ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN 1

EXCEPTIONS VÀ CÁC SYSTEM CALLS ĐƠN GIẢN

Môn: **HỆ ĐIỀU HÀNH 20_3**

Thành phố Hồ Chí Minh 2022

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

EXCEPTIONS VÀ CÁC SYSTEM CALL ĐƠN GIẢN

| Nhóm thực hiện đồ án |

20120307 Phạm Gia Khiêm

20120519 Nguyễn Thị Thúy Liễu

20120540 Võ Hoàng Thảo Nguyên

| Giáo viên hướng dẫn |

***Thầy:* Nguyễn Thanh Quân**

Môn: **HỆ ĐIỀU HÀNH 20_3**

Thành phố Hồ Chí Minh 2022

LỜI CẢM ƠN

Kính gửi: Khoa Công nghệ Thông tin, Trường Đại học Khoa Học Tự Nhiên.

Chúng em xin chân thành cảm ơn *Khoa Công nghệ Thông tin* đã tạo điều kiện thuận lợi cho chúng em học tập và thực hiện bài báo cáo đồ án về xử lý các exceptions và các system calls cơ bản này.

Chúng em xin bày tỏ lòng biết ơn sâu sắc tới thầy **Nguyễn Thanh Quân** đã tận tình hướng dẫn chỉ bảo chúng em trong quá trình thực hiện bài báo cáo.

Chúng em xin cảm ơn sự quan tâm giúp đỡ và ủng hộ của bạn bè trong quá trình thực hiện bài báo cáo. Mặc dù đã cố gắng hoàn thành bài báo cáo luận trong phạm vi và khả năng cho phép nhưng chắc chắn sẽ không tránh khỏi những thiếu sót.

Chúng em rất mong nhận được những sự thông cảm, những góp ý tận tình và những lời chỉ bảo của quý thầy cô và các bạn.

Thành phố Hồ Chí Minh, ngày 16 tháng 10 năm 2022.

Nhóm sinh viên thực hiện:

STT MSSV	MSSV	Họ và tên
1	20120307	Phạm Gia Khiêm
2	20120519	Nguyễn Thị Thúy Liễu
3	20120540	Võ Hoàng Thảo Nguyên

MỤC LỤC

DANH MỤC HÌNH

TỔNG QUAN CÀI ĐẶT

1. Các cài đặt có sẵn:

- **progtest.cc** kiểm tra các thủ tục để chạy chương trình người dùng.
- **syscall.h** system call interface: các thủ tục ở kernel mà chương trình người dùng có thể gọi.
- **exception.cc** xử lý system call và các exception khác ở mức user, ví dụ như lỗi trang, trong phần mã chúng tôi cung cấp, chỉ có 'halt' system call được viết.
- **bitmap.*** các hàm xử lý cho lớp bitmap (hữu ích cho việc lưu vết các ô nhớ vật lý) **filesys.h .**
- **openfile.h** định nghĩa các hàm trong hệ thống file nachos. Trong đồ án này chúng ta sử dụng lời gọi thao tác với file trực tiếp từ Linux, trong đồ án khác chúng ta sẽ triển khai hệ thống file trên ổ đĩa giả lập. (nếu kịp thời gian).
- **translate.*** Phiên bản nachos chúng tôi gửi các bạn, chúng tôi giả sử mỗi địa chỉ ảo là cũng giống hệt như địa chỉ vật lý, điều này giới hạn chúng ta chỉ chạy 1 chương trình tại một thời điểm. Các bạn có thể viết lại phần này để cho phép nhiều chương trình chạy cùng lúc trong đồ án sau.
- **machine.*** mô phỏng các thành phần của máy tính khi thực thi chương trình người dùng: bộ nhớ chính, thanh ghi, v.v.
- **mipssim.cc** mô phỏng tập lệnh của MIPS R2/3000 processor
- **console.*** mô phỏng thiết bị đầu cuối sử dụng UNIX files. Một thiết bị có đặc tính (i) đơn vị dữ liệu theo byte, (ii) đọc và ghi các bytes cùng một thời điểm, (iii) các bytes đến bất đồng bộ.
- **synchconsole.*** nhóm hàm cho việc quản lý nhập xuất I/O theo dòng trong Nachos.
- **../test/*** Các chương trình C sẽ được biên dịch theo MIPS và chạy trong Nachos

2. Các giá trị thanh ghi

- *Register 2*: Lưu mã syscall đồng thời lưu kết quả trả về của mỗi syscall nếu có.
- *Register 4*: Lưu tham số thứ nhất.
- *Register 5*: Lưu tham số thứ hai.
- *Register 6*: Lưu tham số thứ ba.
- *Register 7*: Lưu tham số thứ tư.

CÀI ĐẶT SYSTEM CALLS VÀ EXCEPTIONS

1. **Cài đặt lại các Exception:**

- ***PageFaultException***: Page-fault xuất hiện khi 1 chương trình cố gắng truy cập dữ liệu hoặc mã code, cái mà có trong không gian địa chỉ của nó, nhưng hiện không nằm trong RAM (Random Access Memory) của hệ thống (system).
- ***ReadOnlyException***: Read-only là thuộc tính tệp chỉ cho phép người dùng xem tệp, hạn chế bất kỳ hành động ghi vào tệp.
- ***BusErrorException***: Bus-error là lỗi do phần cứng gây ra, thông báo cho OS (Operating System – Hệ điều hành) rằng 1 quá trình đang cố gắng truy cập bộ nhớ mà CPU (Center Processing Unit) không thể xử lý vật lý (physically address): Một địa chỉ bus không hợp lệ hoặc không có tên.
- ***AddressErrorException***: Một lỗi địa chỉ được phát hiện bằng cách xác định xem đầu ra địa chỉ đích đến trên đường dẫn địa chỉ có được phát hiện tại bộ nhớ hay không.
- ***OverflowException***: Overflow xảy ra khi một phép toán số học tạo ra một kết quả nằm ngoài phạm vi trả về của kiểu dữ liệu mà phép toán trả về.
- ***IllegalInstrException***: Illegal Instruction phát sinh vì không phải mọi giá trị bộ nhớ đều có thể tương ứng với một lệnh hợp lệ.
- ***NumExceptionTypes***: Là một dạng exception xảy ra khi cố gắng chuyển đổi một chuỗi số có định dạng không chính xác thành một giá trị số.

2. **Cài đặt hàm increasePC():**

- Công dụng: Nhằm tránh trường hợp máy tính nạp lệnh tại 1 chỗ, tạo thành vòng lặp vô hạn sau mỗi system call của chương trình.
- Ý tưởng: Tăng giá trị của PC[1] để khi nạp lệnh tiếp theo chương trình không quay lại SC[2] cũ mà tiếp tục chương trình.
- Cách cài đặt hàm ***increasePC()***: Ta lưu giá trị của PC hiện tại cho PC trước, nạp tiếp giá trị kế cho PC hiện tại, nạp giá trị kế tiếp nữa cho PC kế.

3. **Cài đặt syscall ReadChar:**

- Mô tả cài đặt *SC_ReadChar*:
  - Input: 1 ký tự (char).
  - Output: NULL.
  - Công dụng: Đọc 1 ký tự bất kỳ mà người dùng nhập vào từ bàn phím.
- Ý tưởng: Thiết lập cách thức để người dùng nhập từ bàn phím, cho đến khi người dùng enter thì hệ thống sẽ ngừng lấy, giống như ngôn ngữ C, hệ thống sẽ chỉ lấy và in ra ký tự đầu tiên trong chuỗi ký tự mà người dùng vừa nhập vào.
- Cách cài đặt: Dùng lệnh ***GetChar()*** từ **synconsoleInput** để lấy ký tự. Gọi *ch* là ký tự đầu tiên được lấy về và dùng vòng while để lấy hết các ký tự sau nó (nếu có) cho tới khi gặp ký tự xuống dòng '\n', sau đó ghi vào thanh ghi *reg2* ký tự *ch* (ký tự đầu tiên).

4. **Cài đặt syscall PrintChar:**

- Mô tả cài đặt *SC_PrintChar*:
  - Input: NULL.
  - Output: 1 ký tự (char).
  - Công dụng: Lấy 1 ký tự đã được lưu trữ và in ra màn hình
- Ý tưởng: Lấy dữ liệu là 1 ký tự từ thanh ghi tham số (*reg4*) và đưa vào hệ thống, để hệ thống xử lý và in ra màn hình.
- Cách cài đặt: Gọi *ch* là ký tự mà ta đọc từ thanh ghi *reg4*. Sau đó, dùng lệnh ***PutChar()*** từ **synconsoleOutput** để hệ thống in ký tự ra màn hình.

5. **Cài đặt syscall ReadNum:**

- Mô tả cài đặt *SC_ReadNum*:
  - Input: NULL.
  - Output: Số nguyên int.
  - Công dụng: Đọc số nguyên từ bàn phím
- Ý tưởng: Đọc từng ký tự nhập từ bàn phím, nếu là dấu – hoặc số thì hệ thống sẽ nhận lệnh xử lý. Xử lý từng ký tự số sau đó gộp lại và trả về kết quả là 1 số int trong khoảng 4byte.

- Cách cài đặt: Tạo 1 *char* buffer* để chứa ký tự số mà người dùng nhập, sau đó sẽ đọc lần lượt các ký tự giống như ***SC_ReadChar*** cho đến khi kết thúc dòng. Sau đó dựa vào cấu trúc của *buffer* để xét thỏa điều kiện của một số **int** (4byte). Dùng vòng while để kiểm tra từng ký tự xem có phải là chữ số hay không. Tiếp đến, nếu đã xác định được dấu và chữ số mà người dùng nhập vào, thì ta sẽ tiến hành đưa về *int num* thay vì là *char* buffer*. Xác định dấu của *num*, sau đó gán abs(*num*) vào *int result* và lưu giá trị *result* vào thanh ghi *reg2*. Trường hợp ký tự nhập vào không phải là số thì kết quả trả về sẽ là 0 (*result* = 0).

6. Cài đặt syscall PrintNum:

- Mô tả cài đặt *SC_PrintNum*:
  - Input: Số nguyên int.
  - Output: NULL.
  - Công dụng: In một số nguyên ra màn hình.
- Ý tưởng: Lấy giá trị *int number* từ thanh ghi *reg4* sau đó phân tách *number* thành từng số hạng và đưa vào *char* buffer*. Vì kiểm tra từ số hạng nhỏ nhất trong *number* nên *buffer* sẽ bị ngược, do đó chúng ta sẽ ghi từng ký tự trong *buffer* theo chiều ngược lại (*i*--) để kết quả đưa lên hệ thống được chính xác.

- Cách cài đặt: Đọc *int number* từ *reg4*, tạo *char* buffer*, dùng vòng lặp while để tách từng số hạng trong *number*, và *int num* sẽ lưu từng số hạng và gán vào vị trí tương ứng của *buffer*. Xác định number mang giá trị âm hay giá trị dương để quyết định có thêm dấu '-' vào *buffer* hay không. Cuối cùng, lần lượt đưa từng ký tự của *buffer* theo chiều ngược lại (*i*--) cho hệ thống xử lý và in ra màn hình.

7. Cài đặt syscall ReadString:

- Mô tả *cài đặt SC_ReadString*:
  - Input: NULL.
  - Output: Chuỗi ký tự.
  - Công dụng: In một chuỗi ký tự ra màn hình.

- Ý tưởng: Đọc từng ký tự từ màn hình và lưu vào char* buffer, sau đó sẽ chuyển dữ liệu vùng nhớ nhờ ***System2User()***, thành vùng nhớ ảo.
- Cách cài đặt: Tạo một *char* buffer* có độ dài *length* nhất định (do người dùng nhập vào), sau đó duyệt từng ký tự mà người dùng nhập vào có độ dài length, cho đến khi gặp ký tự '\n' hoặc '\0'. Tiếp theo ta sẽ kiểm tra lại trường hợp tràn *buffer*. Cuối cùng, gọi *ptr* là giá trị của *reg4*, gọi hàm ***System2User()*** để tạo vùng nhớ ảo tại *ptr*, với chuỗi ký tự *buffer* và độ dài length.

8. **Cài đặt syscall PrintString:**

- Mô tả cài đặt *syscall PrintString()*:
  - Input: NULL.
  - Output: Một chuỗi ký tự.
  - Công dụng: In 1 chuỗi ký tự ra màn hình.
- Ý tưởng cài đặt: Khi đọc vào ta tạo và lưu vào *reg4*, 1 virtual Address nên ta lấy giá trị từ *reg4* sau đó lần lượt in từng ký tự ra màn hình.
- Cách cài đặt: Gọi ptr là giá trị của *reg4*, sau đó dùng hàm ***User2System()*** để lấy từ vùng nhớ ảo chuỗi ký tự *char** mà ta đã đọc vào. Truyền từng ký tự lênh hệ thống để in ra màn hình.

9. **Cài đặt chương trình help:**

- Mô tả cài đặt *chương trình help:*
  - Input: NULL.
  - Output: Các dòng ký tự.
  - Công dụng: In ra màn hình các dòng cơ bản giới thiệu nhóm, và mô tả vắn tắt chương trình sort với ascii.
- Cách cài đặt: Dựa vào ***SC_PrintString***, hàm ***PrintString()*** để in các câu giới thiệu và bài mô tả chương trình.

10. **Cài đặt chương trình sort:**

- Mô tả cài đặt *chương trình sort*:
  - Input: Chuỗi số mà người dùng nhập từ bàn phím và cách sort.

- Output: Chuỗi số đã được chương trình sort .
- Công dụng: Sort dãy số theo yêu cầu của người dùng.

- Cách cài đặt: Dựa vào ***SC_ReadNum***, dùng hàm ***ReadNum()*** để đọc từng số người dùng nhập vào và lưu vào *int arr[]*, và cách thức sort với: **0** – là tăng dần, **1** – là giảm dần. Sau đó dựa vào ***SC_PrintNum***, dùng hàm ***PrintNum()*** để in các ký tự ra màn hình.

11. Cài đặt chương trình ascii:

- Mô tả cài đặt *chương trình ascii*:
  - Input: NULL.
  - Output: In ra các ký tự.
  - Công dụng: In ra màn hình các ký tự trong bảng mã ascii.
- Cách cài đặt: Dựa vào ***SC_PrintChar***, dùng hàm ***PrintChar()*** để in ra các ký tự trong bảng mã ascii mà người dùng đọc được (từ 32 – 126 theo kiểu dữ liệu *int*)

DEMO CHƯƠNG TRÌNH VÀ HÌNH ẢNH MINH HỌA

1. Biên dịch chương trình:

a) Biên dịch build.linux:

- Khi make depend:

```
st3@remote_nachos:~/OS/HeDieuHanh_20_3/nachos/NachOS-4.0/code/build.linux$ make depend
g++ -I../network -I../filesys -I../userprog -I../threads -I../machine -I../lib -I- -DFILESYS_STUB -DRDATA -DSIM_FIX -Dx86 -DLINUX -DCHANGED -M ../lib/bitmap.cc ../lib/debug.cc ../lib/hash.cc ../lib/libtest.cc ../lib/list.cc ../lib/sysdep.cc ../machine/interrupt.cc ../machine/stats.cc ../machine/timer.cc ../machine/console.cc ../machine/machine.cc ../machine/mipssim.cc ../machine/translate.cc ../machine/network.cc ../machine/disk.cc ../threads/alarm.cc ../threads/kernel.cc ../threads/main.cc ../threads/scheduler.cc ../threads/synch.cc ../threads/synchlist.cc ../threads/thread.cc ../userprog/addrspace.cc ../userprog/exception.cc ../userprog/synchconsole.cc ../filesys/directory.cc ../filesys/filehdr.cc ../filesys/filesys.cc ../filesys/pbitmap.cc ../filesys/openfile.cc ../filesys/synchdisk.cc  ../network/post.cc > makedep
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
cc1plus: note: obsolete option -I- used, please use -iquote instead
```

Hình 1. Biên dịch build.linux make depend

- Khi make:

```
g++ bitmap.o debug.o libtest.o sysdep.o interrupt.o stats.o timer.o console.o machine.o mipssim.o translate.o network.o disk.o alarm.o kernel.o main.o scheduler.o synch.o thread.o addrspace.o exception.o synchconsole.o directory.o filehdr.o filesys.o pbitmap.o openfile.o synchdisk.o post.o switch.o -m32 -o nachos
```

Hình 2. Biên dịch build.linux make

b) Biên dịch test:

```
Loading 4 sections:
        ".text", filepos 0xf0, mempos 0x0, size 0x230
        ".rdata", filepos 0x320, mempos 0x230, size 0x20
        ".data", filepos 0x340, mempos 0x250, size 0x0
        ".bss", filepos 0x0, mempos 0x250, size 0x100
rm -f -f *.o *.ii
rm -f -f *.coff
```

Hình 3. Biên dịch test make

2. Chương trình chạy thử nghiệm các System Call:

a) SC_ReadChar và SC_PrintChar:

- Thực thi tại file test thông qua lệnh: ../build.linux/nachos –x char

```
(base) giakhiem@pzakhim:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x char

Nhap ki tu can in ra:   Xin thay cho em diem 10 Nachos
Ki tu vua nhap la:      X

 Shutdown, initiated by user program.Machine halting!

Ticks: total 885899846, idle 885897334, system 2470, user 42
Disk I/O: reads 0, writes 0
Console I/O: reads 31, writes 43
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Hình 4. Thực thi syscall ReadChar và PrintChar

b) SC_ReadString và SC_PrintString:

- Thực thi tại file test thông qua lệnh: ../build.linux/nachos –x string

```
(base) giakhiem@pzakhim:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x string

Nhap chuoi can in ra:   Mot lan nua em xin thay cho nhom em 10 diem Nachos
Chuoi vua nhap la:      Mot lan nua em xin thay cho nhom em 10 diem Nachos

 Shutdown, initiated by user program.Machine halting!

Ticks: total 1080654395, idle 1080649592, system 4760, user 43
Disk I/O: reads 0, writes 0
Console I/O: reads 51, writes 92
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Hình 5. Thực thi syscall ReadString và PrintString

c) SC_ReadNum và SC_PrintNum:

- Thực thi tại file test thông qua lệnh: ../build.linux/nachos –x num
- Trường hợp tràn số dương: Biên dương: 2.147.483.647

```
(base) giakhiem@pzakhim:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x num

Nhap so can in ra:      2147483648
So vua nhap la: 0

 Shutdown, initiated by user program.Machine halting!

Ticks: total 1235296655, idle 1235294974, system 1640, user 41
Disk I/O: reads 0, writes 0
Console I/O: reads 11, writes 38
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Hình 6. Trường hợp tràn số dương

- Trường hợp tràn số âm: Biên âm: - 2.147.483.648

```
(base) giakhiem@pzakhim:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x num

Nhap so can in ra:      -2147483649
So vua nhap la: 0

 Shutdown, initiated by user program.Machine halting!

Ticks: total 768679565, idle 768677854, system 1670, user 41
Disk I/O: reads 0, writes 0
Console I/O: reads 12, writes 38
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Hình 7. Trường hợp tràn số âm

- Trường hợp nhập vào số không hợp lệ:

```
(base) giakhiem@pzakhim:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x num

Nhap so can in ra:      123Dzooooooooooo
So vua nhap la: 0

 Shutdown, initiated by user program.Machine halting!

Ticks: total 854832635, idle 854830724, system 1870, user 41
Disk I/O: reads 0, writes 0
Console I/O: reads 18, writes 38
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Hình 8. Trường hợp nhập vào số không hợp lệ

- Trường hợp nhập vào đúng: (Có 4 trường hợp: Có khoảng trống ở đầu, cuối, cả đầu và cuối, và không có khoảng trống nhưng em chỉ test 1 trong 4):

```
(base) giakhiem@pzakhim:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x num

Nhap so can in ra:                  123321
So vua nhap la: 123321

 Shutdown, initiated by user program.Machine halting!

Ticks: total 1096979715, idle 1096977574, system 2100, user 41
Disk I/O: reads 0, writes 0
Console I/O: reads 20, writes 43
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Hình 9. Trường hợp nhập vào hợp lệ (có khoảng trống ở đầu)

d) RandomNum:

- Thực thi tại file test thông qua lệnh: ../build.linux/nachos –x random

```
(base) giakhiem@pzakhim:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x random

So nguyen ngau nhien:   9595832
So nguyen ngau nhien:   984663719
So nguyen ngau nhien:   1134437502
So nguyen ngau nhien:   1338138011
So nguyen ngau nhien:   81305314
So nguyen ngau nhien:   2014928409
So nguyen ngau nhien:   905052920
So nguyen ngau nhien:   2075650423
So nguyen ngau nhien:   1597269654
So nguyen ngau nhien:   435547877

 Shutdown, initiated by user program.Machine halting!

Ticks: total 44610, idle 33190, system 11110, user 310
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 332
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Hình 10. Thực thi lệnh random

3. Chương trình help:

- Thực thi trong file test bằng câu lệnh: ../build.linux/nachos –x help

```
(base) giakhiem@pzakhim:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x help

Nhom lap trinh SystemCall, gom 3 thanh vien:
        1. Pham Gia Khiem       - MSSV: 20120307
        2. Nguyen Thi Thuy Lieu - MSSV: 20120519
        3. Vo Hoang Thao Nguyen - MSSV: 20120540

Chuong trinh duoc lap trinh:

1. bubblesort
        -User se nhap vao mang A gom n (0<n<101) phan tu va lua chon phuong thuc sort:
                + Phuong thuc 0: Sort tang dan.
                + Phuong thuc 1: Sort giam dan.
2. ascii
        -Chuong trinh in ra ky tu thuoc bo ma ASCII.


 Shutdown, initiated by user program.Machine halting!

Ticks: total 56345, idle 42190, system 14080, user 75
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 422
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Hình 11. Thực thi chương trình help

4. Chương trình ascii:

- Thực thi trong file test bằng câu lệnh: ../build.linux/nachos –x ascii
  (Vì in ra quá dài nên em chỉ cắt ở đoạn hoàn thành)

```
116     t
117     u
118     v
119     w
120     x
121     y
122     z
123     {
124     |
125     }
126     ~


 Shutdown, initiated by user program.Machine halting!

Ticks: total 88288, idle 63090, system 21370, user 3828
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 631
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Hình 12. Thực thi chương trình ascii

5. Chương trình sort:

- Thực thi trong file test bằng câu lệnh: ../build.linux/nachos –x bubblesort
- Sau khi nhập số phần tử mảng và các phần tử mảng vào có 2 cách chọn là:

- 0 - Sort tăng dần.

```
(base) giakhiem@pzakhim:~/nachos/NachOS-4.0/code/test$ ../build.linux/nach
os -x bubblesort
Nhap so phan tu cua mang:       5
a[0] =  001
a[1] =  -2
a[2] =  5
a[3] =  -03
a[4] =  4
Chon cach sap xep (0: Tang dan, 1: Giam dan):   0
Mang sau sap xep tang dan:      -3      -2      1       4       5

 Shutdown, initiated by user program.Machine halting!

Ticks: total 1608908053, idle 1608900678, system 6200, user 1175
Disk I/O: reads 0, writes 0
Console I/O: reads 19, writes 164
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Hình 13. Thực thi chương trình sort tăng dần

- 1 - Sort giảm dần.

```
(base) giakhiem@pzakhim:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x bubblesort
Nhap so phan tu cua mang:       5
a[0] =  -4
a[1] =  12
a[2] =  3
a[3] =  0
a[4] =  -24
Chon cach sap xep (0: Tang dan, 1: Giam dan):   1
Mang sau sap xep giam dan:      12      3       0       -4      -24

 Shutdown, initiated by user program.Machine halting!

Ticks: total 1580817612, idle 1580810108, system 6240, user 1264
Disk I/O: reads 0, writes 0
Console I/O: reads 18, writes 166
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Hình 14. Thực thi chương trình sort giảm dần

DANH MỤC TỪ VIẾT TẮT

[1] PC: Programming Counter

[2] SC: System Call

TÀI LIỆU THAM KHẢO

[1] How to install NachOS on Ubuntu 18.04 (hcmus.edu.vn)

[2] Linear congruential generator - Wikipedia