

HỆ ĐIỀU HÀNH



ĐỒ ÁN 1: NACHOS

**Giáo viên lý thuyết: Lê Giang Thanh,
Nguyễn Thanh Quân, Lê Hà Minh**

**Thông tin nhóm:
Trần Tiến Hoàng – 20127424
Lâm Kim Nhân – 20127579
Lê Võ Huỳnh Thanh - 20127072
Lớp: 20CLC09**

Table of Contents

I. Phân chia công việc:	4
II. Sơ lược về NachOS:	4
III. Báo cáo đồ án Nachos	4
1. System call int ReadNum():	4
2. System call void PrintNum(int number):	4
3. System call char ReadChar():	5
4. System call void PrintChar(char character):	5
5. System call int RandomNum():	5
6. System call void ReadString (char[] buffer, int length):	5
7. System call void PrintString (char[] buffer):	5
8. System call int Create(char *name):	5
9. System call OpenFileID Open(char* name):	6
10. System call int Close(OpenFileID id):	6
11. System call int Read(char *buffer, int size, OpenFileID id):	6
12. System call int Write(char*buffer, int size, OpenFileID id):	6
13. System call int Seek(int position, OpenFileID id):	6
14. System call int Remove(char *name)	6
15. Chương trình Help:	6
16. Chương trình ASCII:	7
17. Chương trình Sort:	8
18. Chương trình createfile:	9
19. Chương trình cat:	9
20. Chương trình copy:	10
21. Chương trình delete:	10
IV. Tham Khảo	11

I. Phân chia công việc:

Số thứ tự	Họ và tên	MSSV	Công việc	Mức độ hoàn thành
1	Lê Võ Huỳnh Thanh	20127072	Làm mục: 3.2, 3.3, 3.4, 3.7, 3.11 Viết báo cáo	100%
2	Trần Tiến Hoàng	20127424	Làm mục: 3.1, 3.10.14; 3.12; 3.13; 3.18,19,20,21	100%
3	Lâm Kim Nhân	20127579	Làm mục: 3.5; 3.6; 3.8; 3.9; 3.15,16,17	100%

II. Sơ lược về NachOS:

Nachos là một phần mềm mã nguồn mở dùng để giả lập một máy tính ảo và một số thành phần cơ bản của hệ điều hành chạy trên máy tính ảo này nhằm giúp cho việc tìm hiểu và xây dựng các thành phần phức tạp hơn của hệ điều hành.

- Máy ảo được giả lập có kiến trúc MIPS với hầu hết các thành phần và chức năng của một máy thật như: thanh ghi, bộ nhớ, bộ xử lý, bộ lệnh, chu kỳ thực thi lệnh, cơ chế ngắt, chu kỳ đồng hồ.
- Hệ điều hành NachOS chạy trên máy ảo NachOS hiện là một hệ điều hành đơn chương

III. Báo cáo đồ án Nachos

1. System call int ReadNum():

– Giải Thích: system call này sẽ trả về một số nguyên (integer) do người dùng nhập vào từ bàn phím trong phạm vi của kiểu dữ liệu int (từ -2,147,483,648 đến 2,147,483,647), nếu chuỗi đó không phải là một số hoặc số nhập vào vượt quá phạm vi cho phép thì hàm sẽ trả về giá trị 0.

– Cách cài đặt:

Sử dụng hàm “kernel->synchConsoleIn->GetChar()” và kết hợp với vòng lặp để cho người dùng nhập chuỗi từ bàn phím.

Dùng biến đếm để đếm các ký tự nhập sai trong trường hợp người dùng cố tình nhập vào một chuỗi không hợp lệ.

Khởi tạo một biến kết quả với giá trị 0 và giá trị trả về sẽ được tính theo công thức:

$result = result \times 10 + (res - '0')$. Trong đó result là giá trị trả về, res là ký tự nhập vào từ bàn phím. Kết thúc vòng lặp, xét những điều kiện cần thiết để trả về đúng giá trị người dùng cần.

2. System call void PrintNum(int number):

– Giải Thích: system call này có nhiệm vụ in số nguyên number ra màn hình console.

– Cách cài đặt:

Kiểm tra số nguyên là âm hay dương, nếu âm thì in dấu trừ ra màn hình, sau đó tạo một biến lưu trữ trị tuyệt đối của number

Sử dụng hàm “kernel->synchConsoleIn->PutChar()” và kết hợp với vòng lặp với số lần lặp là số ký số của number để in lần lượt các chữ số ra màn hình.

Sử dụng các phép chia lấy phần nguyên và chia dư để lấy các chữ số.

In các chữ số ra màn hình dưới dạng char bằng cách lấy ký số đó cộng với 48 để xuất theo bảng mã ascii.

3. System call char ReadChar():

– Giải Thích: system call này sẽ trả về một ký tự do người dùng nhập vào từ bàn phím.

– Cách cài đặt

Sử dụng hàm “kernel->synchConsoleIn->GetChar()”

4. System call void PrintChar(char character):

– Giải Thích: system call này sẽ in 1 ký tự được truyền vào ra màn hình console.

– Cách cài đặt:

Sử dụng hàm “kernel->synchConsoleOut->PutChar()” để in ký tự người dùng muốn in ra ngoài màn hình

5. System call int RandomNum():

– Giải Thích: system call này sẽ trả 1 số nguyên dương bất kỳ từ 1 đến RAND_MAX.

– Cách cài đặt:

Sử dụng hàm rand() để tạo ra một số ngẫu nhiên bất kỳ. Ở bất kỳ lần chạy nào các kết quả ở mỗi lần chạy đều giống nhau theo thứ tự. Để mỗi lần chạy cho ra 1 chuỗi kết quả khác nhau, thay đổi random seed bằng cách sử dụng hàm srand() và truyền vào 1 con số có thể thay đổi được (thông thường sử dụng đơn vị thời gian hiện tại time(NULL) ở mỗi lần chạy). Đặt dòng code srand(time(NULL)) khi chương trình bắt đầu trước khi hàm rand() được sử dụng.

6. System call void ReadString (char[] buffer, int length):

– Giải Thích: system call này thay đổi giá trị của chuỗi buffer theo giá trị mà người dùng nhập vào từ bàn phím với độ dài tối đa là length

– Cách cài đặt:

Sử dụng hàm “kernel->synchConsoleIn->GetChar()” và kết hợp với vòng lặp để cho người dùng nhập chuỗi từ bàn phím. Người dùng sẽ nhập chuỗi đến hết số length truyền vào hoặc đến khi nhấn xuống dòng. Nếu người dùng nhập quá số ký tự thì hàm sẽ cắt ngay vị trí tối đa.

7. System call void PrintString (char[] buffer):

– Giải Thích: system call này sẽ in chuỗi buffer được truyền vào ra màn hình console.

– Cách cài đặt:

Sử dụng hàm “kernel->synchConsoleIn->PutChar()” và kết hợp với vòng lặp với số lần lặp là số ký tự của buffer để in lần lượt các ký tự của buffer ra màn hình.

8. System call int Create(char *name):

– Giải thích: Create system call sẽ sử dụng Nachos FileSystem Object để tạo một file rỗng. System call Create trả về 0 nếu thành công và -1 nếu có lỗi

– Cách cài đặt:

Sử dụng hàm “kernel->fileSystem->Create()”

9. System call `OpenFileID Open(char* name)`:

- Giải thích: Hàm này dùng để mở file
- Cách cài đặt:

Khai báo “`OpenFile* files`” trong kernel để lưu các file đang được mở và sử dụng hàm “`kernel->fileSystem->Open()`”

10. System call `int Close(OpenFileID id)`:

- Giải thích: Hàm này dùng để đóng file
- Cách cài đặt:

Nếu `kernel->files[id] != NULL` thì delete và đặt NULL cho “`kernel->files[id]`”

11. System call `int Read(char *buffer, int size, OpenFileID id)`:

- Giải thích: Read system call đọc file với id cho trước. Lưu ý cần phải chuyển vùng nhớ giữa user space và system space, và cần phải phân biệt giữa Console IO (`OpenFileID 0, 1`) và File.
- Cách cài đặt:

Nếu `id == 1` thì sử dụng hàm “`kernel->synchConsoleIn->GetChar()`” nếu `id` từ 2 đến 9 thì sử dụng hàm “`OpenFile->Read()`”

12. System call `int Write(char*buffer, int size, OpenFileID id)`:

- Giải thích: Write system call ghi vào file với id cho trước. Lưu ý cần phải chuyển vùng nhớ giữa user space và system space, và cần phải phân biệt giữa Console IO (`OpenFileID 0, 1`) và File.
- Cách cài đặt:

Nếu `id == 0` thì sử dụng hàm “`kernel->synchConsoleOut->PutChar()`” nếu `id` từ 2 đến 9 thì sử dụng hàm “`OpenFile->Write()`”

13. System call `int Seek(int position, OpenFileID id)`:

- Giải thích: Seek sẽ phải chuyển con trỏ tới vị trí thích hợp. position lưu vị trí cần chuyển tới, nếu `pos = -1` thì di chuyển đến cuối file. Trả về vị trí thực sự trong file nếu thành công và -1 nếu bị lỗi. Gọi Seek trên console phải báo lỗi.
- Cách cài đặt:

Sử dụng hàm “`OpenFile->Seek()`”, “`OpenFile->Length()`”. 2 hàm này có chút lỗi nên nhóm đã sửa file “`openfile.h`” trong thư mục “`fileSYS`”

14. System call `int Remove(char *name)`

- Giải thích: Remove system call sẽ sử dụng Nachos FileSystem Object để xóa file. Chú ý: cần kiểm tra file có đang mở hay không trước khi xóa.
- Cách cài đặt:

Nếu file không tồn tại hoặc file được mở thì trả về -1, ngược lại sử dụng hàm “`kernel->fileSystem->Remove()`”

15. Chương trình Help:

- Giải Thích: hàm này giới thiệu về các thành viên của nhóm và mô tả vắn tắt về chương trình Sort và ASCII.
- Cách cài đặt:

Sử dụng system call `PrintString()` để in các nội dung giới thiệu cơ bản về nhóm và các chương trình sort, ascii.

```
tienhoangg@tienhoangg:~/NachOS/NachOS-4.0/code/test$ ../build.linux/nachos -x help
---GROUP INFORMATION---
Le Vo Huynh Thanh - 20127072
Tran Tien Hoang - 20127424
Lam Kim Nhan - 20127579

---SORT---
Sort, sometimes referred to as sinking sort
It is a simple sorting algorithm that repeatedly steps through the list
It compares adjacent elements and swaps them if they are in the wrong order
The pass through the list is repeated until the list is sorted.

---ASCII---
Stands for: American Standard Code for Information Interchange.
ASCII is a character encoding that uses numeric codes to represent characters.
These include upper and lowercase English letters, numbers, and punctuation symbols

Machine halting!

Ticks: total 82913, idle 62090, system 20720, user 103
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 621
Paging: faults 0
Network I/O: packets received 0, sent 0
```

16. Chương trình ASCII:

- Giải Thích: hàm này in ra bảng mã ASCII
- Cách cài đặt:

Sử dụng system call PrintString().

```
tienhoangg@tienhoangg:~/NachOS/NachOS-4.0/code/test$ ../build.linux/nachos -x ascii
-----ASCII-----
33      !      |65      A      |97      a      |
34      "      |66      B      |98      b      |
35      #      |67      C      |99      c      |
36      $      |68      D      |100     d      |
37      %      |69      E      |101     e      |
38      &      |70      F      |102     f      |
39      '      |71      G      |103     g      |
40      (      |72      H      |104     h      |
41      )      |73      I      |105     i      |
42      *      |74      J      |106     j      |
43      +      |75      K      |107     k      |
44      ,      |76      L      |108     l      |
45      -      |77      M      |109     m      |
46      .      |78      N      |110     n      |
47      /      |79      O      |111     o      |
48      0      |80      P      |112     p      |
49      1      |81      Q      |113     q      |
50      2      |82      R      |114     r      |
51      3      |83      S      |115     s      |
52      4      |84      T      |116     t      |
53      5      |85      U      |117     u      |
54      6      |86      V      |118     v      |
55      7      |87      W      |119     w      |
56      8      |88      X      |120     x      |
57      9      |89      Y      |121     y      |
58      :      |90      Z      |122     z      |
59      ;      |91      [      |123     {      |
60      <      |92      \      |124     |      |
61      =      |93      ]      |125     }      |
62      >      |94      ^      |126     ~      |
63      ?      |95      _      |127     |      |
64      @      |96      `      |          |
Machine halting!

Ticks: total 90858, idle 65290, system 22050, user 3518
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 653
Paging: faults 0
Network I/O: packets received 0, sent 0
```

17. Chương trình Sort:

– Giải Thích: chương trình này sẽ sắp xếp một chuỗi n số nguyên do người dùng nhập vào bằng thuật toán bubble sort theo chiều người dùng chọn.

– Cách cài đặt:

Sử dụng system call ReadNum() để đọc số nguyên n được nhập từ bàn phím, đó là số phần tử của mảng số nguyên. Người dùng sẽ nhập vào các số nguyên với số lượng bằng n phần tử đã nhập. Người dùng sẽ chọn thứ tự muốn sắp xếp. Chọn 1 là chọn chiều sắp xếp tăng dần, chọn 2 là chọn chiều sắp xếp giảm dần. Dùng thuật toán Bubble Sort để sắp xếp

mảng. Sử dụng system call PrintNum() để in các số đã sắp xếp lên màn hình console.

```
tienhoangg@tienhoangg:~/NachOS/NachOS-4.0/code/test$ ../build.linux/nachos -x sort
Input n: 7
array[0] = 3
array[1] = 4
array[2] = 2
array[3] = 6
array[4] = 5
array[5] = 1
array[6] = 7
1: Increase
2: Decrease
Choose sort order you want: 1
1 2 3 4 5 6 7
Machine halting!

Ticks: total 561319044, idle 561311176, system 5900, user 1968
Disk I/O: reads 0, writes 0
Console I/O: reads 18, writes 153
Paging: faults 0
Network I/O: packets received 0, sent 0
```

18. Chương trình createfile:

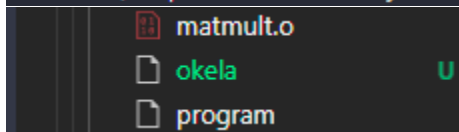
– Giải thích: Tạo file với tên do người dùng nhập vào.

– Cách cài đặt:

Sử dụng system call Create(char*) để tạo file

```
tienhoangg@tienhoangg:~/NachOS/NachOS-4.0/code/test$ ../build.linux/nachos -x createfile
$create okela
Done!
Machine halting!

Ticks: total 290093695, idle 290092993, system 660, user 42
Disk I/O: reads 0, writes 0
Console I/O: reads 6, writes 14
Paging: faults 0
Network I/O: packets received 0, sent 0
```



19. Chương trình cat:

– Giải thích: đọc nội dung trong 1 file và in ra màn hình

– Cách cài đặt:

Mở file, dịch con trỏ về cuối file để lấy số lượng kí tự trong file. Dùng system call Read() để đọc file. Dùng hàm printString() để in ra màn hình.


```
tienhoangg@tienhoangg:~/NachOS/NachOS-4.0/code/test$ ../build.linux/nachos -x cat
$cat okela
You are not my friend
You are my brother
    My friend!
Machine halting!

Ticks: total 105730227, idle 105727893, system 2240, user 94
Disk I/O: reads 0, writes 0
Console I/O: reads 6, writes 61
Paging: faults 0
Network I/O: packets received 0, sent 0
```

20. Chương trình copy:

- Giải thích: nhập tên file nguồn và file đích rồi thực hiện copy
- Cách cài đặt:

Mở file, dịch con trỏ về cuối file để lấy số lượng kí tự trong file. Dùng system call Read() để đọc file. Dùng system call Create() để tạo file đích và dùng system call Write() để viết vào file.

```
tienhoangg@tienhoangg:~/NachOS/NachOS-4.0/code/test$ ../build.linux/nachos -x copy
$copy okela
okelaCopy
Machine halting!

Ticks: total 416877238, idle 416876372, system 750, user 116
Disk I/O: reads 0, writes 0
Console I/O: reads 16, writes 6
Paging: faults 0
Network I/O: packets received 0, sent 0
tienhoangg@tienhoangg:~/NachOS/NachOS-4.0/code/test$ ../build.linux/nachos -x cat
$cat okelaCopy
You are not my friend
You are my brother
    My friend!
Machine halting!

Ticks: total 224669858, idle 224667392, system 2380, user 86
Disk I/O: reads 0, writes 0
Console I/O: reads 10, writes 61
Paging: faults 0
Network I/O: packets received 0, sent 0
```

okela	U
okelaCopy	U
program	M

21. Chương trình delete:

- Giải thích: xóa file có tên do người dùng nhập vào
- Cách cài đặt:

Kiểm tra file có tồn tại không, nếu có thì sử dụng system call “Remove()” để xóa file.

```
tienhoangg@tienhoangg:~/NachOS/NachOS-4.0/code/test$ ../build.linux/nachos -x delete
$delete okelaCopy
Done!
Machine halting!

Ticks: total 392849545, idle 392848693, system 810, user 42
Disk I/O: reads 0, writes 0
Console I/O: reads 10, writes 14
Paging: faults 0
Network I/O: packets received 0, sent 0
```

IV. Tham Khảo

File: [CLC Project 01 HK2 2122 syscallFile](#)

[quanghuy1258/nachos_os_projects_HCMUS](#): Tập hợp đủ 3 đồ án hệ điều hành nachos - môn hệ điều hành - khoa công nghệ thông tin - trường đại học khoa học tự nhiên tphcm - HCMUS (github.com)

[nguyenthanchungfit/Nachos-Programing-HCMUS](#): This project includes: (github.com)

[FIT HCMUS | Hệ điều hành | Đồ án 1 NachOS - YouTube](#)

