không thành công sẽ thông báo cho người dùng.

* 1. **Xây dựng chương trình copy**

Chương trình copy nhận tên file nguồn và tên file đích để thực hiện copy nội dung từ file nguồn sang file đích.

Gọi system call Read 2 lần để đọc 2 tên file nguồn và đích. Tiếp theo gọi system call Open với type = 0 để mở file nguồn, nếu mở file nguồn thành công thì mở file đích (type = 1). OpenFileID của file nguồn và file đích phải khác nhau. Nếu mở cả 2 file thành công thì tiến hành dịch con trỏ đến cuối file nguồn bằng system call Seek để lấy kích thước nội dung file nguồn và sau đó dịch cả con trỏ file nguồn và file đích về đầu file rồi dùng system call Read để đọc từng kí tự file nguồn, Write để chép từng kí tự đọc được sang file đích cho đến khi hết nội dung file nguồn. Sau khi copy xong nội dung thì đóng cả 2 file bằng system call Close. Nếu ở bước tạo file đích xảy ra lỗi thì sẽ thông báo và đóng file nguồn.

## Kiểm thử

* 1. **Open, Read, Seek, Write, Close**
  2. **Nhập/xuất từ console**
  3. **Echo**
  4. **Cat**
  5. **Copy**

# TỔNG KẾT

## Đánh giá đồ án

### Mức độ hoàn thành của các thành viên

|  |  |  |
| --- | --- | --- |
| MSSV | Mức độ hoàn thành công việc | Đóng góp |
| 18120078 |  |  |
| 18120211 |  |  |
| 18120227 |  |  |

### Mức độ hoàn thành đồ án:

**Đánh giá**:

## Nguồn tham khảo