**VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY**

**UNIVERSITY OF SCIENCE**

**FACULTY OF INFORMATION TECHNOLOGY**

Logo

Description automatically generated

**PROJECT 1 REPORT**

**SUBJECT: OPERATING SYSTEM**

**EXCEPTIONS & NACHOS SYSTEM CALLS**

**Class: 20CLC09**

**Theory Lecturer**

Mr. Le Giang Thanh

**Lab Lecturers**

Mr. Le Ha Minh

Mr. Nguyen Thanh Quan

**Students**

Le Tan Dat 20127011

Dang Tien Dat 20127458

Pham Thi Anh Phat 20127680

(2021 - 2022)

1. **INTRODUCTION**

Giới thiệu đồ án tại dungf linux nachos, điểm mạng nachos, …. Chém gió.

1. **REQUIREMENT**
2. **Xử lí exceptions:** hoàn thành các system call được yêu cầu trong file exception.cc (SC\_PrintNum, SC\_PrintString,…). Viết lại file exception.cc để xử lí tất cả các exceptions được liệt kê trong *machine/machine.h*.
3. **Tăng giá trị PC:** trong mã MIPS, sau một chỉ thị lệnh chúng ta phải tăng giá trị PC lên 4 byte (void countValuePC();) nếu không chúng ta sẽ bị một vòng lặp vô hạn gọi thực hiện system call này mãi mãi.
4. **int ReadNum();**

-Comment: System call này có chức năng cho người dùng nhập vào một số và syscall này sẽ sử dụng lớp synchConsoleIn để đọc số nguyên do người dùng nhập vào đó.

-Cách cài đặt (ý tưởng):

+Sao chép các kí tự nhập vào một kiểu chuỗi và sau đó sẽ copy các kí tự số đó vào một buffer.

+Kiểm tra các kí tự trong buffer có phải là số không, có chứa các kí tự đặt biệt hay không

+Nếu thỏa các kí tự đó là số chúng ta bắt đầu chuyển kiểu string về kiểu int

+Và trả về số chuyển đổi được.

1. **void PrintNum(int number);**

-Comment: System call này có chức năng là in ra một số được truyền vào. System call này sử dung synchConsoleOut để xuất số nguyên ra màn hình

-Cách cài đặt (ý tưởng):

+Sử dụng lớp synchConsoleOut->PutChar() để xuất kí tự ra màn hình.

+Kiểm tra đối số truyền vào (number): nếu là số 0 thì put kí tự ‘0’ ra màn hình, nếu nó là số âm thì chúng ta put thêm kí tự ‘-‘ phía trước số đó và chuyển về kiểu kí tự để xuất.

1. **char ReadChar();**

-Comment: System call này có chức năng cho người dùng nhập vào một kí tự và syscall này sẽ sử dụng lớp synchConsoleIn để đọc kí tự do người dùng nhập vào đó.

-Cách cài đặt (ý tưởng):

+Sử dụng lớp kernel->synchConsoleIn->GetChar(); để yêu cầu người dùng nhập vào kì tự.

1. **void PrintChar(char character);**

-Comment: system call này có chức năng in một kí tự ra màn hình console và system call này sử dụng lớp synchConsoleIn để in kí tự đó ra màn hình.

-Cách cài đặt (ý tưởng):

+Sử dụng lớp kernel->synchConsoleOut->PutChar(); để xuất kí tự đó ra màn hình.

1. **int RandomNum();**

-Comment: system call này có chức năng ramdom một số.

-Các cài đặt (ý tưởng):

+Sử dụng thư viện ramdom() (return ramdom()) để ramdon một số.

1. **void ReadString (char \*buffer, int length)**

-Comment: system call này dùng để đọc một một chuỗi string được người dùng nhập từ bàn phím.

-Cách cài đặt (ý tưởng):

+Vì khi người dùng nhập chuỗi thì nội dung được lưu trữ ở kernel space thì chúng ta sử dụng hàm để cuyển đổi dữ liệu từ kernel space sang user space.

1. **void PrintString (char \*buffer);**

-Comment: system call này dùng để in một chuỗi ra màn hình .

-Cách cài đặt (ý tưởng):

+Duyệt hết chuỗi và sử dụng tương tự như PrintChar để xuất từng kí tự ra màn hình.

1. **OpenFileId Open (char \*name);**

-Comment: system call này dùng để mở file.

-Cách cài đặt (ý tưởng):

+Sử dụng lớp kernel->fileSystem->Open(fileName, type) để mở file lên. Đối số truyền vào fileName là tên file cần mở và type là đối số chế độ ‘open to read’ hay ‘open to write’ hay ‘open to read and write’.

1. **int Close(OpenFileId id);**

-Comment: system call này có chức năng dùng để đóng file đã được mở.

-Cách cài dặt (ý tưởng):

+Sử dụng kernel->fileSystem->Close(id) để đóng file, trả về 0 nếu đóng thành công trả về 1 nếu đóng không thành công

1. **int Read(char \*buffer, int size, OpenFileId id);**

-Comment: system call này dùng để đọc file với id cho trước

1. **int Write(char \*buffer, int size, OpenFileId id);**

-Comment: system call này dùng để ghi file với id cho trước.

1. **int Seek(int position, OpenFileId);**

-Comment: system call này dùng để chuyển vị trí con trỏ tới vị trí position trong đối số truyền vào thích hợp trong file id, nếu position = -1 thì chuyển đến cuối file và trả về vị trí thực nếu thành công và trả về -1 nếu bị lỗi.

1. **int Remove(char \*name);**

-Comment: system call này dung để xóa file thư mục có tên name trong đối số truyền vào và sử dụng FileSystem object để xóa file.

* ***Viết các system calls liên quan đến các thao tác như print number, print char, read number, read char…***

Step 1: Create file demo.c in test folder to build the demo program.

Step 2: In Makefile, add name program in line PROGRAMS to run the program (demo) and add following code to expand file.o and file.coff:

Graphical user interface, text

Description automatically generated

Step 3: In file code/test/start.S add MIPS code for the function to be run so the computer knows there is a program to declare this system call for use.

Step 4: In file code/userprog/syscall.h add line that define value passed to system calls and prototype for the function.

Step 5: In file code/userprog/ksyscall.h to define the function.

Step 6: In file code/userprog/exception.cc xử lí các system call trong hàm exceptionHandler, định nghĩa các hàm handle\_SC có nhiệm vụ là xử lí đọc cái địa chỉ của thanh ghi và ghi lại kết quả vào địa chỉ.

Step 7: Viết chương trình người dung trong file demo.c và biên dịch chương trình với lệnh sau:

~nachos/NachOS-4.0/code/build.linux$ make && make depend

~nachos/NachOS-4.0/code/test: make && make depend

Sau khi biên dịch chương trình ta có các file demo.coff demo.o và demo

~nachos/NachOS-4.0/code/test: ../build.linux/nachos -x demo

Note:

* Because the system can not use the value address directly from user, so we need to define user2System function that transforms the value from userspace to kernelspace and contrast - system2User.
* Không để cho user có thể làm sap HĐH …
* ***Các system calls liên quan đến việc thao tác trên file như create file, delete, open, close, seek, …***

Step 1: Get the parameter of the function.

Step 2:

* ***Viết các chương trình người dùng như help, sort, createfile, …***

Step 1:

1. **ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Task** | **Description** | **Done** |
| 1 |  | Understand about NachOS | x |
| 2 |  | Understand the design | x |
| 3 | 1 | Exceptions handler | x |
|  | 2 | Program Counter | x |
|  | 3 | Read Num | x |
|  | 4 | Print Num | x |
|  | 5 | Read Char | x |
|  | 6 | Print Char | x |
|  | 7 | Random Num | x |
|  | 8 | Read String | x |
|  | 9 | Print String | x |
|  | 10, 14 | Create, Remove file | x |
|  | 11 | Open, Close file | x |
|  | 12 | Read file | x |
|  | 13 | Seek in file | x |
|  | 15, 16, 17 | Help, ascii, sort | x |
|  | 18, 19, 20, 21 | Create file, cat, copy, delete | x |
|  |  | Avoid crashing the OS | x |
| 4 |  | Report | x |

1. **REFERENCES**

[1]. <https://courses.ctda.hcmus.edu.vn/mod/folder/view.php?id=48702>

[2]. [https://github.com/leduythuccs/nachos-project/](https://github.com/leduythuccs/nachos-project/blob/master/code/network/post.cc)

[3]. <https://www.youtube.com/playlist?list=PLRgTVtca98hUgCN2_2vzsAAXPiTFbvHpO>