Ho Chi Minh City University of Science

Faculty of Information Technology



**BÁO CÁO ĐỒ ÁN 1**

**MÔN HỆ ĐIỀU HÀNH**

*Sinh viên*: 1512203 – Nguyễn Quốc Huy

1512525 – Trần Quang Thắng

*Lớp*: 15CNTN

1. Các xử lý cho exceptions

Có tất cả 8 loại exception cần được xử lý được liệt kê trong file machine.h (trong thư mục nachos-3.4/code/machine/) như sau:

1. NoException
2. SyscallException
3. PageFaultException
4. ReadOnlyException
5. BusErrorException
6. AddressErrorException
7. OverflowException
8. IllegalInstrException

* Với các exception thuộc loại từ (c) đến (h), hệ điều hành hiển thị ra thông báo lỗi và ngắt hệ thống.
* Với exception (a), hệ thống trả quyền điều khiển lại cho hệ điều hành
* Với exception thuộc loại (b), việc xử lý sẽ được tiếp tục bởi các system call

1. Các hàm hỗ trợ cho việc cài đặt các system call

Để dễ dàng cho việc cài đặt, một số phương thức đã được thêm vào trong lớp machine (trong thư mục machine.h).

1. **Phương thức void Machine::AdjustPCRegs**

* Dùng để thay đổi giá trị thanh ghi PC sau khi thực hiện một system call, để không bị lặp vô tận khi gọi thực hiện system call.

1. **Phương thức char\* Machine::User2System(int virtAddr, int limit)**

* Dùng để chép tối đa limit bytes dữ liệu từ địa chỉ virAddr trong user space sang một bộ đệm của kernel space và trả về địa chỉ của bộ đệm.
* Phương thức này được sử dụng khi xuất một chuỗi ra console.

1. **Phương thức int Machine::System2User(int virtAddr, int len, char \*buffer)**

* Dùng để chép tối đa len bytes dữ liệu từ địa chỉ buffer (thuộc kernel space) sang user space và lưu vào địa chỉ virtAddr trong user space.
* Phương thức này được sử dụng khi nhập một chuỗi từ console (kernel space).

1. Các xử lý cho system calls

Các system call được cài đặt trong đồ án lần này (sau mỗi system call đều gọi hàm AdjustPCRegs đã nêu ở phần 2):

1. **System call void PrintChar(char character)**

* System call này sử dụng lớp SynchConsole để xuất một ký tự ra màn hình.
* Chi tiết:
  + System call PrintChar nhận vào một ký tự, sau đó thông qua lớp SynchConsole để ghi ký tự này lên console.

1. **System call char ReadChar()**

* System call này sử dụng lớp SynchConsole để đọc một ký tự do người dùng nhập vào.
* Chi tiết:
  + System call ReadChar khong nhận tham số, mà đọc một byte (1 ký tự) từ console qua lớp SynchConsole, và trả về ký tự đó.

1. **System call void PrintString(char[] buffer)**

* System call này sử dụng lớp SynchConsole để in chuỗi ký tự từ trong buffer ra màn hình.
* Chi tiết:
  + System call PrintString đưa vào địa chỉ của chuỗi muốn xuất ra console (ở user space).
  + Để in chuỗi ra màn hình, cần một bước trung gian là đưa chuỗi qua bộ đệm trong kernel space, sau đó mới sử dụng lớp SynchConsole để in chuỗi ra console.
  + Tuy nhiên, do chuỗi nằm trong user space, và không cho trước độ dài chuỗi trong danh sách tham số, nên không thể xác định trước kích thước của bộ đệm.
  + Giải pháp đưa ra: chia dữ liệu thành từng block, mỗi block có 32768 bytes (32 kilobytes), và xử lý trên từng block. Từng block dữ liệu (ở user space) được chuyển qua kernel buffer (thông qua phương thức User2System đã nêu ở phần 2), tiền xử lý (như trên một chuỗi bình thường) rồi mới đưa dữ liệu từ kernel buffer sang console qua lớp SynchConsole.
  + Do sử dụng phương pháp trên nên số ký tự tối đa của chuỗi để có thể xuất ra màn hình (trong một lần gọi lệnh) là 220.

1. **System call void ReadString(char[] buffer, int length)**

* System call này sử dụng lớp SynchConsole để đọc một chuỗi ký tự vào buffer (chuỗi sẽ kết thúc khi người dùng ấn enter hoặc có chiều dài lớn hơn hoặc bằng giá trị length).
* Chi tiết:
  + System call ReadString đưa vào địa chỉ của chuỗi (ở user space) và chiều dài tối đa của chuỗi.
  + Đầu tiên, một bộ đệm trên kernel space để chuyển chuỗi từ console về.
  + Cũng tương tự như việc nhập chuỗi, do địa chỉ chuỗi đưa vào là ở user space, nên ta cần đưa dữ liệu từ kernel space về user space thông qua phương thức System2User đã nêu ở phần 2.

1. **System call void PrintInt(int number)**

* System call này sử dụng lớp SynchConsole để xuất một số nguyên ra màn hình.
* Chi tiết:
  + System call PrintInt nhận một số nguyên number từ user space. Số nguyên được chuyển sang chuỗi ký tự (trong kernel space), sau đó được đưa lên console qua lớp SynchConsole.
  + Các cài đặt chi tiết của việc chuyển số sang chuỗi có thể xem trong mã nguồn và các comment trong mã nguồn.

1. **System call int ReadInt()**

* System call này sử dụng lớp SynchConsole để đọc một số nguyên do người dùng nhập vào. Nếu giá trị người dùng nhập không phải là số nguyên thì trả về 0.
* Chi tiết:
  + System call ReadInt không nhận tham số đầu vào.
  + Khi thực hiện system call này, đầu tiên toàn bộ các ký tự phân cách bao gồm ký tự ‘\0’, ‘\t’, ‘\n’ được loại ra khỏi bộ đệm.
  + Nếu ký tự đầu tiên khác ký tự phân cách không phải là một chữ số (‘0’, …, ‘9’) hoặc dấu ‘-‘ thì sẽ trả về 0.
  + Ngược lại, liên tục đọc tiếp cho đến khi gặp ký tự khác ký tự số thì dừng.
  + Mỗi ký tự đọc được được thêm vào kết quả (kiểu số nguyên).
  + Hai trường hợp số âm và số không âm được xử lý riêng (dựa vào dấu ‘-‘).
  + Nếu kết quả tràn số thì cũng trả về 0.

1. Các chương trình khác
2. **Chương trình ascii**: có mã nguồn được lưu tại nachos-3.4/code/test/ascii.c

Chương trình in ra bảng mã ascii (chỉ gồm các ký tự có mã ascii từ 0 tới 127), sử dụng system call PrintChar đã viết ở trên.

1. **Chương trình sort**: có mã nguồn được lưu tại nachos-3.4/code/test/sort.c

Chương trình cho phép người dùng nhập vào một số nguyên n (n không vượt quá 100), tiếp đó là nhập n số nguyên, sau đó sử dụng thuật toán bubble sort để sắp xếp lại mảng và xuất mảng ra màn hình (việc nhập xuất sử dụng các system ReadInt và PrintInt đã viết ở trên).

1. **Chương trình help**: có mã nguồn lưu tại nachos-3.4/code/test/help.c

Chương trình in ra màn hình các dòng giới thiệu cơ bản về nhóm và mô tả vắn tắt hai chương trình sort và ascii (sử dụng system call PrintString đã viết ở trên).

1. Mã nguồn

Chi tiết cài đặt được đính kèm cùng report file và được lưu trữ online tại <https://github.com/LawrenceXavier/HDH_Project/>

1. Tự đánh giá

Độ hoàn thành đồ án của nhóm là 100%.