



Trường: ĐH Khoa học tự nhiên TP.HCM

Khoa: Công nghệ thông tin

Lớp: TH2014/03

# **BẢO CẢO ĐỒ ÁN 1**

## **NACHOS**

**Môn: Hệ điều hành**

**Lớp TH2014/3**

**GVLT: Vũ Minh Trí.**

**GVTH: Trương Phước Hưng**

**Sinh viên thực hiện:**

- |                           |          |                |
|---------------------------|----------|----------------|
| <b>1. Trần Thị Nhã</b>    | <b>-</b> | <b>1412363</b> |
| <b>2. Đoàn Hiếu Tâm</b>   | <b>-</b> | <b>1412477</b> |
| <b>3. Nguyễn Đình Sơn</b> | <b>-</b> | <b>1412465</b> |

## MỤC LỤC

<b>I. PHÂN CÔNG CÔNG VIỆC VÀ ĐÁNH GIÁ.....</b>	<b>3</b>
<b>II. BÁO CÁO KẾT QUẢ ĐỒ ÁN.....</b>	<b>3</b>
<b>1. Sơ lược về NachOS.....</b>	<b>3</b>
1.1 Mô hình giao tiếp chung giữa hệ điều hành và người dùng.....	3
1.2 Giao tiếp giữa hệ điều hành nachos và chương trình người dùng. ....	4
<b>2. Cài đặt các xử lý cho Exceptions và system calls.....</b>	<b>4</b>
a) Xử lý exceptions: .....	4
b) Xử lý System calls: .....	5
c) Xử lý tăng program counter: .....	5
<b>3. Cài đặt system call.....</b>	<b>6</b>
a) Mô tả cài đặt ReadInt: .....	7
b) Mô tả cài đặt PrintInt: .....	7
c) Mô tả cài đặt ReadChar:.....	7
d) Mô tả cài đặt void PrintChar(char character): .....	8
e) Mô tả cài đặt void ReadString(char[]buffer, int lenght): .....	8
f) Mô tả cài đặt void PrintString (char[] buffer): .....	8
<b>4. Viết chương trình: .....</b>	<b>9</b>
a) Chương trình help: .....	9
b) Chương trình ascii: .....	9
c) Chương trình sort: .....	9
<b>5. Các hình ảnh minh họa kết quả: .....</b>	<b>10</b>
<b>III. Tài liệu tham khảo:.....</b>	<b>17</b>

## I. PHÂN CÔNG CÔNG VIỆC VÀ ĐÁNH GIÁ

Công việc thực hiện	Người thực hiện	Mức độ hoàn thành	Đánh giá của nhóm
Tìm hiểu và cài đặt các xử lý cho Exceptions và System calls	Trần Thị Nhã	100%	100%
Cài đặt system call	Đoàn Hiếu Tâm	100%	100%
Viết chương trình	Nguyễn Đình Sơn	100%	100%

## II. BÁO CÁO KẾT QUẢ ĐỒ ÁN

### 1. Sơ lược về NachOS.

#### 1.1 Mô hình giao tiếp chung giữa hệ điều hành và người dùng.

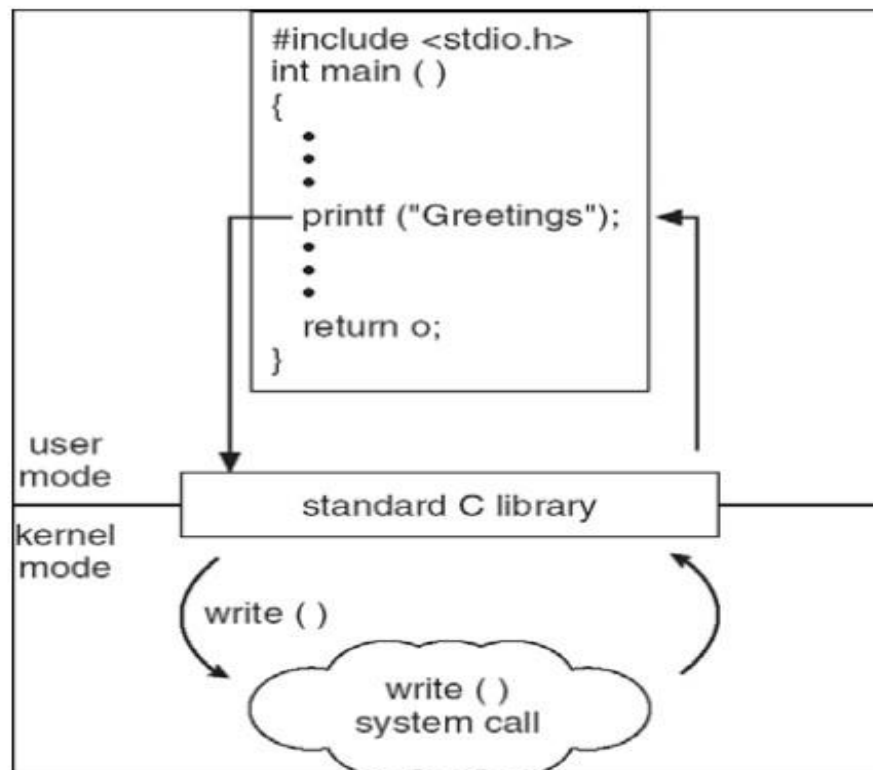


Figure 1: Mô hình giao tiếp giữa HĐH nachos và người dùng.

## 1.2 Giao tiếp giữa hệ điều hành nachos và chương trình người dùng.

Xét chương trình **x.c** được viết để chạy trên hệ điều hành nachos. Để chương trình này chạy được ta phải biên dịch nó. Quá trình biên dịch gồm 3 giai đoạn:

1. **X.c** được biên dịch bởi cross-compiler gcc thành tập tin **x.s** là mã hợp ngữ chạy trên kiến trúc MIPS.
2. Tập tin **x.s** này sẽ được liên kết với tập tin **start.s** để tạo thành tập tin **x.coff**, là định dạng thực thi trên hệ điều hành Linux cho kiến trúc máy MIPS.
3. Tập tin **x.coff** được chuyển thành tập tin **halt.noff** (hình dưới), là dạng thực thi trên hệ điều hành nachos cho kiến trúc máy MIPS, sử dụng tiện ích "**coff2noff**" được cung cấp sẵn trong thư mục code/bin.

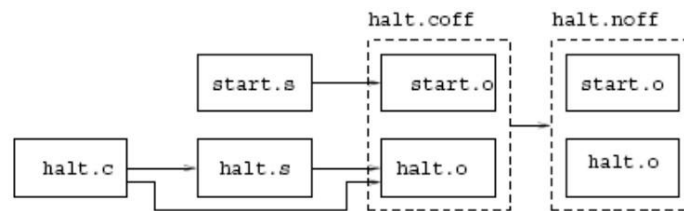


Figure 2: Mô hình giao tiếp giữa hệ điều hành nachos và người dùng, lấy halt làm ví dụ

## 2. Cài đặt các xử lý cho Exceptions và system calls.

### a) Xử lý exceptions:

- Viết lại file exception.cc để xử lý tất cả các exceptions được liên kết trong machine/machine.h. Hầu hết các exception trong này là run-time errors, khi các exception này xảy ra thì user program không thể phục hồi. Trường hợp đặc biệt duy nhất là **no exception** sẽ trả quyền điều khiển về cho hệ điều hành, còn syscalls exception sẽ được xử lý bởi các hàm chúng ta biết cho user system calls. Với tất cả các exception khác,

hệ điều hành hiển thị ra một thông báo lỗi và halt hệ thống.

- Với hàm `ExceptionHandler(ExceptionType which)`, dùng `Switch(which)` để xử lý các exception (`which`), mỗi case tương ứng với mỗi exception (dùng break sau mỗi case để tránh trôi sang case khác).

## b) Xử lý System calls:

- Cài đặt cross-compiler, thêm thư mục "gnu-decstation-ultrix" vào thư mục hdh, đổi tên "gnu-decstation-ultrix" thành "cross-compiler" sau đó vào trong thư mục cross-compiler/decstation-ultrix/ đổi tên thư mục "2.95.3" thành thư mục bin.
- Thêm User2System và System2User . Công dụng: để chuyển đổi vùng nhớ từ User mode sang Kernel Mode và ngược lại. Hai hàm này sẽ trở thành phương thức public trong lớp Machine. Khai báo hàm nằm ở **machine/machine.h** và phần định nghĩa sẽ nằm ở **machine/machine.cc**.
- Viết lại cấu trúc điều khiển của chương trình để nhận các nachos system calls. Kiểm tra cấu trúc mới bằng cách dùng system call **Halt** để kiểm tra tính đúng đắn của cấu trúc mới  
Các system call sẽ được xử lý trong case `SyscallException` bằng `switch(type)`. Mỗi case của `Switch(type)` tương ứng với một system call. Thêm `interrupt-> Halt()` ở cuối cùng case `SyscallException` (trước break) để kiểm tra tính đúng đắn.

## c) Xử lý tăng program counter:

- Tất cả các system call (không phải halt) sẽ yêu cầu nachos tăng program counter trước khi system call trả kết quả về. Nếu không lập trình đúng phần này thì sẽ bị vòng lặp gọi thực hiện system call này mãi mãi. Cũng như các hệ thống khác, MIPS xử lý dựa trên giá trị của program counter, vì vậy bạn phải viết mã để tăng giá trị biến program counter, tìm đoạn mã này trong thư mục **machine**

(file/code/machine/mipssim.c, dòng thứ 118 và dòng thứ 560 đến 564). Bạn phải copy mã này vào vị trí thích hợp trong phần xử lý các system call của bạn. Hiện tại bạn phải dùng **Halt system call** tại cuối mỗi user program.

- Được viết riêng trong hàm **IncreasePC ()**.

```
//Đẩy thanh ghi Program counter của hệ thống về sau 4 byte để tiếp tục ghi  
void IncreasePC()  
{  
    //Tăng program counter (thay thế Halt)  
    machine->registers[PrevPCReg] = machine->registers[PCReg];  
    machine->registers[PCReg] = machine->registers[NextPCReg];  
    machine->registers[NextPCReg] += 4;  
}
```

- Ta sử dụng `machine->registers` để gán `PrevPCReg = PCReg`, `PCReg = NextPCReg`, `NextPCReg = NextPCReg + 4` nhằm đẩy giá trị các thanh ghi trên tăng dần.  
Đặt chuỗi lệnh này ở cuối cùng case `SyscallException` trước break (khi kết thúc một *system call* sẽ vắng khỏi `swich(type)` và nhảy tới chuỗi lệnh này) thay thế cho hàm `Halt()` cũ.

### 3. Cài đặt system call.

*Các bước thực hiện khi cài đặt một system call mới:*

**Bước 1:** Vào `userprog/syscall.h` để define một system call mới kèm theo là mã system call.

**Bước 2:** Vào `code/test/start.c` và `code/test/start.s` để khai báo tên system call sẽ viết trong `exception.cc`.

**Bước 3:** Vào `userprog/exception.cc` cài đặt system call nằm trong cấu trúc rẽ nhánh của `SyscallException`.

**Bước 4:** Tạo một tập tin `.c` và viết chương trình ở mức người dùng trong `code/test`.

**Bước 5:** Vào `code/test/Makefile` thêm tập tin vừa tạo ở hàng `all:... .`

**Bước 6:** Biên dịch lại `nachos`.

### a) Mô tả cài đặt ReadInt:

- Input: Không có.
- Output: Số nguyên.
- Công dụng: Đọc một số nguyên do người dùng nhập vào.
- Cấp phát vùng nhớ kiểu char để chứa chuỗi số đọc vào.
- Đọc thông tin vào vùng nhớ vừa cấp phát bằng phương thức **Read** trong **SynchConsole**. Trả về số byte đọc được vào biến **length**.
- Nếu **length < 1** thì gán thanh ghi kết quả = -1, tăng PC lên.
- Kiểm tra tính đúng đắn của chuỗi ký số nhập vào (chỉ gồm các ký tự từ '0' đến '9'). Nếu sai gán thanh ghi kết quả = 0, tăng PC. Chuyển chuỗi số thành số nguyên  $KQ = ktu - '0' + KQ * 10$  – Viết kết quả vào thanh ghi kết quả.

### b) Mô tả cài đặt PrintInt:

- Input: Số nguyên.
- Output: Chuỗi số nguyên.
- Công dụng: In một số nguyên là tham số truyền vào ra màn hình.
- Đọc giá trị trong thanh ghi số 4 (tham số duy nhất).
- Chuyển đổi số thành chuỗi theo công thức:  $chuoi = so \% 10 + '0'$ .
- Dùng hàm **Write** trong **SynchConsole** để in chuỗi ra màn hình.

### c) Mô tả cài đặt ReadChar:

- Input: Không có.
- Output: Ký tự.
- Công dụng: Đọc một ký tự do người dùng nhập vào.
- Dùng hàm **Read** trong **SynchConsole** để đọc ký tự được nhập vào.
- Dùng hàm **write** để viết vào thanh ghi kết quả.

#### d) Mô tả cài đặt void PrintChar(char character):

- Input: Ký tự
  - Output: Ký tự.
  - Công dụng: In một ký tự làm tham số truyền vào ra màn hình.
- Dùng hàm **Write** trong **SynchConsole** để viết ký tự ra màn hình.
  - Sử dụng lớp SynchConsole để xuất một ký tự ra màn hình.
  - Đọc giá trị từ thanh ghi r4 và dùng hàm **Write** thuộc lớp **SynchConsole** để in ký tự đọc được ra màn hình.

#### e) Mô tả cài đặt void ReadString(char[]buffer, int lenght):

- Input: Buffer và độ dài chuỗi nhập vào.
  - Output: Không có.
  - Công dụng: đọc vào một chuỗi và độ dài của nó.
- Sử dụng lớp SynchConsole 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 **lenght**).
  - Chú ý rằng buffer là một vùng nhớ userspace, khi người dùng nhập chuỗi thì nội dung được lưu trữ ở kernel space -> cần viết một hàm tương ứng để chuyển dữ liệu từ kernelspace sang userspace.
  - Lấy địa chỉ buffer từ r4 và độ dài chuỗi tối đa từ r5, đọc chuỗi bằng **SynchConsole.Read** sau đó copy chuỗi từ System space sang User space bằng hàm System2User.

#### f) Mô tả cài đặt void PrintString (char[] buffer):

- Input: Buffer
  - Output: Chuỗi có địa chỉ buffer.
  - Công dụng: In một chuỗi có địa chỉ buffer truyền vào ra màn hình.
- Chú ý tương tự như ReadString cần phải có hàm để chuyển đổi dữ liệu từ userspace qua kernel space.



- Đưa buffer vào từ thanh ghi r4, chuyển dữ liệu từ User sang phía System bằng User2System, đếm độ dài thật của chuỗi cần in và gọi `SynchConsole.Write` để in chuỗi đó ra màn hình.

#### 4. Viết chương trình:

##### a) Chương trình help:

- Dùng hàm **PrintString()** đã viết ở trên để xuất các chuỗi hướng dẫn ra màn hình.
- Chương trình được viết trong file **help.c**.

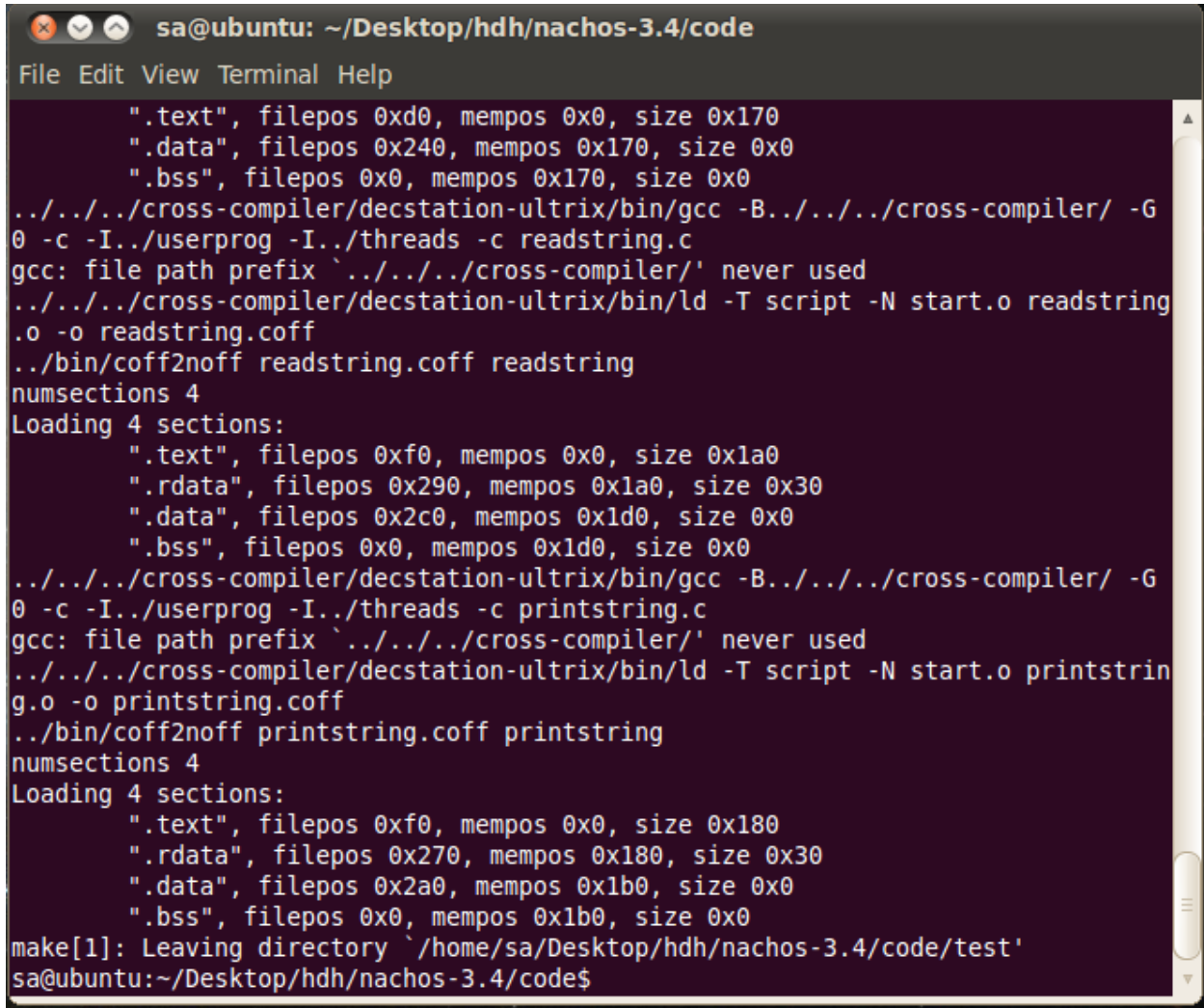
##### b) Chương trình ascii:

- In các kí tự trong bảng mã ASCII bằng các chuyển từ int sang char rồi dùng hàm **PrintChar()** để in các kí tự ra màn hình.
- Chương trình được viết trong file **ascii.c**.

##### c) Chương trình sort:

- Tạo một mảng một chiều, đọc các số nguyên vào các phần tử bằng hàm **ReadInt()**.
- Dùng thuật toán BubbleSort để sắp xếp.
- In các phần tử của mảng đã sắp xếp ra màn hình bằng hàm **PrintInt()**.

## 5. Các hình ảnh minh họa kết quả:



```
sa@ubuntu: ~/Desktop/hdh/nachos-3.4/code
File Edit View Terminal Help

    ".text", filepos 0xd0, mempos 0x0, size 0x170
    ".data", filepos 0x240, mempos 0x170, size 0x0
    ".bss", filepos 0x0, mempos 0x170, size 0x0
../../../../cross-compiler/decstation-ultrix/bin/gcc -B../../../../cross-compiler/ -G
0 -c -I../userprog -I../threads -c readstring.c
gcc: file path prefix `../../../../cross-compiler/' never used
../../../../cross-compiler/decstation-ultrix/bin/ld -T script -N start.o readstring
.o -o readstring.coff
../bin/coff2nooff readstring.coff readstring
numsections 4
Loading 4 sections:
    ".text", filepos 0xf0, mempos 0x0, size 0x1a0
    ".rdata", filepos 0x290, mempos 0x1a0, size 0x30
    ".data", filepos 0x2c0, mempos 0x1d0, size 0x0
    ".bss", filepos 0x0, mempos 0x1d0, size 0x0
../../../../cross-compiler/decstation-ultrix/bin/gcc -B../../../../cross-compiler/ -G
0 -c -I../userprog -I../threads -c printstring.c
gcc: file path prefix `../../../../cross-compiler/' never used
../../../../cross-compiler/decstation-ultrix/bin/ld -T script -N start.o printstrin
g.o -o printstring.coff
../bin/coff2nooff printstring.coff printstring
numsections 4
Loading 4 sections:
    ".text", filepos 0xf0, mempos 0x0, size 0x180
    ".rdata", filepos 0x270, mempos 0x180, size 0x30
    ".data", filepos 0x2a0, mempos 0x1b0, size 0x0
    ".bss", filepos 0x0, mempos 0x1b0, size 0x0
make[1]: Leaving directory `/home/sa/Desktop/hdh/nachos-3.4/code/test'
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$
```

Figure 3: Make all thành công.

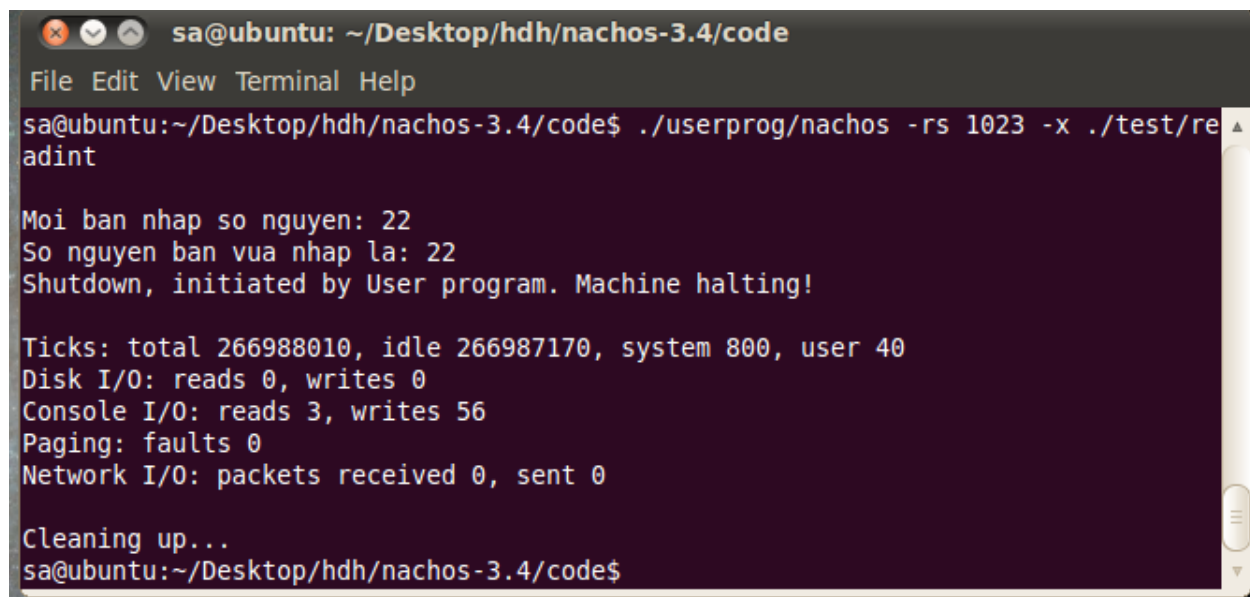
```
make[1]: Leaving directory '/home/sa/Desktop/hdh/nachos-3.4/code/test'
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/halt
lt

Shutdown, initiated by User program. Machine halting!

Ticks: total 42, idle 0, system 30, user 12
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
```

Figure 4: Demo halt.



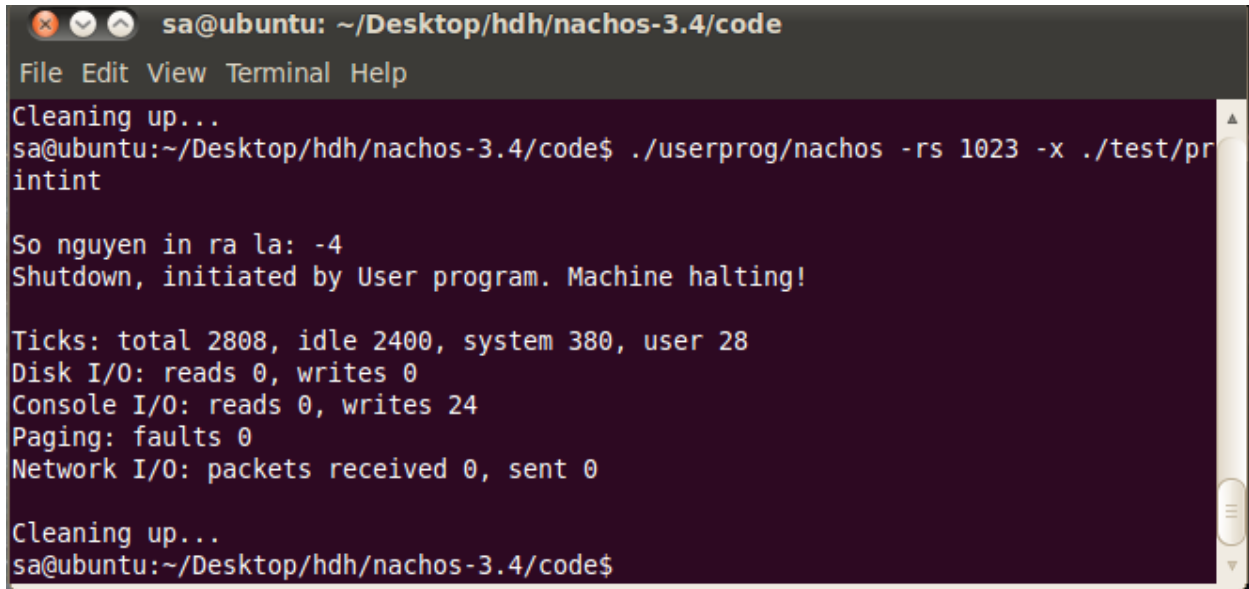
```
sa@ubuntu: ~/Desktop/hdh/nachos-3.4/code
File Edit View Terminal Help
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/readint

Moi ban nhap so nguyen: 22
So nguyen ban vua nhap la: 22
Shutdown, initiated by User program. Machine halting!

Ticks: total 266988010, idle 266987170, system 800, user 40
Disk I/O: reads 0, writes 0
Console I/O: reads 3, writes 56
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$
```

Figure 5: Demo readint.



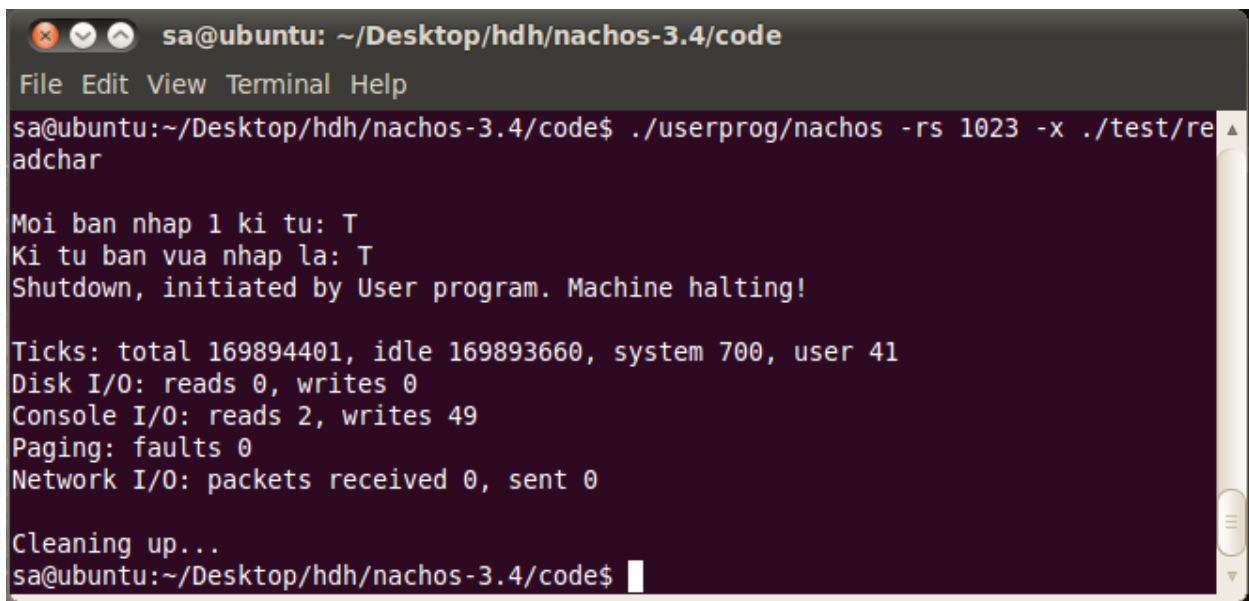
```
sa@ubuntu: ~/Desktop/hdh/nachos-3.4/code
File Edit View Terminal Help
Cleaning up...
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/pr
intint

So nguyen in ra la: -4
Shutdown, initiated by User program. Machine halting!

Ticks: total 2808, idle 2400, system 380, user 28
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 24
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$
```

Figure 6: Demo printint.



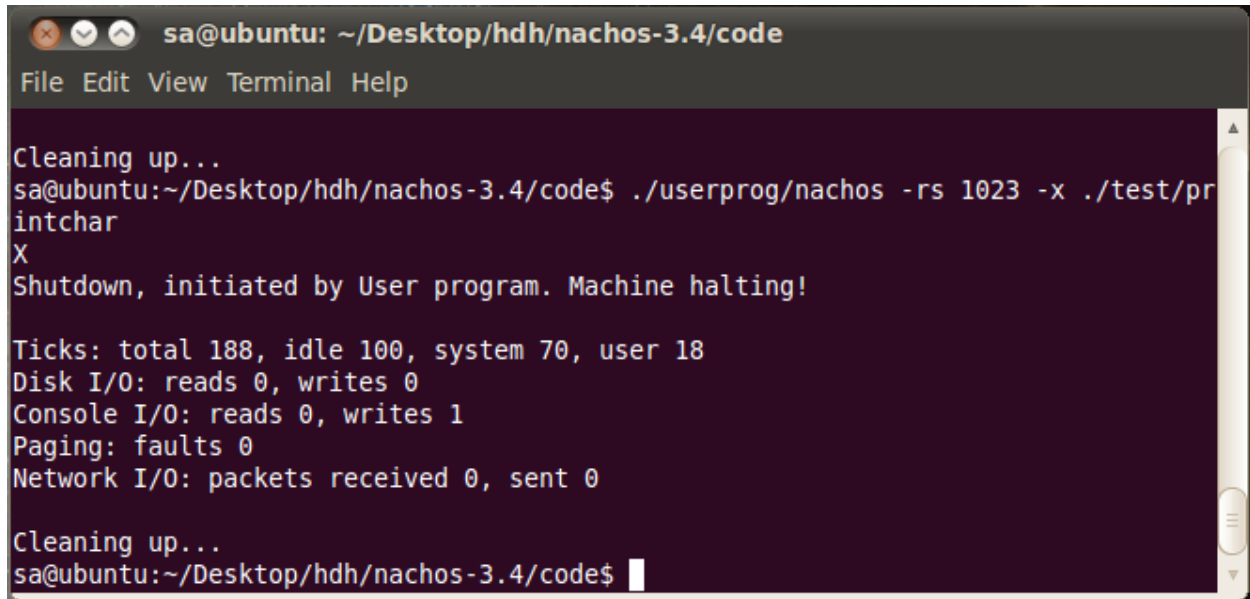
```
sa@ubuntu: ~/Desktop/hdh/nachos-3.4/code
File Edit View Terminal Help
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/re
adchar

Moi ban nhap 1 ki tu: T
Ki tu ban vua nhap la: T
Shutdown, initiated by User program. Machine halting!

Ticks: total 169894401, idle 169893660, system 700, user 41
Disk I/O: reads 0, writes 0
Console I/O: reads 2, writes 49
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$
```

Figure 7: Demo readchar.



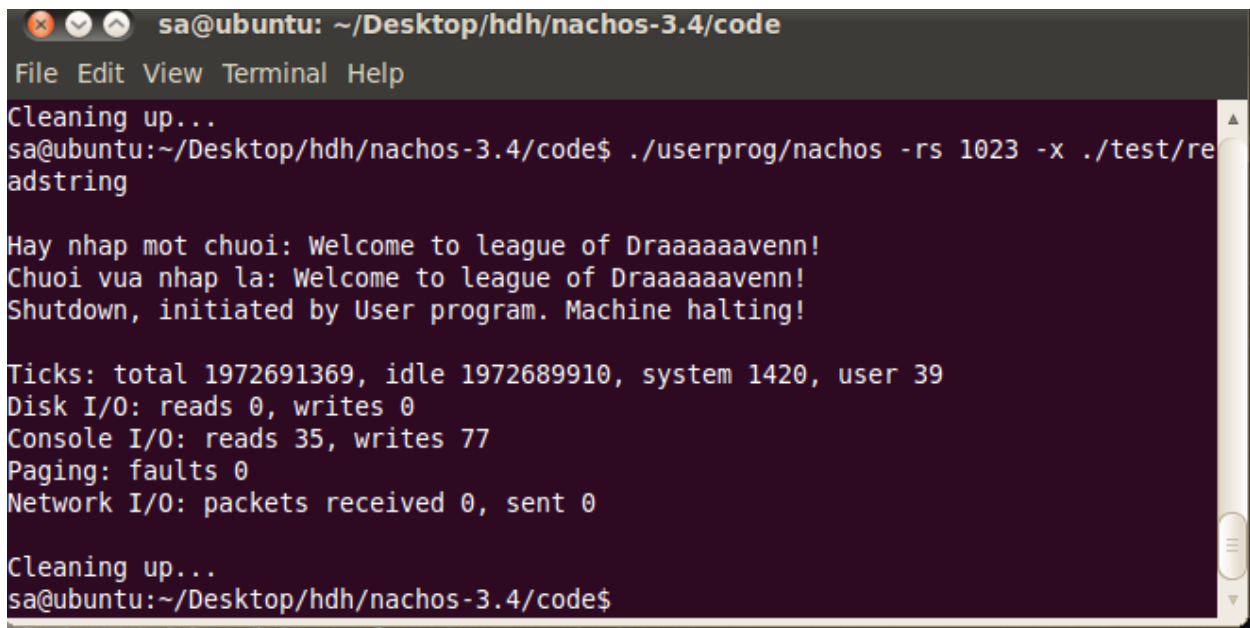
```
sa@ubuntu: ~/Desktop/hdh/nachos-3.4/code
File Edit View Terminal Help

Cleaning up...
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/printchar
X
Shutdown, initiated by User program. Machine halting!

Ticks: total 188, idle 100, system 70, user 18
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 1
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$
```

Figure 8: Demo printchar.



```
sa@ubuntu: ~/Desktop/hdh/nachos-3.4/code
File Edit View Terminal Help

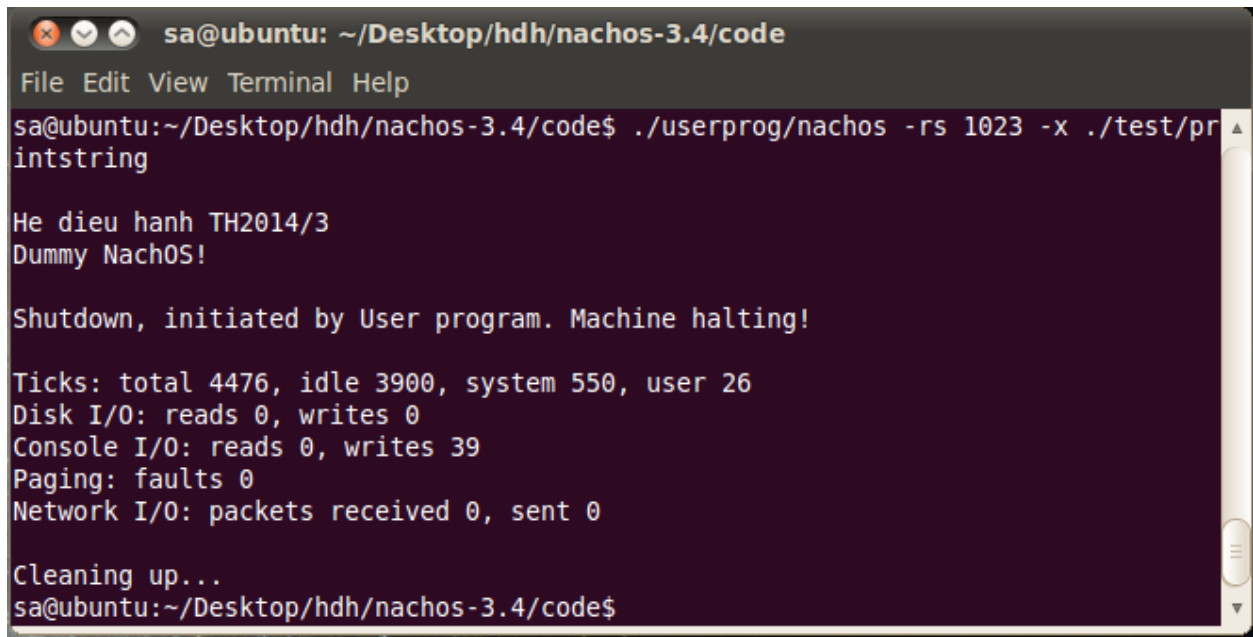
Cleaning up...
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/readstring

Hay nhap mot chuoi: Welcome to league of Draaaaaavenn!
Chuoi vua nhap la: Welcome to league of Draaaaaavenn!
Shutdown, initiated by User program. Machine halting!

Ticks: total 1972691369, idle 1972689910, system 1420, user 39
Disk I/O: reads 0, writes 0
Console I/O: reads 35, writes 77
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$
```

Figure 9: Demo readstring.



```
sa@ubuntu: ~/Desktop/hdh/nachos-3.4/code
File Edit View Terminal Help
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/printstring

He dieu hanh TH2014/3
Dummy NachOS!

Shutdown, initiated by User program. Machine halting!

Ticks: total 4476, idle 3900, system 550, user 26
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 39
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$
```

Figure 10: Demo printstring.

```
sa@ubuntu: ~/Desktop/hdh/nachos-3.4/code
File Edit View Terminal Help
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/help
lp

DH Khoa hoc tu nhien TPHCM
He Dieu Hanh TH2014

----- HELP FILE -----

Thong tin nhom:
Nhom gom co 3 thanh vien:
Doan Hieu Tam - 1412477
Tran Thi Nha - 1412363
Nguyen Dinh Son - 1412465

Mo ta van tat ve chuong trinh Ascii:
Chuong trinh dung de in ra bang ma Ascii gom 95 ki tu (tu 32 -> 127).
Duoc cai dat bang cach su dung vong lap (for statement) va goi ham PrintChar(char) de in ra.

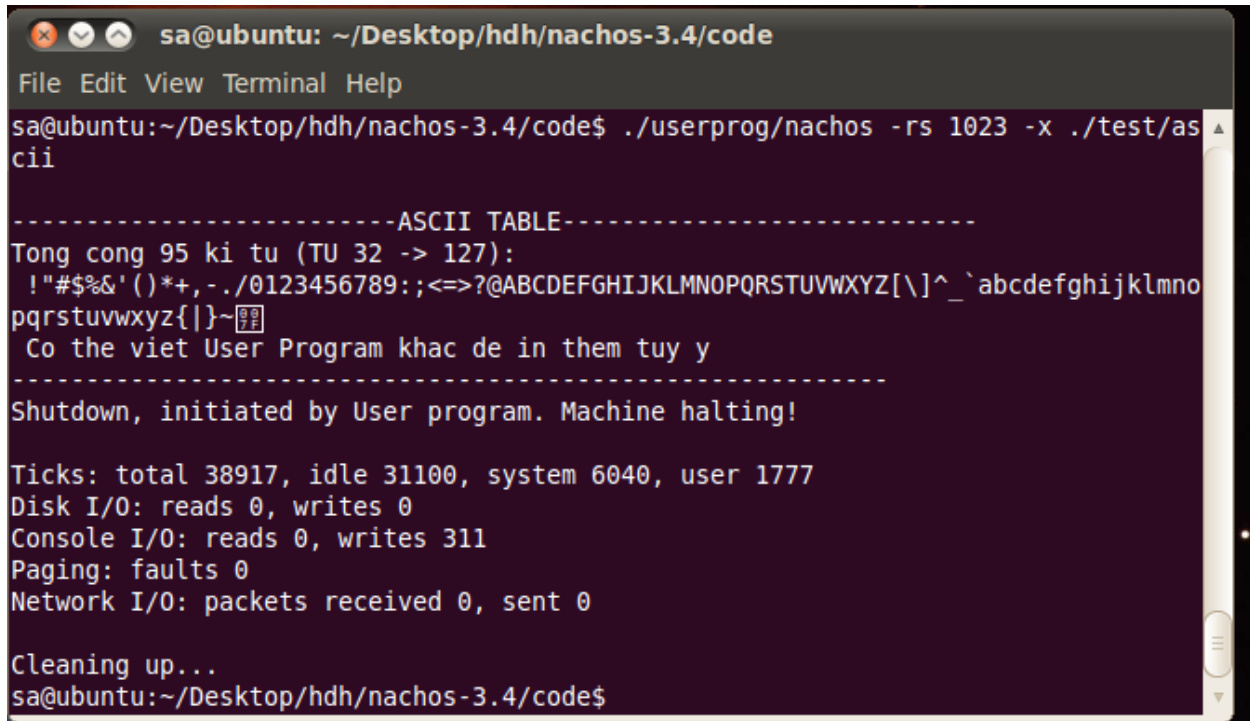
Mo ta van tat ve chuong trinh Sort:
Chuong trinh cho phép người dùng nhập vào mảng n số nguyên (với  $1 \leq n \leq 100$ ).
Duoc cai dat bang cach su dung thuat toan sap xep noi bot (Bubble sort) de sap xep mang va goi cac ham ReadInt(), PrintInt(int) de nhap va in mang.

Chuc thay co mot ngay lam viec vui ve :D

-----
Shutdown, initiated by User program. Machine halting!

Ticks: total 91456, idle 81700, system 9590, user 166
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 817
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Figure 11: Demo help.



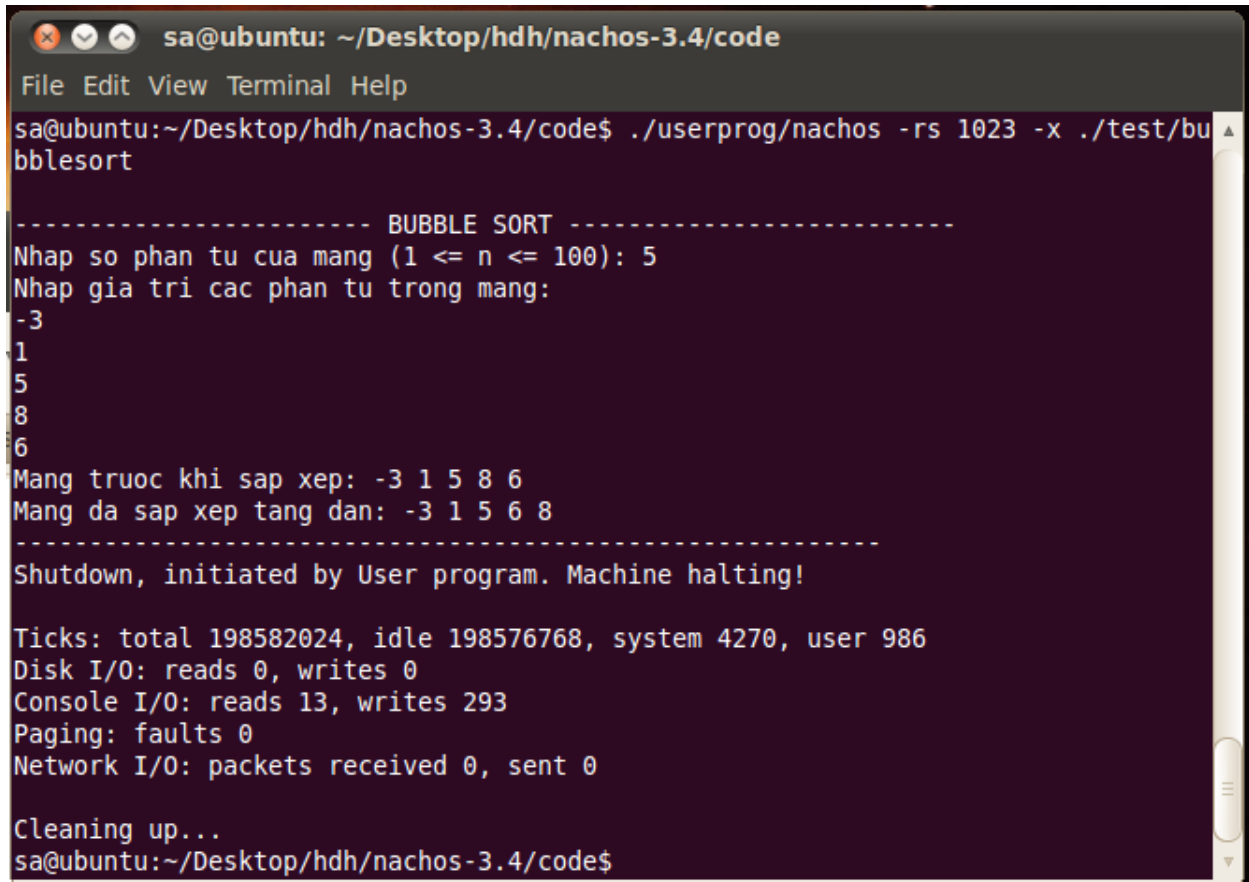
```
sa@ubuntu: ~/Desktop/hdh/nachos-3.4/code
File Edit View Terminal Help
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/ascii
-----ASCII TABLE-----
Tong cong 95 ki tu (TU 32 -> 127):
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN O PQRSTUVWXYZ[\]^_`abcdefghijklmnop
qrstuvwxyz{|}~
Co the viet User Program khac de in them tuy y
-----
Shutdown, initiated by User program. Machine halting!

Ticks: total 38917, idle 31100, system 6040, user 1777
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 311
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$
```

Figure 12: Demo ascii.





```
sa@ubuntu: ~/Desktop/hdh/nachos-3.4/code
File Edit View Terminal Help
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/bubblesort

----- BUBBLE SORT -----
Nhap so phan tu cua mang (1 <= n <= 100): 5
Nhap gia tri cac phan tu trong mang:
-3
1
5
8
6
Mang truoc khi sap xep: -3 1 5 8 6
Mang da sap xep tang dan: -3 1 5 6 8
-----
Shutdown, initiated by User program. Machine halting!

Ticks: total 198582024, idle 198576768, system 4270, user 986
Disk I/O: reads 0, writes 0
Console I/O: reads 13, writes 293
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
sa@ubuntu:~/Desktop/hdh/nachos-3.4/code$
```

Figure 13: Demo bubblesort.

### III. Tài liệu tham khảo:

- Slide[1] Biên dịch và cài đặt Nachos.
- Slide[2] Giao tiếp giữa hệ điều hành Nachos và người dùng.
- Slide[3] Cách viết một system call.
- Slide[4] Cách thêm một lớp vào Nachos.
- <http://dotrungduchd.blogspot.com/2013/05/system-call-of-nachos.html>



Trường: ĐH Khoa học tự nhiên TP.HCM  
Khoa: Công nghệ thông tin  
Lớp: TH2014/03