

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



BÁO CÁO PROJECT 1
HỆ THỐNG QUẢN LÝ TẬP TIN TRÊN WINDOWS

Giảng viên hướng dẫn: Lê Viết Long

Lớp: 21_3

Nhóm sinh viên thực hiện:

Trần Minh Triết	21120578
Lê Anh Tú	21120585
Nguyễn Minh Tuấn	21120587
Nguyễn Thị Như Ý	21120600
Trần Phước Nhân	21120515

Thành phố Hồ Chí Minh, tháng 10 - 2023

MỤC LỤC

1. Thông tin nhóm.....	3
2. Bảng phân chia công việc	4
3. Đánh giá mức độ hoàn thành đồ án	5
4. Mô tả các bước thực hiện	6
4.1. Tìm hiểu về Nachos:	6
4.1.1. Tổng quan về Nachos:	6
4.1.2. Thiết kế:	6
4.1.3. Quá trình biên dịch một chương trình:	7
4.1.4. Thực thi chương trình:	7
4.1.5. Viết một System call mới và cài được chương trình để kiểm tra:	7
4.2. Exception:.....	8
4.2.1. Các Exception:	8
4.2.2. Viết mã để tăng giá trị biến Program Counter:	8
4.3. Các System calls:	8
4.3.1. ReadInt và PrintInt:.....	8
4.3.2. ReadChar và PrintChar:	10
4.3.3. ReadString và PrintString:	12
4.4. Các chương trình:	15
4.4.1. Chương trình help:	15
4.4.2. Chương trình ascii:.....	16
4.4.3. Chương trình sort:	16
5. Tài liệu tham khảo và trích dẫn:	19

1. Thông tin nhóm

STT	Họ và tên	MSSV	Địa chỉ email
1	Trần Minh Triết	21120578	21120578@student.hcmus.edu.vn
2	Lê Anh Tú	21120585	21120575@student.hcmus.edu.vn
3	Nguyễn Minh Tuấn	21120587	21120587@student.hcmus.edu.vn
4	Nguyễn Thị Như Ý	21120600	21120600@student.hcmus.edu.vn
5	Trần Phước Nhân	21120515	21120515@student.hcmus.edu.vn

2. Bảng phân chia công việc

STT	Yêu cầu	Đảm nhiệm	Đánh giá
1	Hiểu mã chương trình nachos Hiểu thiết kế	21120587	Hoàn thành
2	Viết Exceptions	21120600	Hoàn thành
3	Viết lại cấu trúc điều khiển của chương trình để nhận các Nachos system calls	21120600	Hoàn thành
4	Viết mã để tăng giá trị biến program counter	21120600	Hoàn thành
5	Cài đặt system call int ReadInt()	21120578	Hoàn thành
6	Cài đặt system call void PrintInt(int number)	21120578	Hoàn thành
7	Cài đặt system call char ReadChar()	21120515	Hoàn thành
8	Cài đặt system call void PrintChar(char character)	21120515	Hoàn thành
9	Cài đặt đặt system call void ReadString(char[] buffer, int length)	21120515	Hoàn thành
10	Cài đặt đặt system call void PrintString(char[] buffer),	21120515	Hoàn thành
11	Viết chương trình help	21120585	Hoàn thành
12	Viết chương trình ascii để in ra bảng mã ascii	21120585	Hoàn thành
13	Viết chương trình sort	21120578 21120587	Hoàn thành
14	Viết Report	21120578 21120587 21120515 21120600 21120585	Hoàn thành

3. Đánh giá mức độ hoàn thành đồ án

STT	Yêu cầu	Mức độ Hoàn thành	Ghi chú
1	Hiểu mã chương trình nachos Hiểu thiết kế	100%	
2	Viết Exceptions	100%	
3	Viết lại cấu trúc điều khiển của chương trình để nhận các Nachos system calls	100%	
4	Viết mã để tăng giá trị biến program counter	100%	
5	Cài đặt system call int ReadInt()	100%	
6	Cài đặt system call void PrintInt(int number)	100%	
7	Cài đặt system call char ReadChar()	100%	
8	Cài đặt system call void PrintChar(char character)	100%	
9	Cài đặt đặt system call void ReadString (char[] buffer, int length)	100%	
10	Cài đặt đặt system call void PrintString (char[] buffer),	100%	
11	Viết chương trình help	100%	
12	Viết chương trình ascii để in ra bảng mã ascii	100%	
13	Viết chương trình sort	100%	
14	Viết Report		
	Toàn bộ đồ án	100%	

4. Mô tả các bước thực hiện

4.1. Tìm hiểu về Nachos:

4.1.1. Tổng quan về Nachos:

NachOS 3.4 chạy trên môi trường Linux với kiến trúc x86. Về cơ bản, NachOS chỉ là một phần mềm trên Linux, ta cần dùng GCC để biên dịch chương trình này. Quá trình biên dịch NachOS (thứ tự, các liên kết...) được quy định bởi file Makefile trong thư mục code.

Compiler GCC của Linux được sử dụng để biên dịch toàn bộ mã nguồn của NachOS, tạo ra file thực thi NachOS trong thư mục **./userprog** để sử dụng.

4.1.2. Thiết kế:

- **Chương trình ứng dụng** (./test): do người lập trình ứng dụng viết.

Trong thư mục **test** có một file tên MakeFile – file chịu trách nhiệm quy định quá trình **GCC 2.95.3** biên dịch các file .c thành các file .coff rồi được chuyển thành file .noff để thực thi nhờ chương trình coff2noff (./bin).

Cross compiler **GCC 2.95.3** chỉ chịu trách nhiệm biên dịch cho thư mục ./test, còn các thư mục còn lại thì do compiler GCC của Linux chịu trách nhiệm (được cài đặt thông qua command `sudo apt-get install gcc`).

- **Máy ảo MIPS** (./machine): để giả lập các thành phần CPU, RAM, Register, I/O, ...

Trong mã nguồn của file **machine**, có một lớp **Machine** có một số phương thức quan trọng như:

- +**Run()**: chạy chương trình người dùng.

- +**ReadRegister(int number)**: đọc nội dung của thanh ghi number.

- +**WriteRegister(int num, int value)**: ghi giá trị value vào thanh ghi number.

- **Tương tác với màn hình Console** (./thread): Lớp SynchConsole được cài đặt trong các file synchcons.h và synchcons.cpp. Lớp SynchConsole cung cấp 2 phương thức là Write và Read.

- +**Write**: ghi vào thiết bị I/O một lượng numBytes của buffer from, trả về số byte ghi được.

- +**Read**: đọc vào buffer into từ thiết bị I/O một lượng numBytes, trả về số byte đọc được.

- Ngoài ra còn nhiều thư mục khác (**./userprog**, **./thread**, ...): cũng là một phần con của hệ thống NachOS, cấu thành từ một vài file mã nguồn, có nhiệm vụ xử lý các system call.

4.1.3. Quá trình biên dịch một chương trình:

Ví dụ, một chương trình trong thư mục **test** là **halt.c**, quá trình biên dịch xảy ra qua các giai đoạn như sau:

- Cross Compiler sẽ biên dịch chương trình **halt.c** thành tập tin **halt.s** dựa trên mã hợp ngữ chạy trên kiến trúc MIPS.
- File **halt.s** sẽ được liên kết với file **start.s** để tạo thành file **halt.coff** – là định dạng thực thi trên hệ điều hành Linux cho kiến trúc MIPS.
- Tập tin **halt.coff** được chuyển thành tập tin **halt.noff** bằng cách sử dụng chương trình **coff2noff** được cung cấp trong thư mục **./bin/**.

4.1.4. Thực thi chương trình:

Tiếp tục thực thi chương trình **halt** (halt là một system call dùng để tắt NachOS). Sau khi truyền đường dẫn đến thư mục **code** bằng Terminal, để thực thi chương trình, ta nhập command:

./userprog/nachos -rs 1023 -x ./test/halt.

Trong đó:

- ./userprog/nachos**: mở NachOS trong thư mục userprog.
- rs 1023 -x**: các tham số cần thiết cho quá trình thực thi.
- ./test/halt**: tên chương trình cần thực thi trong thư mục test.

4.1.5. Viết một System call mới và cài đặt chương trình để kiểm tra:

Để tạo một System call mới, ta thực hiện lần lượt các bước sau:

- Khai báo system call mới trong file **./userprog/syscall.h**, bao gồm khai báo mã code và prototype cho system call mới.
- Thêm vào file **./test/start.c** và **start.s** để system call có thể được biên dịch thành mã thực thi trên máy ảo MIPS. Lưu ý: Dùng thanh ghi r2 để chứa mã code của system call và chứa giá trị trả về của system call đó. Các tham số khác lần lượt được chứa trong thanh ghi r4 đến r7.
- Xử lý các system call mới trong file **./userprog/exception.cc**.
- Viết chương trình kiểm tra trong thư mục **./test**.
- Sửa lại **Makefile** bên trong thư mục test để biên dịch chương trình .c vừa tạo.
- Biên dịch lại NachOS (dùng command make trong Terminal)
- Kiểm tra chương trình với command như ở phần “Thực thi chương trình”, thay **halt** thành tên chương trình bạn muốn.

4.2. Exception:

4.2.1. Các Exception:

Đọc file code/machine/machine.h ta sẽ có danh sách các exception đã được liệt kê sẵn. Với mỗi một exception đã được liệt kê, thực hiện viết lại các case để xử lý cho các exception bắt được trong quá trình thực hiện chạy chương trình. Mỗi exception được tìm thấy sẽ thực hiện công việc hiển thị thông báo lỗi đó lên màn hình, sau đó gọi hàm halt() để tắt hệ điều hành.

Đối với trường hợp là Syscall Exception, tiến hành kiểm tra loại(type) Syscall được nạp vào sau đó thực hiện các bước xử lý bên trong syscall để trả về kết quả theo đúng với yêu cầu của từng chức năng.

4.2.2. Viết mã để tăng giá trị biến Program Counter:

Viết mã tăng Program Counter để thực hiện nạp lệnh tiếp theo để thực hiện. Trong hàm IncreasePC ta sẽ thực hiện lưu giá trị của register hiện tại cho register trước, register tiếp theo cho register hiện tại và register tiếp nữa cho register kế.

4.3. Các System calls:

4.3.1. ReadInt và PrintInt:

○ ReadInt():

Syscall này đọc một chuỗi số nguyên từ màn hình, xử lý chuỗi số và ghi vào thanh ghi được chỉ định. Việc xử lý bao gồm xử lý dấu (âm dương), ký tự không hợp lệ và đầu vào trống. Cụ thể:

Đầu tiên ta lấy buffer của chuỗi số nguyên được đọc dưới dạng mảng ký tự. Sử dụng hàm Read(buffer, size) trong class gSynchConsole với size = 256. Ta nạp được chuỗi số vào buffer đồng thời lưu độ dài của chuỗi số vào biến bytesRead. Cuối cùng gắn ký tự kết thúc chuỗi '\0' vào cuối buffer.

Tiếp theo là xử lý chuỗi đầu vào. Kiểm tra xem có dữ liệu đầu vào không, sau đó xử lý đầu vào từng ký tự một. Xử lý dấu (âm hoặc dương) và xây dựng giá trị số nguyên từ các chữ số. Nếu chỉ có một ký tự đọc được và nó là 'a' thì coi là không hợp lệ.

Nếu đầu vào hợp lệ, nhân kết quả với dấu và ghi vào thanh ghi số 2. Còn trường hợp không hợp lệ thì in ra thông báo lỗi và ghi 0 vào thanh ghi số 2.

Cuối cùng giải phóng vùng nhớ đã cấp phát cho buffer và tăng PC.

○ PrintInt():

Syscall này in một số nguyên lên console, bao gồm xử lý dấu, tính chiều dài và chuyển đổi số nguyên thành ký tự chuỗi để in. Cụ thể:

Đầu tiên đọc một số nguyên từ thanh ghi số 4 của máy ảo và lưu vào biến number.

Cấp phát vùng nhớ cho một mảng ký tự buffer để lưu số nguyên.

Tiếp theo xử lý chuỗi số nguyên được lưu trong buffer:

Nếu số number bằng 0, nó sẽ in ra màn hình chuỗi "0".

Ngược lại, nếu number là một số âm, nó đảo dấu số và in dấu trừ "-".

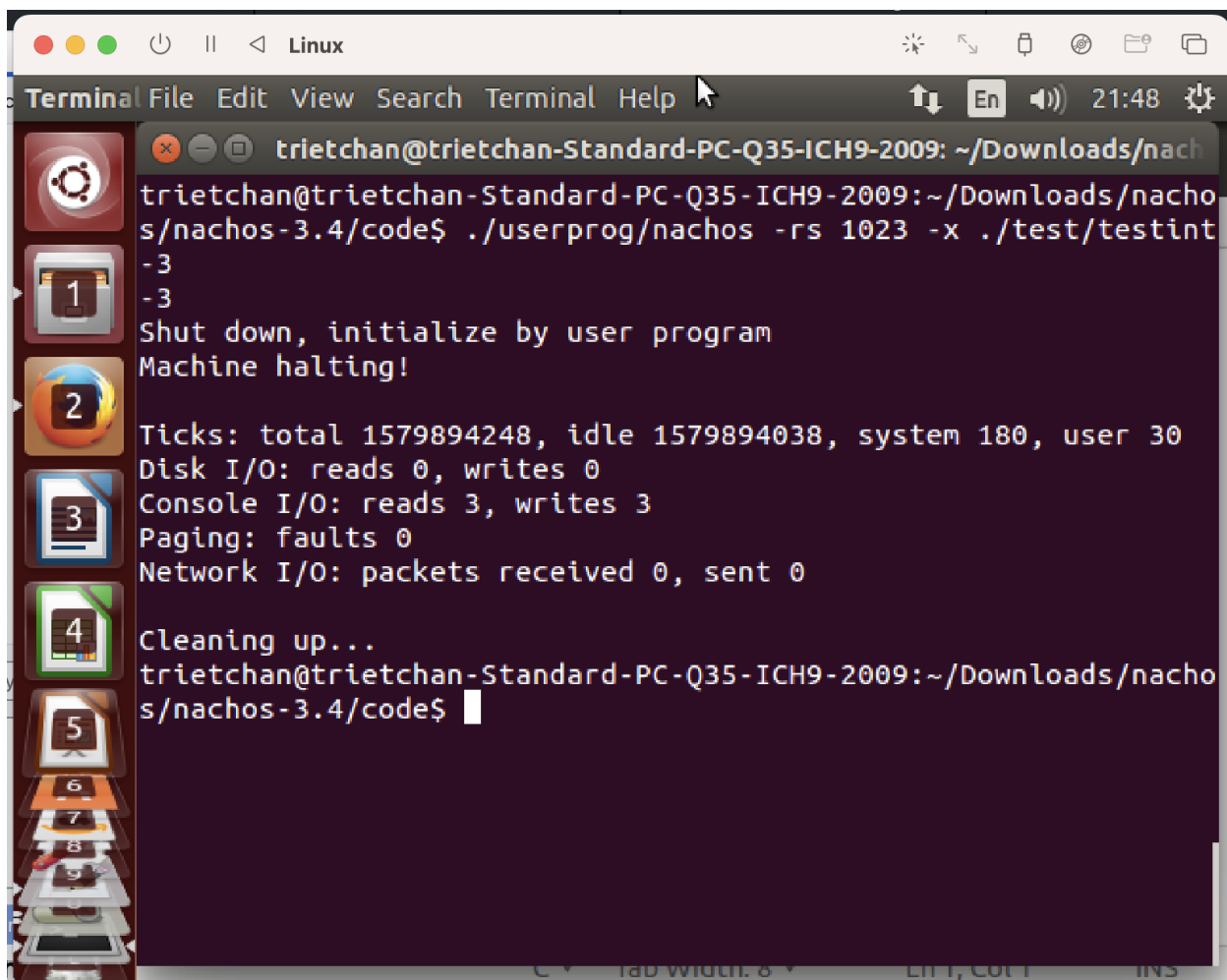
Tiếp theo, tính chiều dài của số nguyên bằng cách đếm số chữ số.

Sau đó, sử dụng vòng lặp để xây dựng chuỗi buffer từ số nguyên, chuyển đổi từng chữ số thành ký tự và lưu vào bộ đệm theo thứ tự đảo ngược.

Thêm ký tự kết thúc chuỗi null và sử dụng gSynchConsole->Write để in chuỗi lên màn hình.

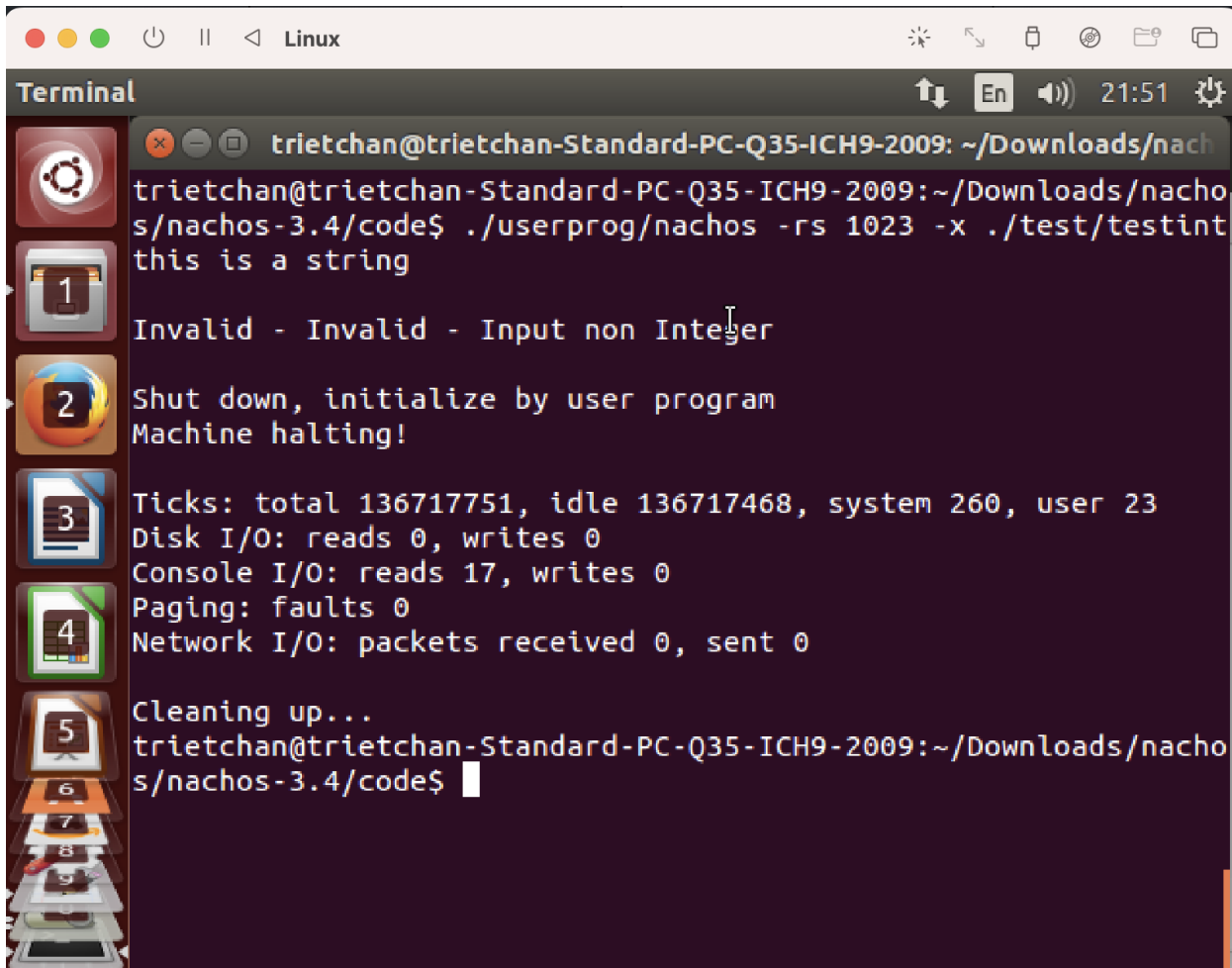
Cuối cùng giải phóng vùng nhớ cho buffer và tăng PC.

Trường hợp nhập đúng số nguyên, chương trình đọc và xuất ra số nguyên vừa được nhập



```
Linux
Terminal File Edit View Search Terminal Help
trietchan@trietchan-Standard-PC-Q35-ICH9-2009: ~/Downloads/nachos
trietchan@trietchan-Standard-PC-Q35-ICH9-2009:~/Downloads/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/testint
-3
-3
Shut down, initialize by user program
Machine halting!
Ticks: total 1579894248, idle 1579894038, system 180, user 30
Disk I/O: reads 0, writes 0
Console I/O: reads 3, writes 3
Paging: faults 0
Network I/O: packets received 0, sent 0
Cleaning up...
trietchan@trietchan-Standard-PC-Q35-ICH9-2009:~/Downloads/nachos/nachos-3.4/code$
```

Trường hợp đầu vào không phải số nguyên, báo lỗi "Invalid - Input non Integer"



```
Linux
Terminal
trietchan@trietchan-Standard-PC-Q35-ICH9-2009: ~/Downloads/nachos
trietchan@trietchan-Standard-PC-Q35-ICH9-2009:~/Downloads/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/testint
this is a string
Invalid - Invalid - Input non Integer
Shut down, initialize by user program
Machine halting!
Ticks: total 136717751, idle 136717468, system 260, user 23
Disk I/O: reads 0, writes 0
Console I/O: reads 17, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0
Cleaning up...
trietchan@trietchan-Standard-PC-Q35-ICH9-2009:~/Downloads/nachos/nachos-3.4/code$
```

4.3.2. ReadChar và PrintChar:

Syscall ReadChar: là syscall dùng để thực hiện đọc vào 1 ký tự từ màn hình console bằng cách sử dụng lớp SynchConsole để đọc ký tự tự do do người dùng nhập vào.

“SC_ReadChar” là 1 exception được định nghĩa trong file exception.cc. Thể hiện của nó chính là hàm “char ReadChar()”

Input: người dùng nhập từ màn hình console

Output: Nếu người dùng nhập 1 ký tự bất kỳ đúng thì sẽ trả về ký tự đó. 0 nếu người dùng nhập chuỗi, rỗng.

Mục đích: đọc 1 ký tự riêng biệt từ màn hình console và trả về kết quả cho thanh ghi số 2.

Luồng logic:

- Khi người dùng nhập vào màn hình console và nhấn enter.
- Hàm dùng SynchConsole tiến hành đọc dữ liệu vào 1 mảng buffer

- Tiến hành kiểm tra số byte thực tế đọc được sau khi hàm Read của SynchConsole thực hiện xong.
 - + Nếu như số byte đọc thực tế là 0 hoặc hơn 1. Tức input không hợp lệ thông báo cho người dùng và tắt chương trình
 - + Nếu số byte đọc thực tế đúng bằng 1 thì đó chính là dữ liệu hợp lệ tiến hành ghi ký tự đọc được vào thanh ghi số 2 chính là thanh ghi lưu trữ giá trị trả về.

Syscall PrintChar: là syscall dùng để thực hiện ghi ra màn hình Console bằng cách sử dụng lớp SynchConsole để ghi ra từ trực truyền thông qua tham số kiểu char.

“SC_PrintChar” là 1 exception được định nghĩa trong file userprog/exception.cc. Thể hiện của nó chính là hàm “void PrintChar(char value)”.

Input: Nhận vào 1 tham số kiểu char

Output: Tiến hành in ra ký tự được truyền vào thông qua tham số.

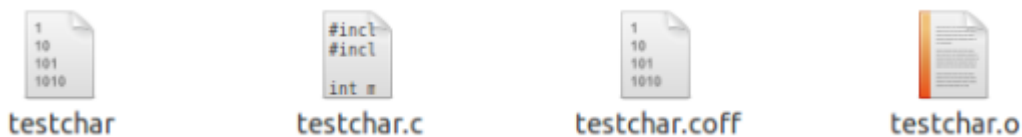
Mục đích: Kết hợp với hàm ReadChar để có thể ghi ra màn hình ký tự người dùng nhập trên màn hình Console.

Luồng logic:

- Tham số được nhận vào dưới dạng ký tự char, được truyền khi kết hợp với hàm ReadChar
- Tiến hành đọc dữ liệu từ tham số bằng hàm ReadRegister của machine từ thanh ghi tham số 4.
- Tiến hành kiểm in ra ký tự được truyền vào ra màn hình Console cho người dùng.

Demo chương trình:

- Hàm “char ReadChar(;)” và “void PrintChar(char value)” được kiểm nghiệm bằng cách xây dựng 1 chương trình nhỏ có tên là testchar.c nằm trong ./code/test



- Để tiến hành thử nghiệm file này chúng ta sử dụng mở terminal trong Ubuntu. Và di chuyển đến đường dẫn ./code

```
allen@allen-virtual-machine:~$ cd Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code/
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$
```

- Tại đây tiến hành nhập lệnh để chạy chương trình
- **./userprog/nachos -rs 1023 -x ./test/testchar**

```
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/testchar
```

- Kết quả:

+ Nhập đúng

```
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/testchar
A
A
Shut down, initialize by user program
Machine halting!

Ticks: total 397461389, idle 397461248, system 110, user 31
Disk I/O: reads 0, writes 0
Console I/O: reads 2, writes 1
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$
```

+ Nhập sai: nhập chuỗi

```
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/testchar
hello

Invalid - Number of characters is more than 1

Shut down, initialize by user program
Machine halting!

Ticks: total 357864251, idle 357864108, system 120, user 23
Disk I/O: reads 0, writes 0
Console I/O: reads 6, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$
```

+ Nhập sai: nhập rỗng

```
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/testchar

Invalid - Input character is empty

Shut down, initialize by user program
Machine halting!

Ticks: total 95306851, idle 95306758, system 70, user 23
Disk I/O: reads 0, writes 0
Console I/O: reads 1, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$
```

4.3.3. ReadString và PrintString:

Syscall ReadString: là syscall dùng để thực hiện đọc vào 1 chuỗi ký tự từ màn hình console bằng cách sử dụng lớp `SynchConsole` để đọc ký tự tự do do người dùng nhập vào.

“`SC_ReadString`” là 1 exception được định nghĩa trong file `userprog/exception.cc`. Thể hiện của nó chính là hàm “`void ReadString(char* buffer, int length)`”

Input: người dùng nhập từ màn hình console chuỗi các ký tự.

Output: Dữ liệu được lấy từ màn hình Console và lưu vào mảng buffer. Nếu người dùng nhập vào là rỗng (enter) thì sẽ thông báo lỗi cho người dùng. Nếu người dùng nhập đúng thì tiến hành chuyển đổi dữ liệu từ system space sang user space và lưu vào biến buffer.

Mục đích: đọc 1 chuỗi ký tự riêng biệt từ màn hình console và lưu lại cho tham số chuỗi buffer.

Luồng logic:

- Khi người dùng nhập vào màn hình console và nhấn enter.

- Hàm sẽ dùng SynchConsole tiến hành đọc dữ liệu vào 1 mảng buffer
- Tiến hành kiểm tra số byte thực tế đọc được sau khi hàm Read của SynchConsole thực hiện xong.
 - + Nếu như số byte đọc thực tế là 0. Tức input không hợp lệ thông báo cho người dùng và tắt chương trình
 - + Nếu số byte đọc thực tế lớn hơn bằng 1 thì đó chính là dữ liệu hợp lệ tiến hành chuyển đổi mảng ký tự đọc được từ system space sang user space và lưu vào tham số mảng buffer truyền vào.

Syscall PrintString: là syscall dùng để thực hiện ghi ra màn hình Console bằng cách sử dụng lớp SynchConsole để ghi ra tứ tự trực truyền thông qua tham số kiểu mảng ký tự char.

“SC_PrintString” là 1 exception được định nghĩa trong file exception.cc. Thể hiện của nó chính là hàm “void PrintString(char* buffer)”.

Input: Nhận vào 1 tham số kiểu con trỏ, trỏ đến mảng dữ liệu dạng char.

Output: Tiến hành dữ liệu từ tham số mảng char truyền vào.

Mục đích: Kết hợp với hàm ReadString để có thể ghi ra màn hình chuỗi ký tự người dùng nhập trên màn hình Console.

Luồng logic:

- Tham số được nhận vào dưới dạng chuỗi ký tự char, được truyền khi kết hợp với hàm ReadString
- Tiến hành đọc dữ liệu từ tham số bằng hàm ReadRegister của machine từ thanh ghi tham số 4.
- Tiến hành kiểm in ra ký tự được truyền vào ra màn hình Console cho người dùng. Bằng cách sử dụng SynchConsole

Demo chương trình:

- Hàm “void ReadString(char* buffer, int length);” và “void PrintString(char* buffer)” được kiểm nghiệm bằng cách xây dựng 1 chương trình nhỏ có tên là teststring.c nằm trong ./code/test



teststring



teststring.c



teststring.coff

- Để tiến hành thử nghiệm file này chúng ta sử dụng mở terminal trong Ubuntu. Và di chuyển đến đường dẫn ./code

```
allen@allen-virtual-machine:~$ cd Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code/
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$
```

- Tại đây tiến hành nhập lệnh để chạy chương trình
./userprog/nachos -rs 1023 -x ./test/teststring

```
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/teststring
```

- Kết quả:
 - + Nhập đúng:

```
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/teststring
Hello World
Hello World
Shut down, initialize by user program
Machine halting!

Ticks: total 415127492, idle 415127115, system 350, user 27
Disk I/O: reads 0, writes 0
Console I/O: reads 12, writes 12
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$
```

- + Nhập sai: rỗng khi chỉ nhập phím enter

```
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/teststring

Invalid - The string is empty -

Shut down, initialize by user program
Machine halting!

Ticks: total 151297282, idle 151297155, system 100, user 27
Disk I/O: reads 0, writes 0
Console I/O: reads 1, writes 1
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
allen@allen-virtual-machine:~/Downloads/hdh/Project2-HDH-main/nachos/nachos-3.4/code$
```

4.4. Các chương trình:

4.4.1. Chương trình help:

Gọi system call `PrintString(char[])` để in ra giới thiệu cơ bản về nhóm và mô tả vắn tắt về chương trình sort và ascii.

```
tule@tule-VirtualBox:~/Desktop/Project2/Project2-HDH/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/testhelp

=====HE DIEU HANH- 2023=====
THONG TIN NHOM
21120515 - TRAN PHUOC NHAN
21120600 - NGUYEN THI NHU Y
21120578 - TRAN MINH TRIET
21120585 - LE ANH TU
21120587 - NGUYEN MINH TUAN

=====

=====IN MA ASCII=====
Bang ma ASCII se co ky tu in duoc va khong
Trong bang ma co 1 ky tu dang tien chuong khong duoc hien thi
Chuong trinh sap xep la testascii.c nam trong thu muc test
Dat terminal tai ../nachos/nachos-3.4/code
Dung lenh: ./userprog/nachos -rs 1023 -x ./test/testascii

=====
```

```
=====

=====SORT=====
Thuat toan Bubble Sort
Nhap vao so luong phan tu n
Nhap cac phan tu trong mang can sap xep
Chuong trinh sap xep la testsort.c nam trong thu muc test
Dat terminal tai ../nachos/nachos-3.4/code
Dung lenh: ./userprog/nachos -rs 1023 -x ./test/testsort

=====

Shut down, initialize by user program
Machine halting!

Ticks: total 100923, idle 90200, system 10550, user 173
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 902
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
```

4.4.2. Chương trình ascii:

Chương trình in ra bảng mã ascii bằng cách duyệt qua từng giá trị trên bảng, với mỗi giá trị được duyệt tới, gọi system call PrintChar(char value) để in ra giá trị dưới dạng ký tự ascii tương ứng.

```
tule@tule-VirtualBox:~/Desktop/Project2/Project2-HDH/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/testascii

-----In bang ma ASCII-----
Cot 0  00 01 02 03 04 05

Cot 1  06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
Cot 2  ! " # $ % & ' ( ) * + , - . /
Cot 3  0 1 2 3 4 5 6 7 8 9 : ; < = > ?
Cot 4  @ A B C D E F G H I J K L M N O
Cot 5  P Q R S T U V W X Y Z [ \ ] ^ _
Cot 6  ` a b c d e f g h i j k l m n o
Cot 7  p q r s t u v w x y z { | } ~ 7F

-----

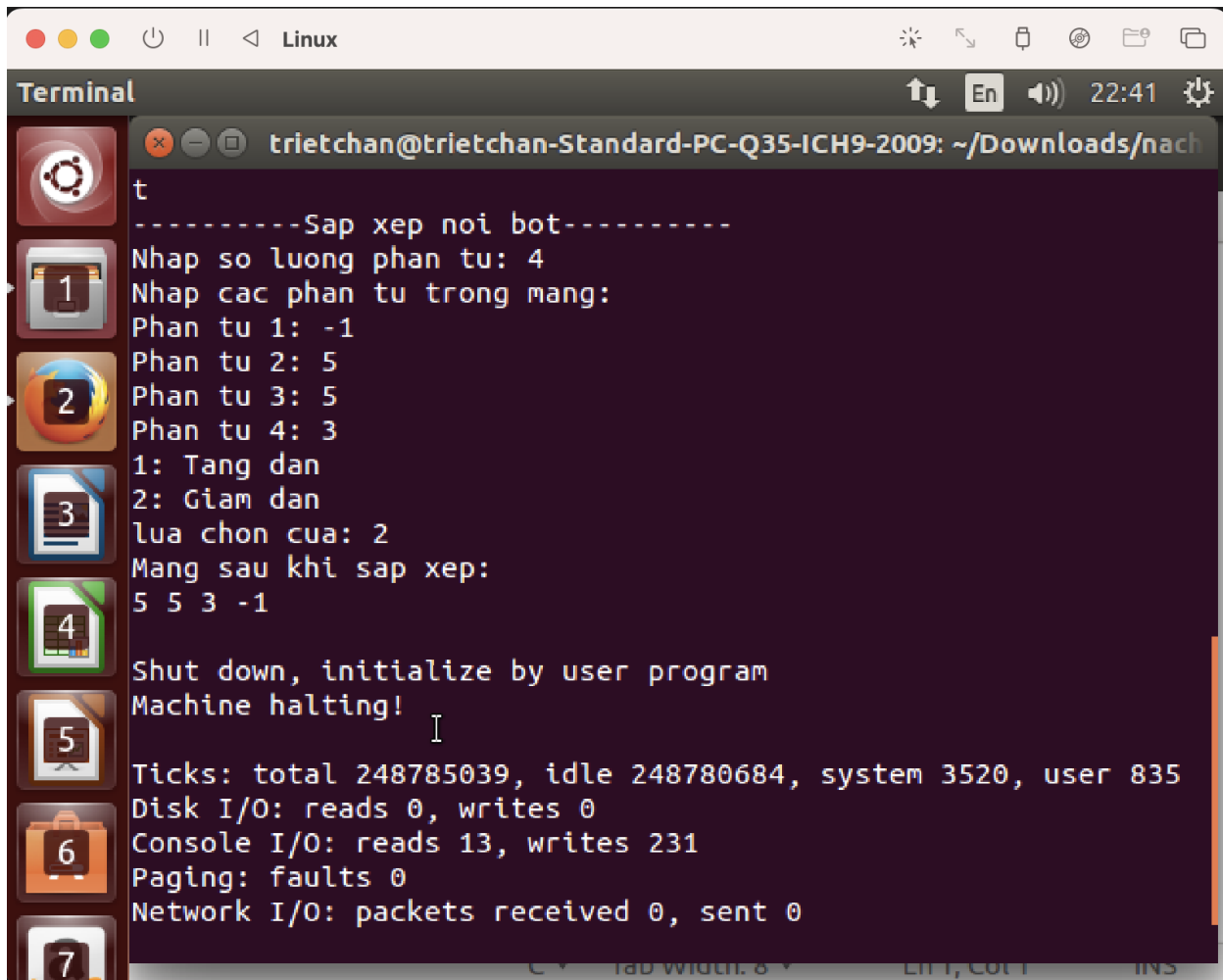
Shut down, initialize by user program
Machine halting!

Ticks: total 89865, idle 70500, system 14730, user 4635
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 705
```

4.4.3. Chương trình sort:

Sử dụng thuật toán bubble sort để sắp xếp tăng dần hay giảm dần theo ý muốn người dùng. Người dùng được yêu cầu nhập số lượng phần tử của mảng cần sắp xếp và giá trị cho từng phần tử của mảng bằng syscall ReadInt(). Sau đó người dùng được yêu cầu chọn cách sắp xếp: phím 1 là tăng dần và phím 2 là giảm dần. Sau khi đã chọn cách sắp xếp, syscall xử lý mảng bằng thuật toán bubble sort. Cuối cùng xuất ra màn hình mảng đã được sắp xếp bằng các syscall PrintInt() và PrintString().

Trường hợp sắp xếp giảm dần:



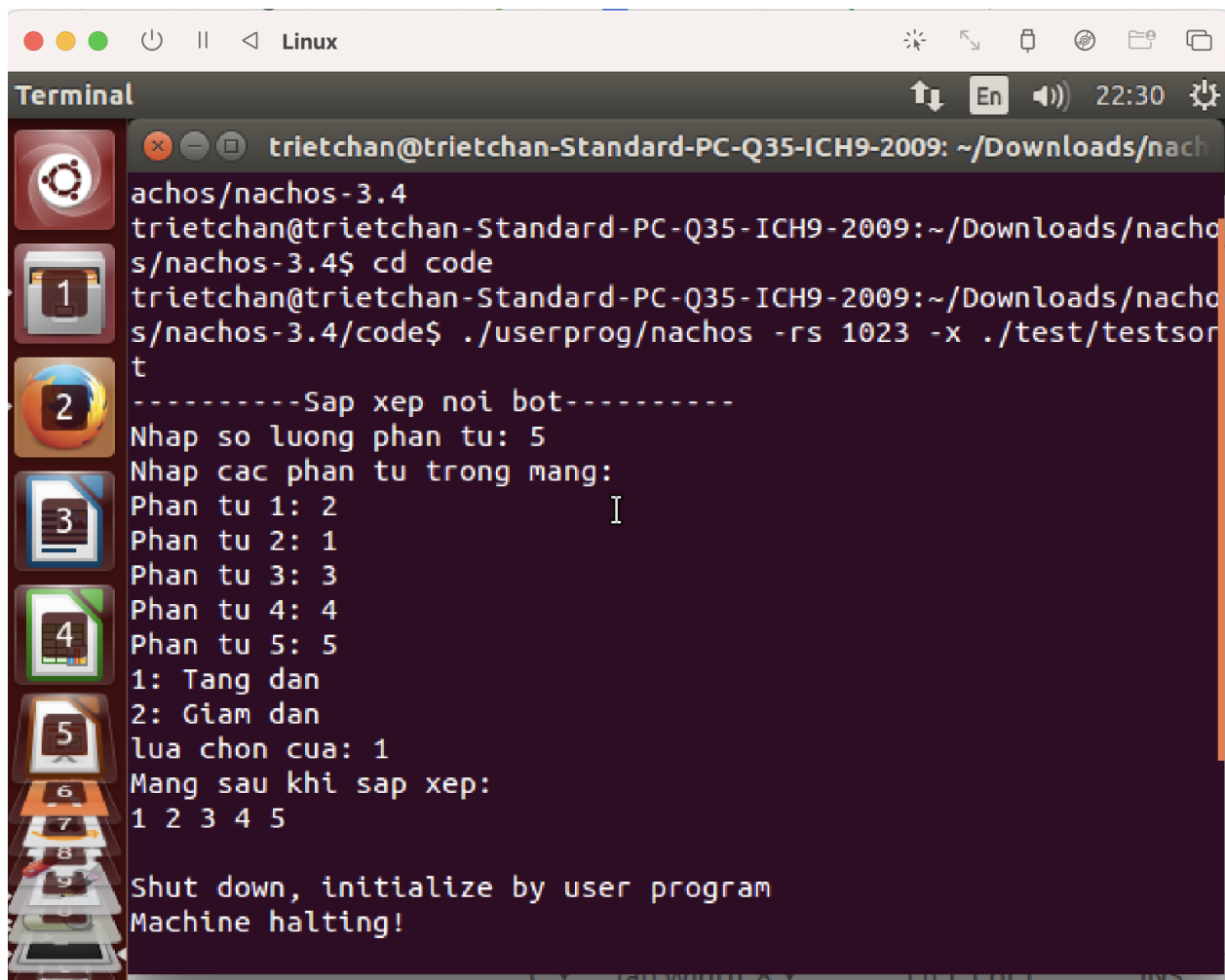
The screenshot shows a Linux terminal window titled "Terminal" with a dark background. The prompt is `trietchan@trietchan-Standard-PC-Q35-ICH9-2009: ~/Downloads/nach`. The user has entered `t`. The program output is in Vietnamese and shows the steps of a sorting algorithm. It asks for the number of elements (4) and the elements themselves (-1, 5, 5, 3). It then asks for the sorting order (1: Tang dan, 2: Giảm dần) and the user selects 2. The output shows the sorted array: 5 5 3 -1. Finally, it displays system statistics: Ticks, Disk I/O, Console I/O, Paging, and Network I/O.

```
t
-----Sap xep noi bot-----
Nhap so luong phan tu: 4
Nhap cac phan tu trong mang:
Phan tu 1: -1
Phan tu 2: 5
Phan tu 3: 5
Phan tu 4: 3
1: Tang dan
2: Giảm dần
lua chon cua: 2
Mang sau khi sap xep:
5 5 3 -1

Shut down, initialize by user program
Machine halting!

Ticks: total 248785039, idle 248780684, system 3520, user 835
Disk I/O: reads 0, writes 0
Console I/O: reads 13, writes 231
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Trường hợp sắp xếp tăng dần:



```
Linux
Terminal
trietchan@trietchan-Standard-PC-Q35-ICH9-2009: ~/Downloads/nachos/nachos-3.4
trietchan@trietchan-Standard-PC-Q35-ICH9-2009:~/Downloads/nachos/nachos-3.4$ cd code
trietchan@trietchan-Standard-PC-Q35-ICH9-2009:~/Downloads/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/testsort
-----Sap xep noi bot-----
Nhap so luong phan tu: 5
Nhap cac phan tu trong mang:
Phan tu 1: 2
Phan tu 2: 1
Phan tu 3: 3
Phan tu 4: 4
Phan tu 5: 5
1: Tang dan
2: Giam dan
lua chon cua: 1
Mang sau khi sap xep:
1 2 3 4 5

Shut down, initialize by user program
Machine halting!
```

5. Tài liệu tham khảo và trích dẫn:

[1] ThS. Lê Viết Long, (2023). Slide và tài liệu lập trình Nachos

[2] Nguyễn Thành Chung, (30-November-2023), Lập trình NaChos HCMUS, <https://www.youtube.com/@thanhchungnguyen2618/playlists>

[3] Trường CNTT & TT, (2023), Hệ điều hành, <https://tailieuhust.com/tai-lieu-mon-he-dieu-hanh-hust/>