**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**

**Môn: Hệ điều hành**

**Lớp: 15CTT2**

**Sinh viên:**

* 1512239: Vũ Thừa Khang
* 1512241: Phan Nguyễn Đông Kha

Phân chia công việc:

|  |  |  |  |
| --- | --- | --- | --- |
| Họ tên | Mã số sinh viên | Công việc | Công việc chung |
| Vũ Thừa Khang | 1512239 | - Cài đặt centos 6  - Biên dịch nachos  - Thêm lớp **synconsole** và xây dựng 2 hàm **User2System** và **System2User**  - Xử lý các exceptions  - Viết lại cấu trúc điều khiển của chương trình để nhận các Nachos system calls  -Tăng program counter trước khi system call trả kết quả về  - Read / Print String  Bubble Sort | Tìm hiểu  Modun 1  và 2 |
| Phan Nguyễn Đông Kha | 1512241 | - Read / Print Int;  - Read / Print Char;  - Help;  - In bảng mã Ascii  Làm báo cáo. |

Modun 1+2:

Cài đặt hệ điều hành linux : centos 6 (final).

Biên dịch nachos và chạy thử chương trình halt.

Modun 3: Exception và System Call

1. Viết lại file exception.cc để xử lý tất cả các exceptions được liệt kê trong machine/machine.h.

Dùng cấu trúc switch case với 2 biến ***which*** và ***type*** để viết lại file. Trong đó, các lỗi đa số đều là không thể quay lại được, vì thế chỉ in kết quả ra console và gọi hàm halt(). Riêng 2 exception là :

* no exception :sẽ trả quyền điều khiển về HĐH.
* syscall exceptions sẽ được xử lý bởi các hàm chúng ta viết cho user system calls.

1. Tương tự câu a.
2. Tăng program counter.

Đoạn mã tăng giá trị biến program counter được lấy trong ***machine/mipssim.cc***, đặt vào cuối

***case SyscallException*** (trước **break**) để các syscall sau khi xử lý xong thì tăng giá trị program counter để tránh vòng lặp vô hạn. Điều chỉnh lại phần **register[NextPCReg] +=** **4** để phù hợp với chương trình

Giải thích về 2 phân b,c : lớp SynchConsole dùng để đọc một chuỗi ký tự vào trong buffer (chuỗi sẽ kết thúc khi người dùng nhấn enter, hoặc có chiều dài lớn hơn hoặc bằng giá trị length (>= length)). Buffer là vùng nhớ thuộc userspace, khi người dùng nhập chuỗi thì nội dung được lưu trữ ở kernel space nên cần viết một hàm tương ứng để chuyển dữ liệu từ kernelspcae qua userspace

1. Cài đặt system call :
2. Thêm lớp **synconsole** vào project như trong video hướng dẫn. Để sử dụng cho việc xây dựng các syscall sau này.
3. Xây dựng thủ tục copy vùng nhớ từ User Space vào System Space. Xây dựng 2 hàm **User2System** và **System2User** bằng cách copy khai báo hàm và định nghĩa của hàm vào 2 file **machine.h** và **machine.cc** trong thư mục machine. Để sử dụng cho các hàm PrintString và ReadString sau này.
4. Định nghĩa và khai báo hàm trong file **syscall.h.** Điều này để chương trình người dùng có thể gọi hàm đó.
5. Định nghĩa hàm trong file assembler start.c và start.s: Khi user gọi 1 lệnh đến hệ điều hành. Hệ điều hành sẽ nhận thông điệp đó, và lưu giá trị này vào. Ở đây giá trị sẽ được lưu vào thanh ghi *$2* và sau đó sẽ được tiến hành gọi các ***system call****.*
6. Tiến hành lập trình trong file **exception.cc** , tùy theo công việc của hàm mà làm một cách thích hợp.

- Mã của system call được đưa vào thanh ghi r2.

- Các biến người dùng sử dụng được đưa vào thanh ghi r4, r5, r6.

- Giá trị trả về của system call được đưa vào thanh ghi r2.

1. Ở thư mục **test** viết một chương trình kiểm tra (nếu cần). Sau đó thêm các chương trình sẽ được biên dịch vào file MakeFile. Cách thêm đã được hướng dẫn.
2. Chuyển đường dẫn làm việc trong terminal về thư mục code và nhập lệnh **gmake all** để biên dịch lại **nachos** lẫn chương trình trong thư mục **test**.

Mức độ hoàn thành đồ án: 100%

|  |  |  |
| --- | --- | --- |
| **Yêu cầu** | **Đánh giá** | **Ghi chú** |
| Xử lý các exceptions | hoàn thành |  |
| Viết lại cấu trúc điều khiển của chương trình để nhận các Nachos system calls | hoàn thành |  |
| Tăng program counter trước khi system call trả kết quả về | hoàn thành |  |
| Read / Print Int | hoàn thành |  |
| Read / Print Char | hoàn thành |  |
| Read / Print String | hoàn thành |  |
| Help | hoàn thành |  |
| In bảng mã Ascii | hoàn thành |  |
| Bubble Sort | hoàn thành |  |