

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

UNIVERSITY OF SCIENCES



BÁO CÁO ĐỒ ÁN 2 - NACHOS

Môn học: Hệ điều hành

Học kỳ I (2021 – 2022)

Sinh viên: 19120338 - Trần Hoàng Quân
19120383 - Huỳnh Tấn Thọ
19120407 - Lâm Hải Triều
19120426 - Phan Đặng Diễm Uyên
19120469 - Sử Nhật Đăng

Trưởng: Đại học Khoa học Tự Nhiên, ĐHQG - HCM

Giáo viên: Thầy Lê Viết Long

Thành phố Hồ Chí Minh, tháng 12 năm 2021

| | |
|---|----|
| Thông tin nhóm và công việc phân công | 3 |
| Cài đặt Nachos và Ubuntu 14.04 | 3 |
| - Biên dịch NachOS | 3 |
| - Chạy chương trình test | 4 |
| Cài đặt các yêu cầu của đề bài:..... | 4 |
| 1. Xử lý các exception..... | 4 |
| 2. Viết lại cấu trúc điều khiển của chương trình để nachos nhận các system call | 4 |
| 3. Viết lại hàm tăng program counter..... | 6 |
| 4. Xử lý các system call..... | 6 |
| • Cài đặt và khai báo synchcons cho console của nachos..... | 6 |
| • Cài đặt System2User và User2System để có thể chuyển vùng nhớ User về Kernel và ngược lại | 8 |
| • Forward declaration cho các system call để user program có thể call những system call này | 10 |
| • Cài đặt system call int ReadInt() | 12 |
| • Cài đặt system call void PrintInt(int number)..... | 12 |
| • Cài đặt system call char ReadChar() | 13 |
| • Cài đặt system call void PrintChar(char character)..... | 14 |
| • Cài đặt system call void ReadString(char* buffer, int length) | 14 |
| • Cài đặt system call void PrintString(char* buffer) | 14 |
| • Test thử các system call đã được cài..... | 15 |
| ○ Test PrintInt/ReadInt | 15 |
| ○ Test PrintChar/ReadChar | 17 |
| ○ Test PrintChar/ReadChar | 19 |
| 5. Viết các chương trình trên nachos | 20 |
| • Tóm tắt một số điều kiện cần | 20 |
| • Các bước chuẩn bị để viết 1 chương trình trên nachos | 20 |
| • Viết chương trình help | 21 |
| • Viết chương trình ascii | 23 |
| • Viết chương trình sort..... | 26 |
| Tài liệu tham khảo | 33 |

Thông tin nhóm và công việc phân công

| STT | MSSV | Họ và tên | Công việc | Tỉ lệ hoàn thành |
|-----|----------|---------------------|---|------------------|
| 01 | 19120338 | Trần Hoàng Quân | - Hướng dẫn cách sử dụng và compile nachos bằng Ubuntu bằng makefile - Cài đặt các exception - Viết báo cáo | 100% |
| 02 | 19120383 | Huỳnh Tấn Thọ | - Thêm hàm tăng program counter cho class Machine - Cài đặt PrintString/ReadString - Cài đặt chương trình ascii | 100% |
| 03 | 19120407 | Lâm Hải Triều | - Cài đặt ReadChar/PrintChar - Cài đặt chương trình help - Viết báo cáo | 100% |
| 04 | 19120426 | Phan Đăng Diễm Uyên | - Thêm hàm User2System/System2User vào class Machine - Viết chương trình chạy test system call (ioInt/ioChar/ioString) - Cài đặt PrintInt | 100% |
| 05 | 19120469 | Sử Nhật Đăng | - Cài đặt ReadInt - Cài đặt chương trình sort - Viết báo cáo | 100% |

Cài đặt Nachos và Ubuntu 14.04

Nhóm tiến hành cài đặt NachOS trên hệ điều hành Ubuntu 14.04 (sử dụng Oracle VirtualBox hoặc VMWare Workstation). Để thực hiện, Ubuntu phải cài đặt trình biên dịch GNU gcc bằng lệnh

```
sudo apt-get update && sudo apt-get upgrade
```

```
sudo apt-get install gcc
```

- Biên dịch NachOS

1. Mở file Makefile trong thư mục nachos-3.4/code/Makefile, sửa dòng `Make = gmake` thành `Make = make`

2. Biên dịch chương trình: browse vào đường dẫn [nachos-3.4/code](#), chạy lệnh `make`.

- Chạy chương trình test

Nhóm cung cấp các chương trình để test các system calls đã được yêu cầu cài đặt, bao gồm các chương trình `ioChar`, `ioInt`, `ioString` được đặt trong thư mục [nachos-3.4/code/test](#)

Cài đặt các yêu cầu của đề bài:

1. Xử lý các exception

Khi một exception được xử lý, ta sẽ quan tâm đến biến `which` (kiểu `exception`) và giá trị thanh ghi 2 (loại exception). Nói ngắn gọn: biến `which` giúp ta hiểu exception được throw là exception nào, trong khi giá trị thanh ghi 2 là kiểu của exception đó. Ta có các exception sau:

1. `SyscallException` được throw khi một syscall được gọi. Giá trị thanh ghi 2 là giá trị cụ thể của syscall đó.
2. `PageFaultException`
3. `ReadOnlyException`
4. `BusErrorException`
5. `AddressErrorException`
6. `OverflowException`
7. `IllegalInstrException`

Để bắt các exception này, ta cần một cấu trúc `switch..case` để bắt các exception. Các phần xử lý này được viết lại ở hàm `ExceptionHandler` trong file [nachos-3.4/code/userprog/exception.cc](#)

2. Viết lại cấu trúc điều khiển của chương trình để nachos nhận các system call

Rất đơn giản, ta chỉ cần viết từ cấu trúc `if else` trên method

`ExceptionHandler(ExceptionType)` trên file `exception.cc` thành `switch case`, cụ thể như sau:

```

331 ExceptionHandler(ExceptionType which)
332 {
333     int type = machine->ReadRegister(2);
334
335     switch (which) {
336     case NoException:
337         return;
338
339     case SyscallException:
340         // Syscall exceptions goes here.
341         switch (type) {
342         case SC_Halt:
343             DEBUG('a', "Shutdown, initiated by user program.\n");
344             interrupt->Halt();
345             break;
346         case SC_ReadInt:
347             SHandler::ReadInt();
348             break;
349         case SC_PrintInt:
350             SHandler::PrintInt(
351                 machine->ReadRegister(R4));
352             break;
353         case SC_ReadChar:
354             SHandler::ReadChar();
355             break;
356         case SC_PrintChar:
357             SHandler::PrintChar(
358                 machine->ReadRegister(R4));
359             break;
360         case SC_ReadString:
361             SHandler::ReadString(
362                 machine->User2System(
363                     machine->ReadRegister(R4),
364                     machine->ReadRegister(R5)),
365                 machine->ReadRegister(R5));
366             break;
367         case SC_PrintString:
368             SHandler::PrintString(
369                 machine->User2System(
370                     machine->ReadRegister(R4),
371                     MAX_BUFFER_LENGTH));
372             break;
373         default:
374             printf("Unexpected user mode exception %d %d\n", which, type);
375             ASSERT(FALSE);
376         }
377         break;
378
379         // Catching all exceptions
380     case PageFaultException:
381     case ReadOnlyException:
382     case BusErrorException:
383     case AddressErrorException:
384     case OverflowException:
385     case IllegalInstrException:
386         DEBUG('a', "Exception triggered");
387         printf("Exception triggered: %d\n", which);
388         ASSERT(FALSE);
389         interrupt->Halt();
390         break;
391     }
392 }
393 }
394 }
395

```

3. Viết lại hàm tăng program counter

Do trong quá trình làm, nhóm chúng em không tìm thấy hàm tăng program counter mà thầy đã cho, nên bọn em quyết định viết 1 method IncreasePC() trong class Machine ở file machine.cc/machine.h

Tại sao lại là class Machine? Tại vì Machine chứa các thành Register của nachos bằng biến int register[NumTotalRegs], mặt khác, Machine cũng define sẵn các macro của thanh ghi hiện tại (PCReg), thanh ghi tiếp theo (NextPCReg) và thanh ghi trước đó (PrevPCReg)

Thuật toán của method IncreasePC() được cài như sau:

```
275
276  /*
277  Purpose:  - Set the current PC register as previous PC register
278            - Set the next PC register as current PC register
279            - Set the current PC register + 4 as the next PC register
280  */
281  void Machine::IncreasePC() {
282
283      // set the current PC register as previous PC register
284      int tempPCReg = this->ReadRegister(PCReg);
285      this->WriteRegister(PrevPCReg, tempPCReg);
286
287      // Set the next PC register as current PC register
288      tempPCReg = this->ReadRegister(NextPCReg);
289      this->WriteRegister(PCReg, tempPCReg);
290
291      // Set the current PC register + 4 as the next PC register
292      this->WriteRegister(NextPCReg, tempPCReg + 4);
293  }
```

Giải thích ngắn gọn thuật toán:

- Địa chỉ lệnh hiện tại trở thành quá khứ: chuyển dữ liệu từ register[PCReg] về register[PrevPCReg]
- Địa chỉ lệnh tiếp theo trở thành địa chỉ lệnh hiện tại: chuyển dữ liệu từ register[NextPCReg] về register[PCReg]
- Địa chỉ lệnh sau lệnh tiếp theo trở thành địa chỉ lệnh tiếp theo: chuyển dữ liệu từ register[NextPCReg + 4] về register[NextPCReg] (+4 vì nachos sử dụng các thanh ghi lệnh có kích thước 4 bytes)

4. Xử lí các system call

- Cài đặt và khai báo synchcons cho console của nachos

Từ thư mục \nachos\, ta copy các file synchcons.cc và synchcons.h vào thư mục \nachos\nachos-3.4\code\threads

Để file synchcons.cc và synchcons.h được compile, ta cần phải thay đổi 1 chút ở file Makefile: ở thư mục \nachos\nachos-3.4\code\, ta hãy mở file "Makefile.common", và thêm vào như sau:

- Ở dòng có "USERPROG_H=", hãy thêm vào đoạn "../threads/synchcons.h", nhưng đừng để ở hàng cuối cùng (vì '\' là kí tự xuống dòng trong Makefile):

```
USERPROG_H = ../userprog/addrspace.h\  
            ../userprog/bitmap.h\  
            ../filesystem/filesys.h\  
            ../threads/synchcons.h\  
            ../filesystem/openfile.h\  
            ../machine/console.h\  
            ../machine/machine.h\  
            ../machine/mipssim.h\  
            ../machine/translate.h
```

- Làm tương tự như vậy:
 - o Ở dòng có "USERPROG_C=", hãy thêm "../threads/synchcons.cc"

```
USERPROG_C = ../userprog/addrspace.cc\  
            ../userprog/bitmap.cc\  
            ../userprog/exception.cc\  
            ../threads/synchcons.cc\  
            ../userprog/progtest.cc\  
            ../machine/console.cc\  
            ../machine/machine.cc\  
            ../machine/mipssim.cc\  
            ../machine/translate.cc
```

- Ở dòng có "USERPROG_O=" hãy thêm "synchcons.o"

```
USERPROG_O = addrspace.o bitmap.o exception.o progtest.o  
            console.o machine.o synchcons.o \  
            mipssim.o translate.o
```

- Sau đấy tiến hành chạy lệnh make để recompile nachos là xong

Tiếp theo, ta tiến hành khai báo các biến của class SynchConsole trong synchcons.h vào system.h/system.cpp để có thể đồng bộ hóa terminal của Ubuntu với console của Nachos:

Vì SynchConsole chỉ sử dụng khi ta sử dụng con trỏ toàn cục machine, vậy nên ở file system.h, ta tiến hành include các header cần thiết cho con trỏ synchConsole của class SynchConsole (dòng 36)

```

19 // Initialization and cleanup routines
20 extern void Initialize(int argc, char **argv); // Initialization,
21 // called before anything else
22 extern void Cleanup(); // Cleanup, called when
23 // Nachos is done.
24
25 extern Thread *currentThread; // the thread holding the CPU
26 extern Thread *threadToBeDestroyed; // the thread that just finished
27 extern Scheduler *scheduler; // the ready list
28 extern Interrupt *interrupt; // interrupt status
29 extern Statistics *stats; // performance metrics
30 extern Timer *timer; // the hardware alarm clock
31
32 #ifdef USER_PROGRAM
33 #include "machine.h"
34 #include "synchcons.h"
35 extern Machine* machine; // user program memory and registers
36 extern SynchConsole* synchConsole;
37 #endif
38

```

Mặt khác ở system.cpp, ta tiến hành forward declaration, khởi tạo và hủy con trỏ synchConsole khi cần thiết: có 3 vị trí cần phải tùy biến:

Mẹo: Vì synchConsole chỉ sử dụng khi người dùng sử dụng console, khi đẩy macro USER_PROGRAM sẽ được define, do đó ta chỉ cần tùy biến synchConsole tương ứng với các region đã define USER_PROGRAM

```

29
30 #ifdef USER_PROGRAM // requires either FILESYS or FILESYS_STUB
31 Machine *machine; // user program memory and registers
32 SynchConsole *synchConsole;
33 #endif
34
150
151 #ifdef USER_PROGRAM
152 machine = new Machine(debugUserProg); // this must come first
153 synchConsole = new SynchConsole();
154 #endif
155
156 #ifdef FILESYS
180
181 #ifdef USER_PROGRAM
182 delete machine;
183 delete synchConsole;
184 #endif
185

```

- Cài đặt System2User và User2System để có thể chuyển vùng nhớ User về Kernel và ngược lại

Vì chúng em không tìm thấy method System2User và User2System, nên chúng em sẽ copy 2 method này (đã được đề cập ở file "[3] Cách Viết Một SystemCall.pdf") vào class Machine

Tại sao lại là class Machine? Vì System2User và User2System sử dụng method ReadMem() và WriteMem() của class Machine để cài đặt

Em có 2 method lúc này nằm trong file machine.cc như sau

```
220
221  /*
222   Input:  - User space address (int)
223           - Limit of buffer (int)
224   Output: - Buffer (*char)
225   Purpose: Copy buffer from User memory space to system memory space
226
227  */
228
229  char* Machine::User2System(int virtAddr, int limit) {
230      int i; //index
231      int oneChar;
232      char* kernelBuf = NULL;
233
234      kernelBuf = new char[limit + 1]; //need for terminal string
235      if (!kernelBuf)
236      {
237          return kernelBuf;
238      }
239
240      memset(kernelBuf, 0, limit + 1);
241
242      for (int i = 0; i < limit; ++i) {
243          this->ReadMem(virtAddr + i, 1, &oneChar);
244          kernelBuf[i] = (char)oneChar;
245          if (0 == oneChar) //null terminated appeared
246          {
247              break;
248          }
249      }
250
251      return kernelBuf;
252  }
253
254  /*
255   Input:  - User space address (int)
256           - Limit of buffer (int)
257           - Buffer (char[])
258   Output: - Number of bytes copied (int)
259   Purpose: Copy buffer from System memory space to User memory space
260  */
261  int Machine::System2User(int virtAddr, int len, char* buffer) {
262      if (len < 0) return -1;
263      if (0 == len) return len;
264      int i = 0;
265      int oneChar = 0;
266      do {
267          oneChar = (int)buffer[i];
268          this->WriteMem(virtAddr + i, 1, oneChar);
269          ++i;
270      } while (i < len && oneChar != 0);
271
272      return i;
273  }
274
```

- Forward declaration cho các system call để user program có thể call những system call này

Đầu tiên, ta phải define các macro của các system call cần cài đặt, để sau này có thể giao tiếp với MIPS của nachos để call syscall

Ta thực hiện define ở file syscall.h

```

7  *
8  * Copyright (c) 1992-1993 The Regents of the University of California.
9  * All rights reserved. See copyright.h for copyright notice and limitation
10 * of liability and disclaimer of warranty provisions.
11 */
12
13 #ifndef SYSCALLS_H
14 #define SYSCALLS_H
15
16 #include "copyright.h"
17
18 /* system call codes -- used by the stubs to tell the kernel which system call
19  * is being asked for
20  */
21 #define SC_Halt      0
22 #define SC_Exit      1
23 #define SC_Exec      2
24 #define SC_Join      3
25 #define SC_Create    4
26 #define SC_Open      5
27 #define SC_Read      6
28 #define SC_Write     7
29 #define SC_Close     8
30 #define SC_Fork      9
31 #define SC_Yield     10
32
33 /*User defined*/
34 #define SC_ReadInt 11
35 #define SC_PrintInt 12
36 #define SC_ReadChar 13
37 #define SC_PrintChar 14
38 #define SC_ReadString 15
39 #define SC_PrintString 16
40
41 #ifndef IN_ASM

```

Tiếp theo, ta thực hiện forward declaration các method của system call trên chính file syscall.h này, để user program có thể gọi các method này để sử dụng system call

```

121  /* User-level thread operations: Fork and Yield. To allow multiple
122  * threads to run within a user program.
123  */
124
125  /* Fork a thread to run a procedure ("func") in the *same* address space
126  * as the current thread.
127  */
128  void Fork(void (*func)());
129
130  /* Yield the CPU to another runnable thread, whether in this address space
131  * or not.
132  */
133  void Yield();
134
135
136  /******
137  /******
138  /*USER DEFINE PROTOTYPES*/
139  /******
140  /******
141
142  int ReadInt();
143  void PrintInt(int number);
144  char ReadChar();
145  void PrintChar(char character);
146  void ReadString(char* buffer, int length);
147  void PrintString(char* buffer);
148
149  #endif /* IN_ASM */
150
151  #endif /* SYSCALL_H */
152

```

Tiếp theo, ta cần phải viết thêm 1 số lệnh MIPS để nachos có thể call những system call mà ta đã declaration

Các lệnh MIPS này sẽ được thêm vào cuối file start.c và start.s

Dưới đây là mẫu của 1 đoạn lệnh cho system call ReadInt

```

132
133  /****** User define *****/
134  .globl ReadInt
135  .ent ReadInt
136  ReadInt :
137      addiu $2, $0, SC_ReadInt
138      syscall
139      j $31
140  .end ReadInt
141

```

Hướng dẫn cách thêm đoạn lệnh MIPS

- Tất cả những system call khác đều có cùng format như đoạn lệnh mẫu ở trên
- Giả sử ta cần thêm system call a, và đã define macro SC_a, ta chỉ cần thay ở đoạn lệnh trên bằng:
 - o ReadInt thay bằng a
 - o SC_ReadInt thay bằng SC_a

Quan trọng: nhớ phải thêm đoạn lệnh MIPS ở cả file start.c và start.s

- Cài đặt system call `int ReadInt()`

Input của system call này chính là buffer đọc được từ bản thân console

Để lấy được buffer này, ta cần khởi tạo 1 mảng char `buffer[MAX_BUFFER_LENGTH + 1]` với `MAX_BUFFER_LENGTH` được em để mặc định là 255, sau đấy, sử dụng method `Read(buffer, MAX_BUFFER_LENGTH)` từ object `synchConsole` (mà ta đã forwarding declaration ở mục trước) của class `SynchConsole` để load tối đa 255 kí tự vào biến `buffer`. Đồng thời, hàm `Read(buffer, MAX_BUFFER_LENGTH)` cũng trả ra số lượng kí tự thật sự đọc được, ta gán giá trị này vào biến `int numberOfReadChar`

Sử dụng buffer đọc được và biến `numberOfReadChar`, ta tiến hành điều chế buffer này để trả ra số nguyên đọc được, method hỗ trợ việc này tên là `int StringToNumber(char* buffer, int numBytes)`, trong đó việc xử lí diễn ra như sau:

- `StringToNumber()` sẽ loại bỏ **TOÀN BỘ** các kí tự dẫn cách ' ' ở cả trước và sau buffer (giống hàm `Trim()` trong Java)
- `StringToNumber()` sẽ trả ra 0 nếu như buffer không phải là 1 số nguyên/thực
- `StringToNumber()` sẽ trả ra phần nguyên của số thực nếu buffer là 1 số thực
- `StringToNumber()` hoạt động hoàn toàn bình thường đối với số âm

Sau khi điều chế xong, giá trị trả về của `StringToNumber()` sẽ được gán vào biến `int value`. Sử dụng method `WriteRegister(2, value)` để ghi giá trị `value` vào thanh ghi số 2 (Vì theo description của nachos, thanh ghi số 2 trả ra giá trị trả về của system call)

Cuối cùng, sử dụng method `IncreasePC()` mà ta đã cài để tăng program counter, tiếp tục chương trình.

- Cài đặt system call `void PrintInt(int number)`

Input của system call này chính là dữ liệu trên thanh ghi số 4 (vì theo description của Nachos, thanh ghi số 4 chính là tham số thứ 1 của system call)

Ta tiến hành điều chế biến `int number` tham số thành 1 chuỗi giống với giá trị `number`, hàm xử lí việc này tên là `NumberToString(int number, int& length)`, trong đó, `length` là 1 tham chiếu output dùng để đếm số lượng kí tự của số `number`. Việc xử lí diễn ra như sau

- Nếu `number = 0`, ngay lập tức, `length = 1` và `NumberToString()` trả về "0"
- Nếu `number` là số âm, thì 1 biến boolean `isNegative = true`
- Nếu `number` là số âm, thì `number = number - 1`, từ đây về cuối ta chỉ sử dụng `number` như giá trị tuyệt đối
- Gán `length = 0`

- Ta tiến hành đếm số lượng chữ số của number bằng cách chia liên tục cho 10, nếu isNegative = true, thì số lượng chữ số sẽ tăng thêm 1 => length cũng tăng theo
- Khởi tạo 1 mảng buffer bằng số lượng chữ số đã đọc được.
- Nếu isNegative = true, buffer[0] = '-', length tăng 1 đơn vị
- Ta tiến hành duyệt các chữ số của number theo thứ tự từ phải sang trái bằng cách chia liên tục cho 10, sử dụng toán tử % 10 để lấy từng kí tự của number, sau đấy gán cho buffer
- Kết thúc việc gán giá trị, ta gán kí tự cuối của buffer cho null-terminated
- Trả về giá trị buffer

Giá trị trả về của NumberToString(int number, int& length) sẽ được gán vào biến char* buffer (mặt khác còn có iến length là chiều dài của buffer cũng được tham chiếu trả về), để có thể xuất buffer này lên console, ta sử dụng method Write(buffer, length) của object synchConsole của class SynchConsole để in length kí tự từ mảng buffer.

Cuối cùng, sử dụng method IncreasePC() mà ta đã cài để tăng program counter, tiếp tục chương trình.

- **Cài đặt system call char ReadChar()**

Input của system call này chính là buffer đọc được từ bản thân console

Để lấy được buffer này, ta cần khởi tạo 1 mảng char buffer[MAX_BUFFER_LENGTH + 1] với MAX_BUFFER_LENGTH được em để mặc định là 255, sau đấy, sử dụng method Read(buffer, MAX_BUFFER_LENGTH) từ object synchConsole (mà ta đã forwarding declaration ở mục trước) của class SynchConsole để load tối đa 255 kí tự vào biến buffer. Đồng thời, hàm Read(buffer, MAX_BUFFER_LENGTH) cũng trả ra số lượng kí tự thật sự đọc được, ta gán giá trị này vào biến int numberOfReadChar

Ta tiến hành kiểm tra giá trị của numberOfReadChar:

- Nếu numberOfReadChar <= 0, tức là có sự cố xảy ra trong quá trình read buffer từ console bằng method Read(buffer, MAX_BUFFER_LENGTH), do đó, ngay lập tức, ta ném ra 1 thông báo, và sử dụng method WriteRegister(2, 0) của object machine từ biến Machine để ghi thẳng giá trị trả về của system call này là 0, khi xảy ra lỗi, tiếp tục chương trình, ta tăng program counter bằng method IncreasePC() và kết thúc system call

Ngược lại, nếu numberOfReadChar > 0, ta dùng WriteRegister(2, buffer[0]) để ghi trực tiếp kí tự đầu tiên của buffer đọc được làm output cho system call

=> Với input là 1 kí tự hay 1 xâu, luôn luôn trả ra kí tự đầu tiên

Cuối cùng, tăng program counter bằng method IncreasePC() và kết thúc system call

- Cài đặt system call void PrintChar(char character)

Input của system call này chính là dữ liệu trên thanh ghi số 4 (vì theo description của Nachos, thanh ghi số 4 chính là tham số thứ 1 của system call)

Ta sử dụng method Write(&character, 1) của object synchConsole của class SynchConsole để in ra 1 kí tự từ con trỏ &character ra console

Cuối cùng, tăng program counter bằng method IncreasePC() và kết thúc system call

- Cài đặt system call void ReadString(char* buffer, int length)

Input của system call này chính là:

- Buffer: Vùng nhớ chuyển từ User sang Kernel bằng method User2System(int virtualAddr, int limit), trong đó, 2 tham số của User2System sẽ là:
 - + virtualAddr: dữ liệu trên thanh ghi số 4 (tham số đầu tiên của system call)
 - + limit: dữ liệu trên thanh ghi số 5 (tham số thứ 2 của system call)
- Length: dữ liệu trên thanh ghi số 5 (chiều dài cấp phát cho mảng)

Ta sử dụng method Read(buffer, length) từ object synchConsole (mà ta đã forwarding declaration ở mục trước) của class SynchConsole để đọc đúng length kí tự từ buffer trên console về chính bản thân buffer đã được cung cấp vùng nhớ

Sau khi đọc xong console, ta chuyển dữ liệu đọc được trên buffer về vùng nhớ của User bằng System2User(int virtAddr, int len, char* buffer) với

- VirtAddr: dữ liệu trên thanh ghi số 4 (giống với ban đầu)
- Len: chính là length đã truyền vào
- Buffer: buffer đã được đọc bằng console

Cuối cùng, tăng program counter bằng method IncreasePC() và kết thúc system call

- Cài đặt system call void PrintString(char* buffer)

Input của system call này chính là:

- Buffer: Vùng nhớ chuyển từ User sang Kernel bằng method User2System(int virtualAddr, int limit), trong đó, 2 tham số của User2System sẽ là:
 - + virtualAddr: dữ liệu trên thanh ghi số 4 (tham số đầu tiên của system call)

+ limit: MAX_BUFFER_LENGTH (255) vì ta không biết chính xác được sẽ phải in ra bao nhiêu kí tự, nên bọn em đặt mặc định 1 lần in ra tối đa 255 kí tự

Ta dùng 1 biến int length để lưu giữ chiều dài thật sự của buffer đọc được, bằng cách duyệt cả mảng buffer cho tới khi gặp null-terminated.

Nếu length = MAX_BUFFER_LENGTH, tức là mảng cần đọc quá dài, không thể đọc hết, ta ném ra 1 thông báo, tăng program counter và kết thúc chương trình.

Nếu length < MAX_BUFFER_LENGTH, tức là mảng hợp lệ, ta tiến hành ghi buffer này ra console bằng method Write(buffer, length) của object synchConsole của class SynchConsole để in ra toàn bộ buffer.

Cuối cùng, tăng program counter bằng method IncreasePC() và kết thúc system call

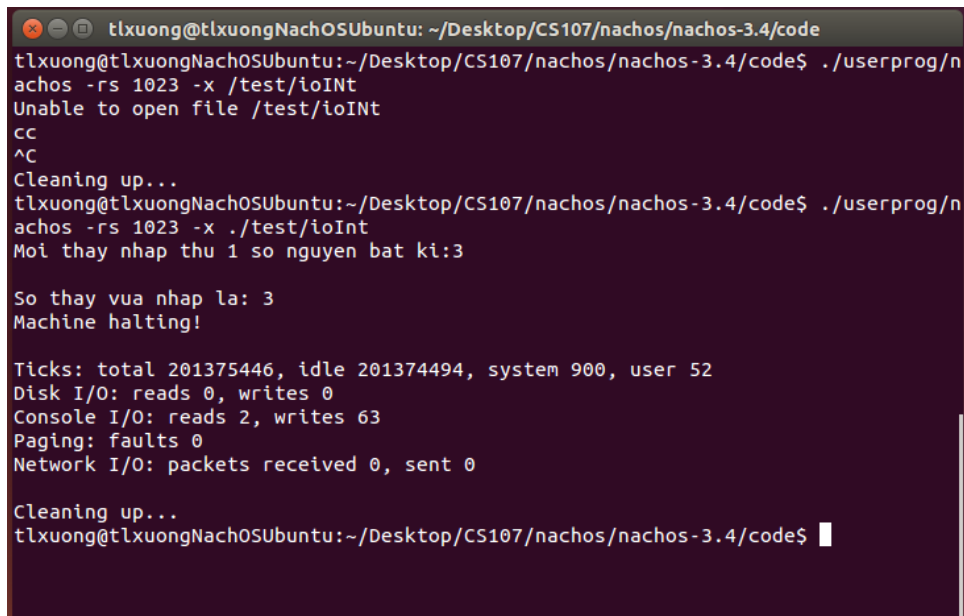
- Test thử các system call đã được cài
 - Test PrintInt/ReadInt

Chương trình test cho ReadInt/PrintInt có tên là ioInt, khởi chạy bằng cách:

- B1: Trên terminal, cd đến folder /nachos/nachos-3.4/code/
- B2: Trên terminal, nhấn lệnh: ./userprog/nachos -rs 1023 -x ./test/ioInt

Các trường hợp test để demo:

- Số nhập vào là số nguyên dương: input = "3"



```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/n
achos -rs 1023 -x ./test/ioInt
Unable to open file ./test/ioInt
cc
^C
Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/n
achos -rs 1023 -x ./test/ioInt
Moi thay nhap thu 1 so nguyen bat ki:3

So thay vua nhap la: 3
Machine halting!

Ticks: total 201375446, idle 201374494, system 900, user 52
Disk I/O: reads 0, writes 0
Console I/O: reads 2, writes 63
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

- Số nhập vào là số nguyên âm: input = "-3"

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
Machine halting!

Ticks: total 201375446, idle 201374494, system 900, user 52
Disk I/O: reads 0, writes 0
Console I/O: reads 2, writes 63
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/n
achos -rs 1023 -x ./test/ioInt
Moi thay nhap thu 1 so nguyen bat ki:-3

So thay vua nhap la: -3
Machine halting!

Ticks: total 94078236, idle 94077294, system 890, user 52
Disk I/O: reads 0, writes 0
Console I/O: reads 3, writes 64
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

- Số nhập vào là số thực: input = "3.123"

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
Machine halting!

Ticks: total 94078236, idle 94077294, system 890, user 52
Disk I/O: reads 0, writes 0
Console I/O: reads 3, writes 64
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/n
achos -rs 1023 -x ./test/ioInt
Moi thay nhap thu 1 so nguyen bat ki:3.123

So thay vua nhap la: 3
Machine halting!

Ticks: total 102475026, idle 102474054, system 920, user 52
Disk I/O: reads 0, writes 0
Console I/O: reads 6, writes 63
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

- Số nhập vào là số thực âm, có khoảng trắng trước lẫn sau: input = " -13.321 "


```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
Machine halting!

Ticks: total 102475026, idle 102474054, system 920, user 52
Disk I/O: reads 0, writes 0
Console I/O: reads 6, writes 63
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/n
achos -rs 1023 -x ./test/ioInt
Moi thay nhap thu 1 so nguyen bat ki: -13.321

So thay vua nhap la: -13
Machine halting!

Ticks: total 614190156, idle 614189044, system 1060, user 52
Disk I/O: reads 0, writes 0
Console I/O: reads 15, writes 65
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

- Số nhập vào không phải là số: input = "abc"

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
Machine halting!

Ticks: total 614190156, idle 614189044, system 1060, user 52
Disk I/O: reads 0, writes 0
Console I/O: reads 15, writes 65
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/n
achos -rs 1023 -x ./test/ioInt
Moi thay nhap thu 1 so nguyen bat ki:abc

So thay vua nhap la: 0
Machine halting!

Ticks: total 195770836, idle 195769874, system 910, user 52
Disk I/O: reads 0, writes 0
Console I/O: reads 4, writes 63
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

o Test PrintChar/ReadChar

Chương trình test cho ReadChar/PrintChar có tên là ioChar, khởi chạy bằng cách:

- B1: Trên terminal, cd đến folder /nachos/nachos-3.4/code/
- B2: Trên terminal, nhấn lệnh: ./userprog/nachos -rs 1023 -x ./test/ioChar

Các trường hợp test:

- Ký tự nhập vào là 1 ký tự: input = "c"

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
Machine halting!

Ticks: total 195770836, idle 195769874, system 910, user 52
Disk I/O: reads 0, writes 0
Console I/O: reads 4, writes 63
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/n
achos -rs 1023 -x ./test/ioChar
Moi thay nhap thu 1 ki tu bat ki:q

Ki tu thay vua nhap la: q
Machine halting!

Ticks: total 165885357, idle 165884444, system 860, user 53
Disk I/O: reads 0, writes 0
Console I/O: reads 2, writes 62
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

- Kí tự nhập vào là 1 chuỗi: input = "abc def 123"

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
Machine halting!

Ticks: total 165885357, idle 165884444, system 860, user 53
Disk I/O: reads 0, writes 0
Console I/O: reads 2, writes 62
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/n
achos -rs 1023 -x ./test/ioChar
Moi thay nhap thu 1 ki tu bat ki:abc def 123

Ki tu thay vua nhap la: a
Machine halting!

Ticks: total 157543877, idle 157542834, system 990, user 53
Disk I/O: reads 0, writes 0
Console I/O: reads 12, writes 62
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

- Không nhập gì cả, chỉ nhấn enter

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
Machine halting!

Ticks: total 157543877, idle 157542834, system 990, user 53
Disk I/O: reads 0, writes 0
Console I/O: reads 12, writes 62
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/n
achos -rs 1023 -x ./test/ioChar
Moi thay nhap thu 1 ki tu bat ki:

Ki tu thay vua nhap la:
Machine halting!

Ticks: total 60916317, idle 60915354, system 910, user 53
Disk I/O: reads 0, writes 0
Console I/O: reads 1, writes 62
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

○ Test PrintChar/ReadChar

Chương trình test cho ReadString/PrintString có tên là ioString, khởi chạy bằng cách:

- B1: Trên terminal, cd đến folder /nachos/nachos-3.4/code/
- B2: Trên terminal, nhấn lệnh: ./userprog/nachos -rs 1023 -x ./test/ioString

Các trường hợp test:

- Xâu kí tự nhập vào là 1 chuỗi: input = "abc def 123"

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
Machine halting!

Ticks: total 60916317, idle 60915354, system 910, user 53
Disk I/O: reads 0, writes 0
Console I/O: reads 1, writes 62
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/n
achos -rs 1023 -x ./test/ioString
Moi thay nhap thu 1 chuoi bat ki:abc def 123

Chuoi nhap la: abc def 123
Machine halting!

Ticks: total 143014304, idle 143013243, system 1010, user 51
Disk I/O: reads 0, writes 0
Console I/O: reads 12, writes 64
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

- Không nhập gì cả, chỉ nhấn enter

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
Machine halting!

Ticks: total 143014304, idle 143013243, system 1010, user 51
Disk I/O: reads 0, writes 0
Console I/O: reads 12, writes 64
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/n
achos -rs 1023 -x ./test/ioString
Mời thay nhập thu 1 chuỗi bat ki:

Chuỗi nhập là:
Machine halting!

Ticks: total 36106874, idle 36106053, system 770, user 51
Disk I/O: reads 0, writes 0
Console I/O: reads 1, writes 53
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

5. Viết các chương trình trên nachos

- Tóm tắt một số điều kiện cần
 - Các chương trình ta cần viết phải nằm trong folder /test/
 - Các chương trình nhóm đã làm trong đồ án này:
 - o Ba chương trình nhóm viết thêm vào để test các system call: ioInt, ioChar, ioString
 - o Chương trình help (yêu cầu đồ án)
 - o Chương trình ascii (yêu cầu đồ án)
 - o Chương trình sort (yêu cầu đồ án)
- Các bước chuẩn bị để viết 1 chương trình trên nachos
 - Hãy giả sử, ta cần viết 1 chương trình có tên là abc
 - Ta sẽ tạo ra 1 file tên là abc.c, viết code trên đấy, rồi đặt nó trong folder /test/
 - Sau đấy, ta cần phải setup để abc.c có thể được compile thành công bằng makefile do nachos cung cấp, ta cần phải làm như sau:
 - o Mở Makefile trong \nachos\nachos-3.4\code\test
 - o Thêm "abc" vào dòng "all: halt shell matmult sort"

```
all: halt shell matmult sort abc
```

- o Thêm vào dưới cùng của file 1 đoạn sau (y hệt như những halt/shell/matmult/sort khác):

```

abc.o: abc.c
      $(CC) $(CFLAGS) -c abc.c
abc: abc.o start.o
      $(LD) $(LDFLAGS) start.o abc.o -o abc.coff
      ../bin/coff2noff abc.coff abc

```

- Trên terminal, cd về folder /nachos/nachos-3.4/code, chạy lệnh make để recompile lại nachos
- Để chạy được chương trình abc đã compile, cực kì đơn giản, ta làm như sau (đã đề cập ở phần chạy test system call):
 - B1: Trên terminal, cd đến folder /nachos/nachos-3.4/code/
 - B2: Trên terminal, nhấn lệnh: ./userprog/nachos -rs 1023 -x ./test/abc
- Như vậy, đây là phương pháp để có thể cài đặt và chạy 1 user program trên nachos, tất cả các user program chúng ta cần cài và chạy sẽ cần có những bước giống y hệt như phía trên.
- [Viết chương trình help](#)
- Chương trình help được viết trên file "help.c" đặt trong folder /nachos/nachos-3.4/code/test/
- Không có gì phức tạp ở chương trình này, chỉ đơn giản là in ra thông tin nhóm và chi tiết 2 chương trình ascii và sort ta sắp phải cài (ngoài ra còn có thông tin về 3 chương trình ioInt/ioChar/ioString để test các system call)

```

1  #include "syscall.h"
2
3  int main() {
4      PrintString("- Cac thanh vien trong nhom:\n");
5      PrintString("\t+ 19120338 - Tran Hoang Quan\n");
6      PrintString("\t+ 19120383 - Huynh Tan Tho\n");
7      PrintString("\t+ 19120407 - Lam Hai Trieu\n");
8      PrintString("\t+ 19120426 - Phan Dang Diem Uyen\n");
9      PrintString("\t+ 19120469 - Su Nhat Dang\n");
10
11     PrintString("- Ngoai 2 chuong trinh ascii va sort, ");
12     PrintString("bon em con lam them 3 chuong trinh de test 6 system call tuong ung, do la:\n");
13     PrintString("\t+ ioInt: test PrintInt/ReadInt\n");
14     PrintString("\t+ ioChar: test PrintChar/ReadChar\n");
15     PrintString("\t+ ioString: test PrintString/ReadString\n");
16     PrintString("- Ca 3 chuong trinh nay deu nam trong folder ./test/\n\n");
17
18     PrintString("- Gioi thieu chuong trinh ascii:\n");
19     PrintString("Chuong trinh ascii cho phep xuat ra gia tri Dec cua 1 ki tu va ki tu trong pham vi tu [32; 126], ");
20     PrintString("vi day la vung ma console co the xuat ra ki tu ascii co the nhin thay duoc\n\n");
21
22     PrintString("- Gioi thieu chuong trinh sort:\n");
23     PrintString("Chuong trinh sort cho phep nguoi dung nhap kieu sort (tang/giam), so luong phan tu (<=100) va ");
24     PrintString("tung phan tu trong mang can sort. Chuong trinh se xuat ra man hinh ket qua cua mang sau khi sort\n");
25
26     Halt();
27 }

```

- Làm theo hướng dẫn cách để compile và chạy 1 chương trình trên trên nachos như ta đã nói ở trên, ta dễ dàng chạy được chương trình test
- Kết quả chạy chương trình help như sau:

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code

tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/help
- Các thành viên trong nhóm:
  + 19120338 - Tran Hoang Quan
  + 19120383 - Huynh Tan Tho
  + 19120407 - Lam Hai Trieu
  + 19120426 - Phan Dang Diem Uyen
  + 19120469 - Su Nhat Dang
- Ngoài 2 chương trình ascii và sort, bọn em còn làm thêm 3 chương trình để test 6 system call tương ứng, đó là:
  + ioInt: test PrintInt/ReadInt
  + ioChar: test PrintChar/ReadChar
  + ioString: test PrintString/ReadString
- Cả 3 chương trình này đều nằm trong folder ./test/

- Giới thiệu chương trình ascii:
Chương trình ascii cho phép xuất ra giá trị Dec của 1 kí tự và kí tự trong phạm vi từ [32; 126], vì đây là vùng mà console có thể xuất ra kí tự ascii có thể nhìn thấy được

- Giới thiệu chương trình sort:
Chương trình sort cho phép người dùng nhập kiểu sort (tăng/giảm), số lượng phần tử (<=100) và từng phần tử trong mảng cần sort. Chương trình sẽ xuất ra màn hình kết quả của mảng sau khi sort
Machine halting!

Ticks: total 100348, idle 89800, system 10410, user 138
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 898
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

- Viết chương trình ascii
 - Chương trình ascii được viết trên file "ascii.c" đặt trong folder /nachos/nachos-3.4/code/test/
 - Không có gì phức tạp ở chương trình này chỉ đơn giản là dùng 1 vòng lặp để in ra các kí tự có giá trị ascii thập phân trong đoạn [32; 126]
 - Q: Tại sao lại là [32; 126]?
 - A: Vì trong khoảng [32; 126] là toàn bộ những kí tự **có thể in ra được** khi xuất ra màn hình console

```

1  #include "syscall.h"
2  #define MAX_ASCII_VISIBLE 126
3  #define MIN_ASCII_VISIBLE 32
4
5  int main() {
6      int i;
7      PrintString("Sau day la bang ma ASCII cho cac ki tu co the nhin thay duoc (la cac ki tu nam trong khoang [32, 126])\n");
8      for (i = MIN_ASCII_VISIBLE; i <= MAX_ASCII_VISIBLE; ++i) {
9          PrintInt(i);
10         PrintChar(':');
11         PrintChar((char)i);
12         PrintChar('\n');
13     }
14
15     Halt();
16 }

```

- Kết quả chạy chương trình ascii như sau:



```

tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/nachos -rs
./test/ascii
Sau day la bang ma ASCII cho cac ki tu co the nhin thay duoc (la cac ki tu nam trong khoan
26])
32:
33:!
34:"
35:#
36:$
37:%
38:&
39:'
40:(
41:)
42:*
43:+
44:,
45:-
46:.
47:/
48:0
49:1
50:2
51:3
52:4
53:5
54:6
55:7

```



```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
55:7
56:8
57:9
58::
59:;
60:<
61:=
62:>
63:?
64:@
65:A
66:B
67:C
68:D
69:E
70:F
71:G
72:H
73:I
74:J
75:K
76:L
77:M
78:N
79:O
80:P
81:Q
82:R
```

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
82:R
83:S
84:T
85:U
86:V
87:W
88:X
89:Y
90:Z
91:[
92:\
93:]
94:^
95:_
96:`
97:a
98:b
99:c
100:d
101:e
102:f
103:g
104:h
105:i
106:j
107:k
108:l
109:m
```

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
109:m
110:n
111:o
112:p
113:q
114:r
115:s
116:t
117:u
118:v
119:w
120:x
121:y
122:z
123:{
124:|
125:}
126:~
Machine halting!

Ticks: total 79853, idle 60600, system 15710, user 3543
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 606
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

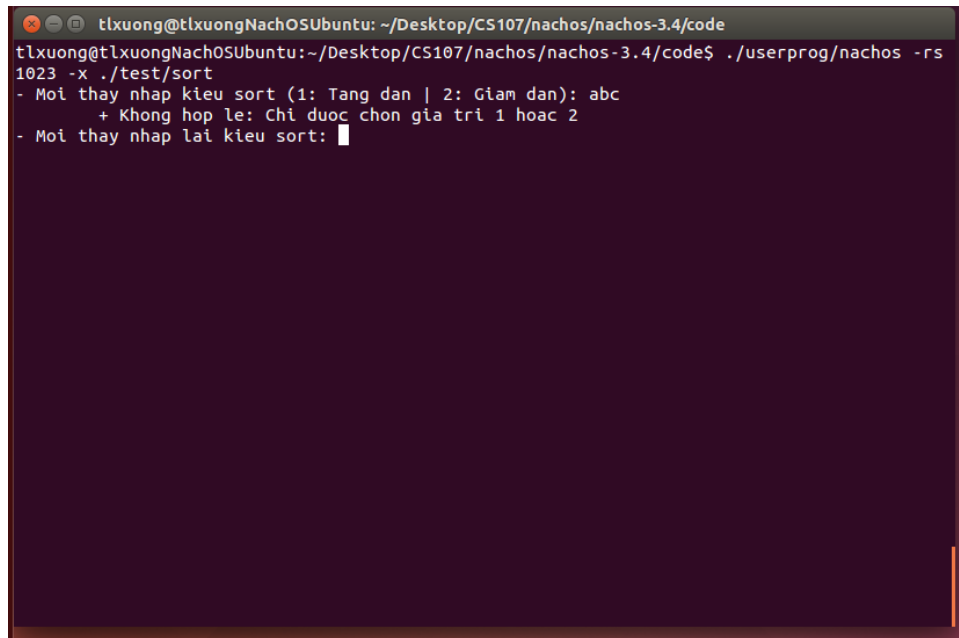
- **Viết chương trình sort**

- Chương trình sort được viết trên file "sort.c" đặt trong folder /nachos/nachos-3.4/code/test/
- Chương trình sort yêu cầu người dùng nhập như sau:
 - o Đầu tiên, chương trình sẽ yêu cầu người dùng nhập kiểu sort:

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/nachos -rs
1023 -x ./test/sort
- Mời thay nhập kiểu sort (1: Tang dan | 2: Giam dan):
```

- Để sort tăng dần: Nhập "1" **hoặc** 1 với khoảng trắng đầu và cuối (ví dụ: " 1 ") **hoặc** số thực có phần nguyên là 1 (ví dụ: "1.52124"). Bởi vì giá trị nhập sort tuân theo cách hành xử của system call ReadInt

- Để sort giảm dần: Nhập "2" **hoặc** 2 với khoảng trắng đầu và cuối (ví dụ: " 2 ") **hoặc** số thực có phần nguyên là 2 (ví dụ: "2.52124"). Bởi vì giá trị nhập sort tuân theo cách hành xử của system call ReadInt
- Nếu người dùng nhập bất kì cái gì ngoài những cái đã nêu trên, hoặc đơn giản là nhấn enter, 1 thông báo lỗi sẽ được hiện ra, và kèm theo thông báo ép người dùng nhập lại. Người dùng buộc phải nhập lại. Chương trình sẽ tiếp tục ném thông báo này cho đến khi nào người dùng nhập các giá trị đã được cho phép



```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/nachos -rs
1023 -x ./test/sort
- Mọi thay nhập kiểu sort (1: Tăng dần | 2: Giảm dần): abc
+ Không hợp lệ: Chỉ được chọn giá trị 1 hoặc 2
- Mọi thay nhập lại kiểu sort: █
```

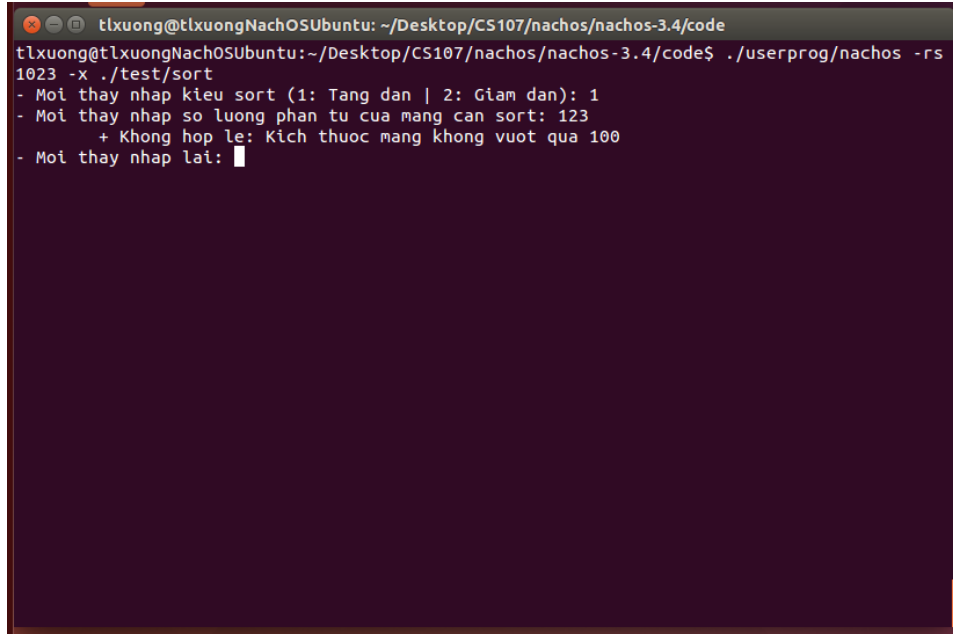
- Tiếp theo, chương trình sẽ yêu cầu người dùng nhập kiểu số lượng phần tử trong mảng. Số lượng phần tử phải là số dương nhỏ hơn hoặc bằng 100:

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/nachos -rs
1023 -x ./test/sort
- Mọi thay nhập kiểu sort (1: Tang dan | 2: Giam dan): abc
  + Không hợp lệ: Chỉ được chọn giá trị 1 hoặc 2
- Mọi thay nhập lại kiểu sort: 1
- Mọi thay nhập số lượng phần tử của mảng cần sort: █
```

- Nếu người dùng nhập một số bé hơn hoặc bằng 0 vào số lượng phần tử, 1 thông báo lỗi sẽ được hiện ra, và kèm theo thông báo ép người dùng nhập lại. Chương trình sẽ tiếp tục ném thông báo này cho đến khi nào người dùng nhập các giá trị đã được cho phép

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/nachos -rs
1023 -x ./test/sort
- Mọi thay nhập kiểu sort (1: Tang dan | 2: Giam dan): 1
- Mọi thay nhập số lượng phần tử của mảng cần sort: -3
  + Không hợp lệ: Số lượng phần tử phải là số dương!
- Mọi thay nhập lại:
```

- Nếu người dùng nhập một số lớn hơn 100 vào số lượng phần tử, 1 thông báo lỗi sẽ được hiện ra, và kèm theo thông báo ép người dùng nhập lại. Chương trình sẽ tiếp tục ném thông báo này cho đến khi nào người dùng nhập các giá trị đã được cho phép



```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/nachos -rs
1023 -x ./test/sort
- Moi thay nhap kieu sort (1: Tang dan | 2: Giam dan): 1
- Moi thay nhap so luong phan tu cua mang can sort: 123
+ Khong hop le: Kich thuc mang khong vuot qua 100
- Moi thay nhap lai: 
```

- Tiếp theo, chương trình sẽ yêu cầu người dùng nhập từng phần tử, cách nhập phần tử tuân theo behaviour mà ta đã cài đặt cho system call ReadInt
 - Nếu nhập 1 chuỗi số, nhưng có khoảng trắng trước và sau, thì hàm vẫn tự động đọc chuỗi số 1 cách bình thường, tự động bỏ qua các khoảng trắng (ví dụ " 12 ", giá trị đọc được sẽ là 12)
 - Nếu nhập 1 số thực, thì sẽ đọc được phần nguyên của số thực (ví dụ "12.123", giá trị đọc được sẽ là 12)
 - Nếu nhập 1 số âm, thì vẫn đọc ra kết quả bình thường (ví dụ "-12", giá trị đọc được sẽ là -12)
 - Kết hợp với 3 kiểu trên, vẫn tuân thủ đầy đủ cả 3 quy tắc (ví dụ "- 12.123 ", giá trị đọc được sẽ là -12)
 - Nếu nhập bất kì 1 kí tự chữ nào, hoặc là số không hợp lệ (ví dụ " 1 23", hay "1. 12"), tự động trả về 0 (như yêu cầu của đề án)
- Cuối cùng, chương trình sẽ tự động chạy ra kết quả sau khi sort:

- Kết quả chạy chương trình sort như sau:
 - o Sắp xếp tăng dần, tất cả các số đều bình thường

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/nachos -rs
1023 -x ./test/sort
- Mọi thay nhập kiểu sort (1: Tăng dần | 2: Giảm dần): 1
- Mọi thay nhập số lượng phần tử của mảng cần sort: 3
a[0] = 3
a[1] = 2
a[2] = 1
- Mảng sau khi đã sắp xếp bằng bubble sort là:
1 2 3
Machine halting!

Ticks: total 360869367, idle 360865800, system 2920, user 647
Disk I/O: reads 0, writes 0
Console I/O: reads 10, writes 192
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

- o Sắp xếp giảm dần, tất cả các số đều bình thường

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/nachos -rs
1023 -x ./test/sort
- Mọi thay nhập kiểu sort (1: Tăng dần | 2: Giảm dần): 2
- Mọi thay nhập số lượng phần tử của mảng cần sort: 3
a[0] = 1
a[1] = 2
a[2] = 3
- Mảng sau khi đã sắp xếp bằng bubble sort là:
3 2 1
Machine halting!

Ticks: total 299062383, idle 299058830, system 2900, user 653
Disk I/O: reads 0, writes 0
Console I/O: reads 10, writes 192
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$
```

- Input lỗi: kiểu sort không hợp lệ

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/nachos -rs
1023 -x ./test/sort
- Mọi thay nhập kiểu sort (1: Tăng dần | 2: Giảm dần): abc
  + Không hợp lệ: Chỉ được chọn giá trị 1 hoặc 2
- Mọi thay nhập lại kiểu sort:
```

- Input lỗi: số lượng phần tử bé hơn hoặc bằng 0

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/sort
- Mọi thay nhập kiểu sort (1: Tăng dần | 2: Giảm dần): 2
- Mọi thay nhập số lượng phần tử của mảng cần sort: -17
  + Không hợp lệ: Số lượng phần tử phải là số dương!
- Mọi thay nhập lại: █
```

- Input lỗi: số lượng phần tử lớn hơn 100

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/sort
- Moi thay nhap kieu sort (1: Tang dan | 2: Giam dan): 2
- Moi thay nhap so luong phan tu cua mang can sort: 123
  + Khong hop le: Kich thuc mang khong vuot qua 100
- Moi thay nhap lai: 
```

- Các phần tử nhập vào của mảng không phải là kiểu đúng format thông thường (số thực âm có khoảng cách trước sau, số nguyên sai format, chuỗi, không nhập gì cả)

```
tlxuong@tlxuongNachOSUbuntu: ~/Desktop/CS107/nachos/nachos-3.4/code
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ ./userprog/nachos -rs 1023 -x ./test/sort
- Moi thay nhap kieu sort (1: Tang dan | 2: Giam dan): 2
- Moi thay nhap so luong phan tu cua mang can sort: 4
a[0] = -14.123
a[1] = -3 .14
a[2] = abc
a[3] = 
- Mang sau khi da sap xep bang bubble sort la:
0 0 0 -14
Machine halting!

Ticks: total 1207856929, idle 1207852614, system 3400, user 915
Disk I/O: reads 0, writes 0
Console I/O: reads 28, writes 205
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
tlxuong@tlxuongNachOSUbuntu:~/Desktop/CS107/nachos/nachos-3.4/code$ 
```


Tài liệu tham khảo

- [1] File được giảng viên cung cấp: Bien dich va Cai dat Nachos.pdf
- [2] File được giảng viên cung cấp: Giao tiep giua HDH Nachos va chuong trinh nguoi dung.pdf
- [3] File được giảng viên cung cấp: Cach Viet Mot SystemCall.pdf
- [4] File được giảng viên cung cấp: Cach Them 1 Lop Vao Nachos.pdf
- [5] https://courses.cs.duke.edu/fall03/cps110/lectures/nachos_lab4.pdf
- [6] <http://www.cas.mcmaster.ca/~qiao/courses/cs3mh3/roadmaps/3.4/userprog.html>