**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

Logo

Description automatically generated

**BÁO CÁO**

**Project 2: Systemcall Nachos**

Sinh viên: Bùi Hữu Thịnh, Lê Quốc Thịnh, Trần Đình Tiến Anh, Nguyễn Thành Luân, Nguyễn Phùng Mai Đan

Lớp: 19\_5

Giảng viên phụ trách: Lê Viết Long

**MỤC LỤC**

[**A.** **Thông tin cá nhân** 3](#_Toc104062329)

[**B.** **Phân công và đánh giá kết quả** 3](#_Toc104062330)

[**C.** **Chi tiết** 3](#_Toc104062331)

[**D.** **Demo chương trình** 5](#_Toc104062332)

[**E.** **Tài liệu tham khảo** 9](#_Toc104062333)

1. **Thông tin cá nhân**

* Bùi Hữu Thịnh – 18600257.
* Lê Quốc Thịnh – 18600260.
* Trần Đình Tiến Anh – 19120167.
* Nguyễn Thành Luân – 19120285.
* Nguyễn Phùng Mai Đan – 19120466.

1. **Phân công và đánh giá kết quả**

|  |  |  |
| --- | --- | --- |
| **Họ tên** | **Công việc** | **Đánh giá** |
| Bùi Hữu Thịnh | Tìm tài liệu, làm báo cáo | 50% |
| Lê Quốc Thịnh | Tìm tài liệu, làm báo cáo | 50% |
| Trần Đình Tiến Anh | Không tham gia | 0% |
| Nguyễn Thành Luân | Nghiên cứu, cài đặt, làm báo cáo | 100% |
| Nguyễn Phùng Mai Đan | Nghiên cứu, cài đặt. làm báo cáo | 100% |

1. **Chi tiết**
   * 1. **Viết lại file exception.cc**

* Thay đổi cấu trúc file thành switch case để phù hợp với từng trường hợp của các exception nhất định. Đồng thời, việc sử dụng switch case cũng giúp code dễ nhìn và dễ quản lý hơn.
  + 1. **Viết lại cấu trúc điều khiển của chương trình để nhận các Nachos System call**
* Sử dụng cấu trúc switch case cho từng system call, với mỗi system call tách ra thành một hàm riêng biệt để dễ xử lý hơn, sau khi thực hiện hàm system call thì gọi hàm tăng program counter lên, hàm tăng program counter được thực hiện ở câu c.
  + 1. **Viết mã để tăng giá trị biến program counter**
* Đoạn mã gốc để tăng program counter nằm trong machine/mipssim.cc, thay đổi lại đoạn mã này cho phù hợp với file exception và tách ra thành một hàm mới lên là IncreaseProgramCounter.
  + 1. **Cài đặt system call int ReadInt()**
* Thực hiện cài đặt system call theo hướng dẫn trong file [4].
* Nhập một số nguyên 4 bytes (có giá trị trong khoảng từ -2,147,483,648 đến 2,147,483,647) từ màn hình console bằng hàm Read của lớp SynchConsole.
* Chương trình kiểm tra bao gồm việc nhập giá trị số bất kì, nếu hợp lệ thì xuất giá trị này ra (có sử dụng hàm PrintInt và PrintString ở câu e và i để có giao diện dễ sử dụng hơn).
  + 1. **Cài đặt system call void PrintInt(int number)**
* Thực hiện cài đặt system call theo hướng dẫn trong file [4].
* Xuất một số nguyên 4 bytes ra màn hình console bằng hàm Write của lớp SynchConsole.
* Chương trình kiểm tra sẽ xuất ra giá trị 100000.
  + 1. **Cài đặt system call char ReadChar()**
* Thực hiện cài đặt system call theo hướng dẫn trong file [4].
* Nhập một kí tự từ màn hình console bằng hàm Read của lớp SynchConsole.
* Chương trình kiểm tra bao gồm việc nhập giá trị số bất kì, nếu hợp lệ thì xuất giá trị này ra (có sử dụng hàm PrintChar và PrintString ở câu g và i để có giao diện dễ sử dụng hơn).
  + 1. **Cài đặt system call void PrintChar(char character)**
* Thực hiện cài đặt system call theo hướng dẫn trong file [4].
* Xuất một kí tự ra màn hình console bằng hàm Write của lớp SynchConsole.
* Chương trình kiểm tra sẽ xuất ra kí tự ‘L’.
  + 1. **Cài đặt system call void ReadString(char[] buffer, int length)**
* Thực hiện cài đặt system call theo hướng dẫn trong file [4].
* Nhập một kí tự từ màn hình console bằng hàm Read của lớp SynchConsole kết hợp với hàm User2System để chuyển chuyển dữ liệu từ kernelspace qua userspace.
* Chương trình kiểm tra bao gồm việc nhập chuỗi kí tự bất kì, sau đó xuất giá trị này ra (có sử dụng hàm PrintString ở câu i để có giao diện dễ sử dụng hơn).
  + 1. **Cài đặt system call void PrintString(char[] buffer)**
* Thực hiện cài đặt system call theo hướng dẫn trong file [4].
* Xuất một chuỗi kí tự ra màn hình console bằng hàm Write của lớp SynchConsole kết hợp với hàm System2User để chuyển chuyển dữ liệu từ userspace qua kernelspace.
* Chương trình kiểm tra sẽ xuất ra chuỗi kí tự ‘Nguyen Thanh Luan’.
  + 1. **Viết chương trình help**
* Chương trình help sẽ xuất thông tin về các thành viên trong nhóm (họ tên, mã số sinh viên) và thông tin (tên chương trình, input, output) về hai chương trình ascii và sort.
  + 1. **Viết chương trình ascii**
* Chương trình ascii sẽ in ra bảng mã ascii từ 0 đến 127.
  + 1. **Viết chương trình sort**
* Chương trình sort sẽ nhận một mảng số nguyên gồm n phần tử nhập từ người dùng (n<=100), sau đó sắp xếp mảng bằng giải thuật bubble sort và in mảng sau khi sắp xếp ra màn hình Console.

1. **Demo chương trình**
   * 1. **ReadInt**

Text

Description automatically generated

* + 1. **PrintInt**

Text

Description automatically generated

* + 1. **ReadChar**

Text

Description automatically generated

* + 1. **PrintChar**

Text

Description automatically generated

* + 1. **ReadString**

Text

Description automatically generated

* + 1. **PrintString**

Text

Description automatically generated

* + 1. **Help**

Text

Description automatically generated

* + 1. **Ascii**

Text

Description automatically generated

* + 1. **Sort**

Text

Description automatically generated

1. **Tài liệu tham khảo**

* [Nachos-Programing-HCMUS](https://github.com/nguyenthanhchungfit/Nachos-Programing-HCMUS).
* [Nachos\_os\_projects\_HCMUS](https://github.com/quanghuy1258/nachos_os_projects_HCMUS).