

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



**MÔN: HỆ ĐIỀU HÀNH
ĐỒ ÁN THỰC HÀNH 2**

**TÌM HIỂU SYSTEM CALLS
CÁC THAO TÁC VỚI FILES**

Ngày 16 tháng 4 năm 2022

MỤC LỤC

I. Thông tin nhóm:.....	1
II. Đánh giá mức độ hoàn thành:.....	1
III. Cài đặt system calls thao tác với các file	1
1. Cài đặt system call: int Create(char *name).	1
2. Cài đặt system call OpenFileID Open(char *name)	1
3. Cài đặt system call int Close(OpenFileID id)	2
4. Cài đặt system call int Read(char *buffer, int size, OpenFileID id).....	2
5. Cài đặt system call int Write(char *buffer, int size, OpenFileID id).....	3
6. Cài đặt system call int Seek(int position, OpenFileID id)	3
7. Cài đặt system call int Remove(char *name)	4
IV. Demo chương trình người dùng:.....	5
1. Chương trình Createfile:	5
2. Chương trình cat:.....	5
3. Chương trình copy:.....	6
4. Chương trình Delete:	6
5. Chương trình concatenate:	7
V. Bảng phân công công việc:	8

I. Thông tin nhóm:

Lớp: 20CLC05

Thành viên:

STT	MSSV	HỌ VÀ TÊN	MAIL
1	20127287	Đinh Cao Hồng Phước	20127287@student.hcmus.edu.vn
2	20127546	Võ Thanh Lâm	20127546@student.hcmus.edu.vn
3	20127561	Nguyễn Hoài Mẫn	20127561@student.hcmus.edu.vn

II. Đánh giá mức độ hoàn thành:

- Mức độ hoàn thành đồ án của nhóm: 100%.

III. Cài đặt system calls thao tác với các file

1. Cài đặt system call: int Create(char *name).

- Ý tưởng: Sử dụng hàm có sẵn trong `filesystem.h`.
- Mô tả: Tạo một file với tên có sẵn.

2. Cài đặt system call OpenFileID Open(char *name)

- Ý tưởng:

+ Thêm vào class một con trỏ cấp 2 có tên là `fileTable` để lưu địa chỉ của các file đang được mở.

+ Kiểm tra xem file có đang được mở hay chưa bằng cách kiểm tra trong `fileTable`, nếu file đã được mở thì trả về giá trị -1.

+ Tìm vị trí trống trong fileTable, nếu tìm thấy thì new OpenFile và trả về vị trí đó. Nếu không tìm thấy thì trả về giá trị -1.

- Mô tả: Mở một file với tên có sẵn.

3. Cài đặt system call int Close(OpenFileID id)

- Ý tưởng:

+ Kiểm tra xem fileTable[id] có hay không, nếu có thì delete và gán bằng NULL rồi trả về TRUE, nếu không thì trả về FALSE.

- Mô tả: Đóng một file với tên có sẵn.

4. Cài đặt system call int Read(char *buffer, int size, OpenFileID id)

- Ý tưởng:

+ Các tham số đầu vào:

- buffer: Nội dung đọc được từ file.
- size: Số ký tự đọc từ file.
- Id: Vị trí trong fileTable.

+ Kiểm tra tính hợp lệ của size và id.

+ Sử dụng hàm có sẵn trong thư viện openfile.h.

- Mô tả: Đọc một file có sẵn với kích thước và id cho trước. Nội dung đọc được thì gán vào buffer.

5. Cài đặt system call int Write(char *buffer, int size, OpenFileID id)

- Ý tưởng:

+ Các tham số đầu vào:

- buffer: Nội dung cần ghi xuống file.
- size: Số ký tự ghi xuống file.
- id: Vị trí trong fileTable.

+ Kiểm tra tính hợp lệ của size và id.

+ Sử dụng hàm có sẵn trong thư viện openfile.h.

- Mô tả: Ghi nội dung có trong buffer xuống file với id cho trước.

6. Cài đặt system call int Seek(int position, OpenFileID id)

- Ý tưởng:

+ Cài đặt bên openfile.h.

+ Kiểm tra tính hợp lệ của position, nếu position < -1 hoặc position > Length() thì trả về -1.

+ Nếu position = -1 thì gán currentOffset = Length() để dịch chuyển con trỏ về cuối file rồi trả về vị trí đó.

+ Nếu position >= 0 thì gán currentOffset = position rồi trả về vị trí đó.

- Mô tả: Dịch chuyển con trỏ trong file đến vị trí cho trước.

7. Cài đặt system call int Remove(char *name)

- Ý tưởng:

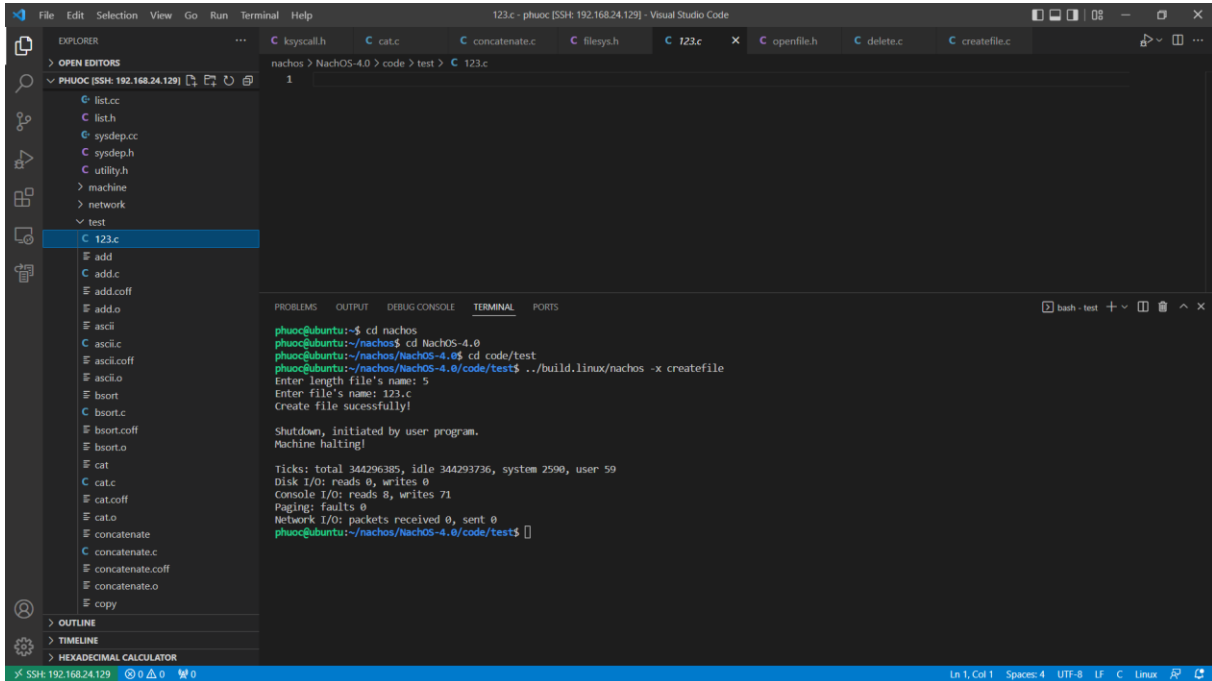
+ Kiểm tra xem file có đang được mở hay không bằng cách kiểm tra xem name có trong fileTable hay không, nếu có thì return FALSE.

+ Dùng hàm có sẵn Unlink để xóa file

- Mô tả: Xóa một file với tên có sẵn

IV. Demo chương trình người dùng:

1. Chương trình Createfile:



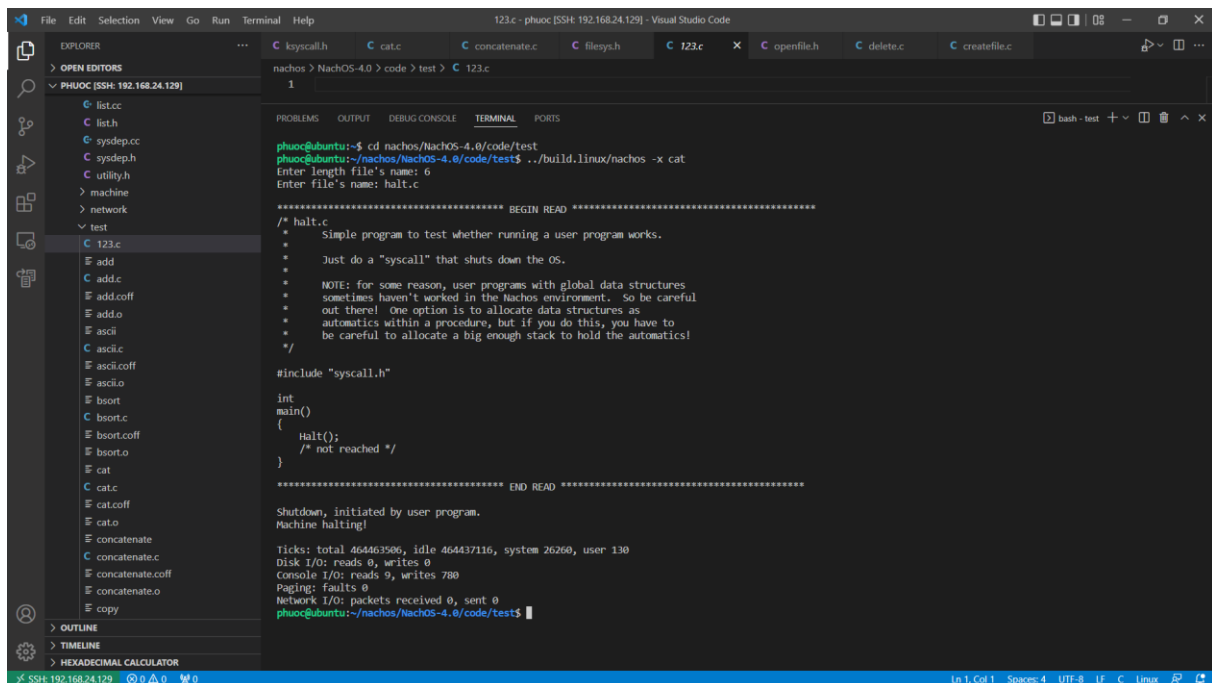
The screenshot shows the Visual Studio Code interface with the 'createfile' program being executed. The Explorer pane on the left shows the file structure of the Nachos project. The main editor displays the output of the program, which includes the following text:

```
phuc@ubuntu:~$ cd nachos
phuc@ubuntu:~/nachos$ cd NachOS-4.0
phuc@ubuntu:~/nachos/NachOS-4.0$ cd code/test
phuc@ubuntu:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x createfile
Enter length file's name: 5
Enter file's name: 123.c
Create file successfully!

Shutdown, initiated by user program.
Machine halting!

Ticks: total 344296385, idle 344293736, system 2590, user 59
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 71
Paging: faults 0
Network I/O: packets received 0, sent 0
phuc@ubuntu:~/nachos/NachOS-4.0/code/test$
```

2. Chương trình cat:



The screenshot shows the Visual Studio Code interface with the 'cat' program being executed. The Explorer pane on the left shows the file structure of the Nachos project. The main editor displays the output of the program, which includes the following text:

```
phuc@ubuntu:~$ cd nachos/NachOS-4.0/code/test
phuc@ubuntu:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x cat
Enter length file's name: 6
Enter file's name: halt.c

***** BEGIN READ *****

/* halt.c
 * Simple program to test whether running a user program works.
 * Just do a "syscall" that shuts down the OS.
 * NOTE: for some reason, user programs with global data structures
 * sometimes haven't worked in the Nachos environment. So be careful
 * out there! One option is to allocate data structures as
 * automatics within a procedure, but if you do this, you have to
 * be careful to allocate a big enough stack to hold the automatics!
 */

#include "syscall.h"

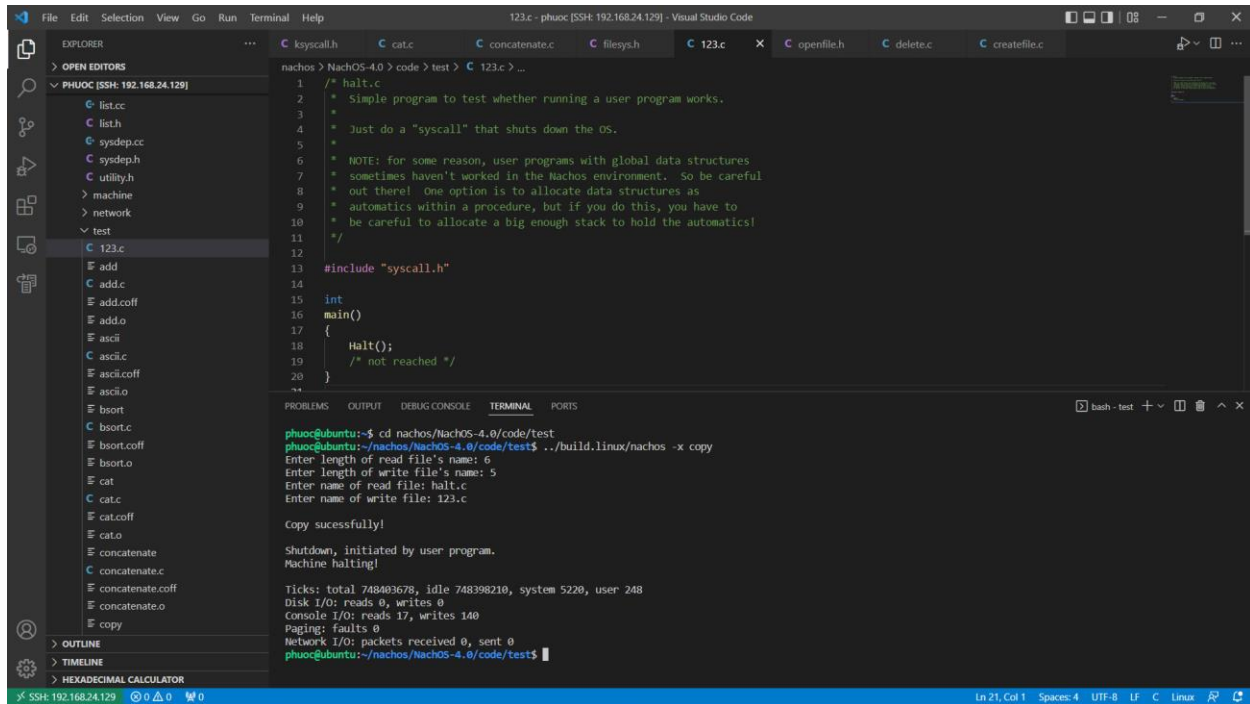
int
main()
{
    Halt();
    /* not reached */
}

***** END READ *****

Shutdown, initiated by user program.
Machine halting!

Ticks: total 464463506, idle 464437116, system 26260, user 130
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 780
Paging: faults 0
Network I/O: packets received 0, sent 0
phuc@ubuntu:~/nachos/NachOS-4.0/code/test$
```

3. Chương trình copy:



```
1 /* halt.c
2  * Simple program to test whether running a user program works.
3  *
4  * Just do a "syscall" that shuts down the OS.
5  *
6  * NOTE: for some reason, user programs with global data structures
7  * sometimes haven't worked in the Nachos environment. So be careful
8  * out there! One option is to allocate data structures as
9  * automatics within a procedure, but if you do this, you have to
10 * be careful to allocate a big enough stack to hold the automatics!
11 */
12
13 #include "syscall.h"
14
15 int
16 main()
17 {
18     Halt();
19     /* not reached */
20 }
21
```

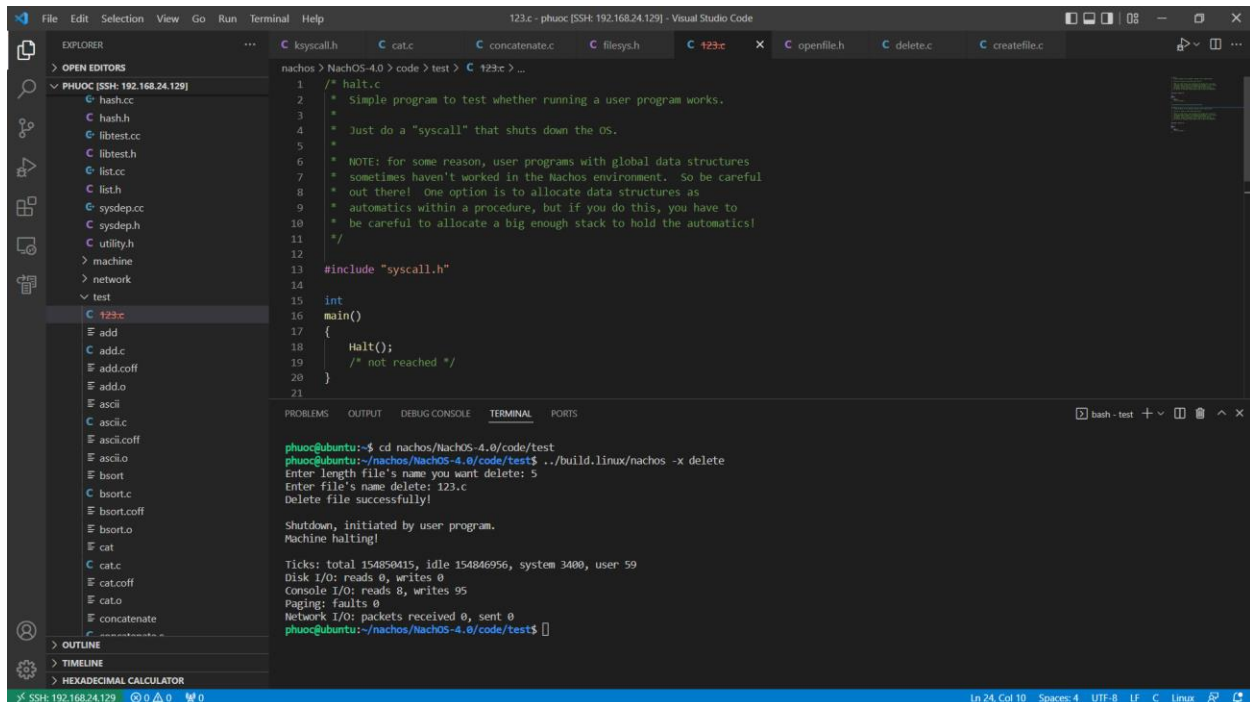
```
phuc@ubuntu:~$ cd nachos/NachOS-4.0/code/test
phuc@ubuntu:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x copy
Enter length of read file's name: 6
Enter length of write file's name: 5
Enter name of read file: halt.c
Enter name of write file: 123.c

Copy successfully!

Shutdown, initiated by user program.
Machine halting!

Ticks: total 748403678, idle 748398210, system 5220, user 248
Disk I/O: reads 0, writes 0
Console I/O: reads 17, writes 140
Paging: faults 0
Network I/O: packets received 0, sent 0
phuc@ubuntu:~/nachos/NachOS-4.0/code/test$
```

4. Chương trình Delete:



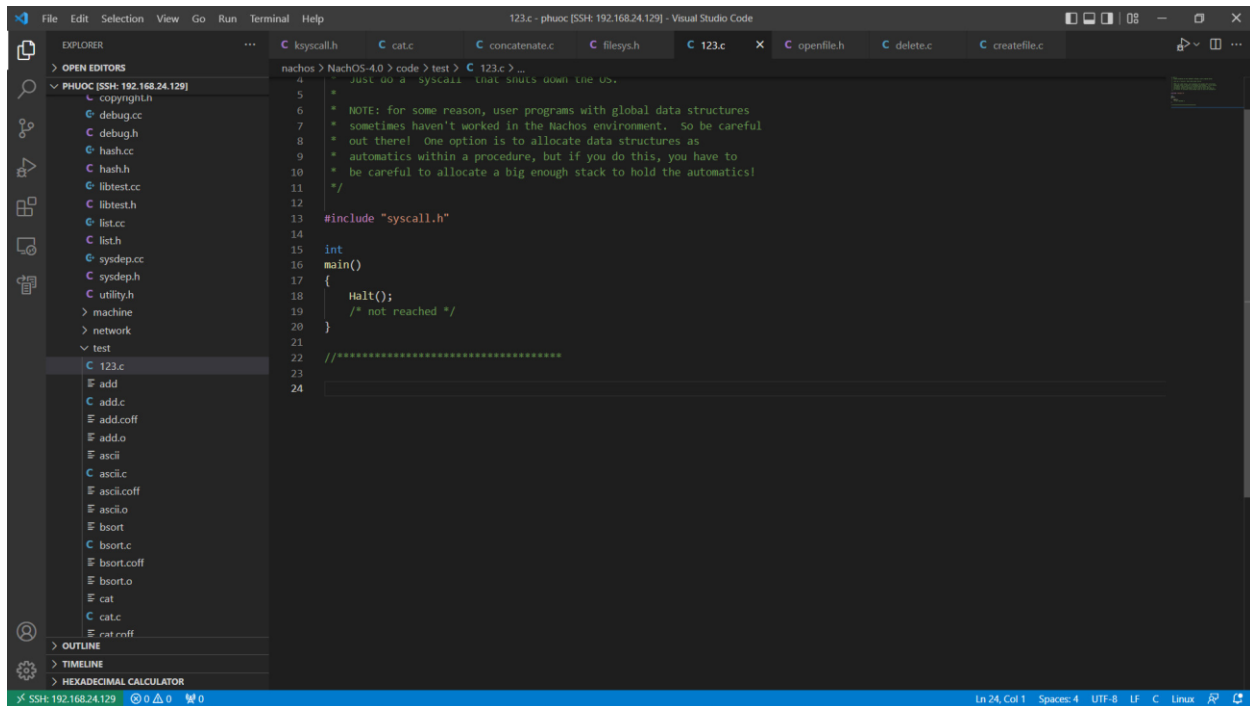
```
1 /* halt.c
2  * Simple program to test whether running a user program works.
3  *
4  * Just do a "syscall" that shuts down the OS.
5  *
6  * NOTE: for some reason, user programs with global data structures
7  * sometimes haven't worked in the Nachos environment. So be careful
8  * out there! One option is to allocate data structures as
9  * automatics within a procedure, but if you do this, you have to
10 * be careful to allocate a big enough stack to hold the automatics!
11 */
12
13 #include "syscall.h"
14
15 int
16 main()
17 {
18     Halt();
19     /* not reached */
20 }
21
```

```
phuc@ubuntu:~$ cd nachos/NachOS-4.0/code/test
phuc@ubuntu:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x delete
Enter length file's name you want delete: 5
Enter file's name delete: 123.c
Delete file successfully!

Shutdown, initiated by user program.
Machine halting!

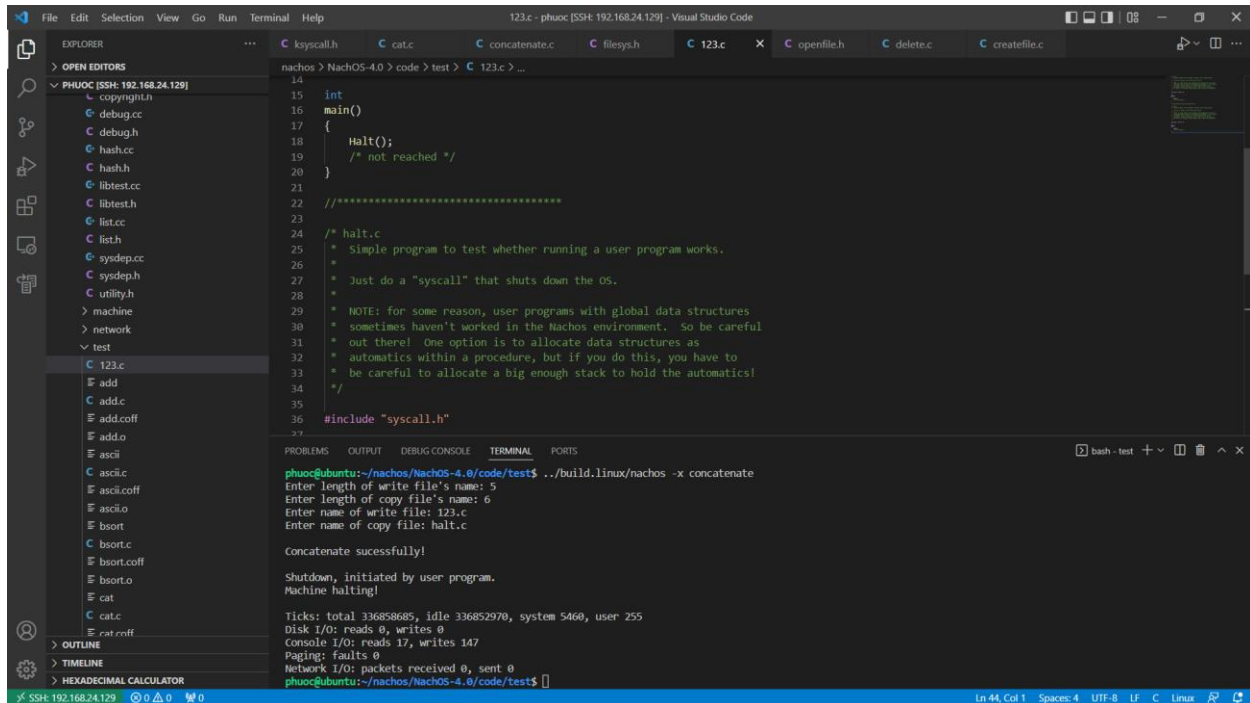
Ticks: total 154858415, idle 154846956, system 3400, user 59
Disk I/O: reads 0, writes 0
Console I/O: reads 8, writes 95
Paging: faults 0
Network I/O: packets received 0, sent 0
phuc@ubuntu:~/nachos/NachOS-4.0/code/test$
```


5. Chương trình concatenate:



The screenshot shows the Visual Studio Code editor with the file `concatenate.c` open. The file is located in the `test` directory of the `NachOS-4.0` project. The code is as follows:

```
1  nachos > NachOS-4.0 > code > test > C 123.c > ...
2  Just do a "syscall" that shuts down the OS.
3
4
5  *
6  * NOTE: for some reason, user programs with global data structures
7  * sometimes haven't worked in the Nachos environment. So be careful
8  * out there! One option is to allocate data structures as
9  * automatics within a procedure, but if you do this, you have to
10 * be careful to allocate a big enough stack to hold the automatics!
11 */
12
13 #include "syscall.h"
14
15 int
16 main()
17 {
18     Halt();
19     /* not reached */
20 }
21
22 //*****
23
24
```



The screenshot shows the Visual Studio Code editor with the file `concatenate.c` open. The file is located in the `test` directory of the `NachOS-4.0` project. The code is as follows:

```
15 int
16 main()
17 {
18     Halt();
19     /* not reached */
20 }
21
22 //*****
23
24 /* halt.c
25 * Simple program to test whether running a user program works.
26 *
27 * Just do a "syscall" that shuts down the OS.
28 *
29 * NOTE: for some reason, user programs with global data structures
30 * sometimes haven't worked in the Nachos environment. So be careful
31 * out there! One option is to allocate data structures as
32 * automatics within a procedure, but if you do this, you have to
33 * be careful to allocate a big enough stack to hold the automatics!
34 */
35
36 #include "syscall.h"
37
```

The terminal output shows the execution of the `concatenate` program:

```
phuc@ubuntu:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x concatenate
Enter length of write file's name: 5
Enter length of copy file's name: 6
Enter name of copy file: halt.c
Concatenate successfully!
Shutdown, initiated by user program.
Machine halting!
Ticks: total 336858685, idle 336852970, system 5460, user 255
Disk I/O: reads 0, writes 0
Console I/O: reads 17, writes 147
Paging: faults 0
Network I/O: packets received 0, sent 0
phuc@ubuntu:~/nachos/NachOS-4.0/code/test$
```

V. Bảng phân công công việc:

MSSV	Tên	Phân công	Đánh giá
20127287	Đinh Cao Hồng Phước	<ul style="list-style-type: none">- Open- Close- Sao chép vùng nhớ kernel-user- Chương trình concatenate- Viết báo cáo	100%
20127546	Võ Thanh Lâm	<ul style="list-style-type: none">- Create- Seek- Remove- Chương trình createfile- Chương trình Delete	100%
20127561	Nguyễn Hoài Mẫn	<ul style="list-style-type: none">- Read- Write- Chương trình cat- Chương trình copy	100%