

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

-----o0o-----

BÁO CÁO ĐỒ ÁN HỆ ĐIỀU HÀNH

Project 1: System calls & File - Network Operations



Giáo viên hướng dẫn: Nguyễn Thanh Quân

Lê Giang Thanh

Lê Hà Minh

Thành phố Hồ Chí Minh, tháng 3 năm 2023

MỤC LỤC

PHẦN 1: THÔNG TIN CHUNG

1. Thông tin chung:

Tên giảng viên: Nguyễn Thanh Quân, Lê Giang Thanh, Lê Hà Minh

Tên đồ án: System calls & File - Network Operations

Thời gian thực hiện: Từ 6/2/2023 tới 2/4/2023

2. Thông tin nhóm:

MSSV	Họ & tên	Email
21127443	Trần Ngọc Trường Thịnh	tntthinh21@clc.fitus.edu.vn
21127712	Lê Quang Trường	lqtruong21@clc.fitus.edu.vn
21127329	Châu Tấn Kiệt	ctkiet21@clc.fitus.edu.vn

PHẦN 2: ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH

1. Bảng đánh giá mức độ hoàn thành:

Part		Describe	Mức độ hoàn thành
1	1	Syscall Create	100%
	2	Syscall OpenFileID, Close	100%
	3	Syscall Read, Write	90%
	4	Syscall Seek	100%
	5	Syscall Remove	100%
2	1	Syscall socketTCP	100%
	2	Syscall Connect	100%
	3	Syscall Send	100%
	4	Syscall Receive	100%
3		Advanced	100%
4		Test Program	100%
	1,2,3,4,5	Create, copy, cat, delete, concatenate	100%
	6	Echo	100%
	7	File transfer	90%
		Report	100%

PHẦN 3: THÔNG TIN ĐỒ ÁN

1. Syscall cho các thao tác quản lý tập tin:

Hàm User2System: chuyển 1 chuỗi được lưu vào vùng nhớ của chương trình vào vùng nhớ của hệ điều hành

Hàm System2User: chuyển 1 chuỗi được lưu vào vùng nhớ của hệ điều hành vào vùng nhớ của chương trình

Hàm IncreasePC: tăng programe counter để nạp lệnh tiếp theo để thực hiện. Lưu giá trị của PC hiện tại cho PC trước, nạp giá trị kế cho PC hiện tại, nạp giá trị kế tiếp theo cho PC kế.

1.1 Syscall Create:

Chương trình tạo ra một file với tên do người dùng đặt ra và trả về 0 khi thành công và -1 khi thất bại vào thanh nhớ 2. Tạo file thất bại khi tên file không hợp lệ, khi không có đủ bộ nhớ để tạo file và nếu file cùng tên không tồn tại trong thư mục thì quá trình tạo file đã bị lỗi và chương trình trả về -1. Chương trình nhận địa chỉ file vào thanh ghi số 4 và sử dụng hàm Create trong fileSystem. File mới được tạo ra sẽ được thêm vào mảng các file dưới dạng OpenFile. Mặc định file mới tạo ra sẽ nằm ở vị trí thứ 3 trở đi trong mảng.

1.2 Syscall OpenFileID, Close:

Open: Chương trình nhận vào tên file rồi xuất ra nội dung của file rồi sử dụng hàm open trong fileSystem để thực hiện chức năng nếu thành công hoặc -1 nếu thất bại. Chương trình đọc tên file trong thanh ghi số 4 và loại file trong thanh ghi số 5.

Close: Chương trình nhận vào tên file và đóng file sau đó trả về 0 nếu thành công và -1 nếu thất bại. Chương trình đọc tên file trong thanh ghi số 4, đóng file thất bại khi tên file không tồn tại hoặc khi file không còn mở.

1.3 Syscall Read, Write:

Read: Chương trình nhận vào tên file và in ra lượng bytes đã đọc trong file bằng hàm Read trong fileSystem sau đó trả về 0 nếu thành công hoặc -1 nếu thất bại khi file là không tồn tại hoặc file là Write only. Chương trình đọc tên file trong thanh ghi số 4, kích thước file trong thanh ghi số 5 và loại file trong thanh ghi số 6.

Write: Chương trình mở file định sẵn và ghi vào “Hello, what’s your name?” và trả về 0 sau đó sử dụng hàm Write trong fileSystem để thực hiện chức năng nếu thành công, -1 nếu thất bại khi file là Read only hoặc tên file không tồn tại. Chương trình đọc tên file trong thanh ghi số 4, kích thước file trong thanh ghi số 5 và loại file trong thanh ghi số 6.

1.4 Syscall Seek:

Chương trình di chuyển con trỏ vào 1 vị trí trong file sau đó trả về vị trí mới nếu thành công và -1 nếu thất bại. Chương trình đọc vị trí cần di chuyển tới trong thanh ghi số 4, fileID trong thanh ghi số 5 và sử dụng hàm seek trong filesystem để thực hiện chức năng.

1.5 Syscall Remove:

Chương trình nhận vào tên file rồi kiểm tra điều kiện để file có thể bị xóa hay không. Nếu file đang mở hoặc địa chỉ file nhập vào là sai thì chương trả về -1. Nếu không gặp điều kiện nào thì file được xóa rồi chương trình trả về 0. Chương trình đọc địa chỉ file trong thanh ghi số 4 và thực hiện chức năng bằng hàm removeFile trong fileSystem.

2. Syscall cho các thao tác Network

2.1 Syscall socketTCP:

Hàm thực hiện thao tác chính của syscall này là int FileSystem::AppendSocket(). Trong hàm này, ta sẽ lấy slot còn trống cho file tiếp theo bằng hàm BlankSpace() trả về vị trí còn trống trong mảng file descriptor (giới hạn 20 file). Tiếp theo tạo một file descriptor mới với định dạng là một file socket, tạo đối tượng OpenFile bằng file socket đã tạo tại slot còn trống. Trả về vị trí file socket trong mảng các file descriptor.

Dữ liệu đầu vào của các System call sẽ có socketid là vị trí của socket file trong mảng các file descriptor. Ta trích xuất file descriptor thông qua FileSystem, lấy được đối tượng OpenFile của socket file.

2.2 Syscall Connect:

Việc kết nối được cài đặt trong class OpenFile, bao gồm các thao tác cơ bản để kết nối đến một host với địa chỉ IP và port được truyền vào.

2.3 Syscall Send:

System call này được cài đặt ở hàm int Send(char *buffer, int len) với thao tác chính chỉ là gửi thông qua socket file. Tuy nhiên việc gọi Syscall này sẽ thực hiện hàm int Write(char *from, int numBytes) ở phần 1. Hàm Write được cài đặt sẽ thực hiện thao tác khác nhau đối với định dạng file khác nhau. Ở file có định dạng là IS_SOCKET nên hàm sẽ thực hiện gọi hàm Send ở trên.

2.4 Syscall Receive:

System call sẽ gọi hàm `int Read(char *into, int numBytes)` cài đặt ở phần 1. Hàm `Read` được cài đặt sẽ thực hiện thao tác khác nhau đối với định dạng file khác nhau. Ở file có định dạng là `IS_SOCKET`, hàm `Read` sẽ thực hiện đọc dữ liệu nhận được từ socket file vào buffer.

3. Advanced

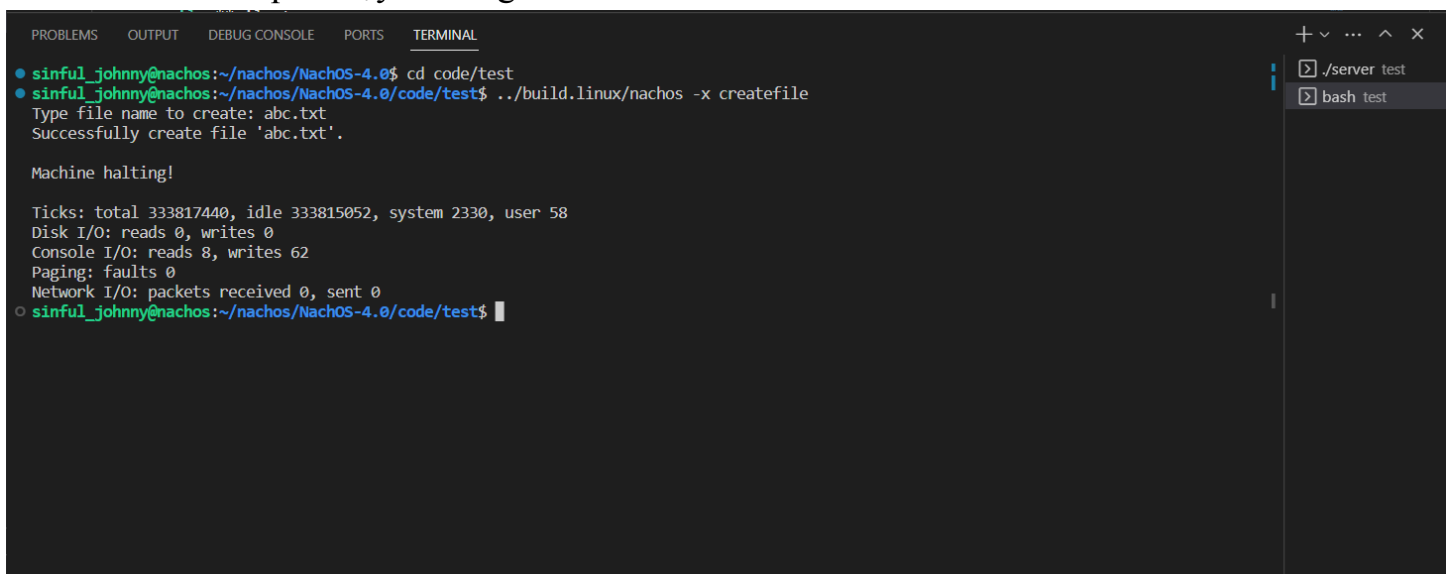
- Chương trình sử dụng duy nhất một array là `OpenFile **FilePtr` của class `FileSystem`
- Người dùng sử dụng hàm `Write` và `Read` đối với socket file cho việc gửi nhận dữ liệu, thể hiện qua các chương trình test sau đây.

4. Tests

4.1 Create:

Chương trình sẽ đọc tên file muốn tạo từ console và tạo file thông qua syscall `Read`.

Kết quả chạy chương trình:



```
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL
● sinful_johnny@nachos:~/nachos/NachOS-4.0$ cd code/test
● sinful_johnny@nachos:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x createfile
Type file name to create: abc.txt
Successfully create file 'abc.txt'.

Machine halting!

Ticks: total 333817440, idle 333815052, system 2330, user 58
Disk I/O: reads 0, writes 0
Console I/O: reads 8, writes 62
Paging: faults 0
Network I/O: packets received 0, sent 0
○ sinful_johnny@nachos:~/nachos/NachOS-4.0/code/test$
```

4.2 Cat:

Chương trình thực hiện đọc tên file từ console và mở file đó với chế độ `READWRITE` bằng Syscall `Open`. Chương trình đọc file bằng Syscall `Read` và in ra màn hình

Kết quả chạy chương trình:

```
kiet@kiet-virtual-machine: ~/nachos/NachOS-4.0/code/test
kiet@kiet-virtual-machine:~/nachos/NachOS-4.0/code/test$ cat abc.txt
hello therekiet@kiet-virtual-machine:~/nachos/NachOS-4.0/code/test$ S
```

4.3 Copy:

Chương trình thực hiện đọc tên file để copy và file đích đến từ console. Nếu file đích đến chưa được tạo thì Create và mở file đó với chế độ READWRITE bằng Syscall Open. Sau đó đọc nội dung của file muốn copy và sao chép sang file đích đến

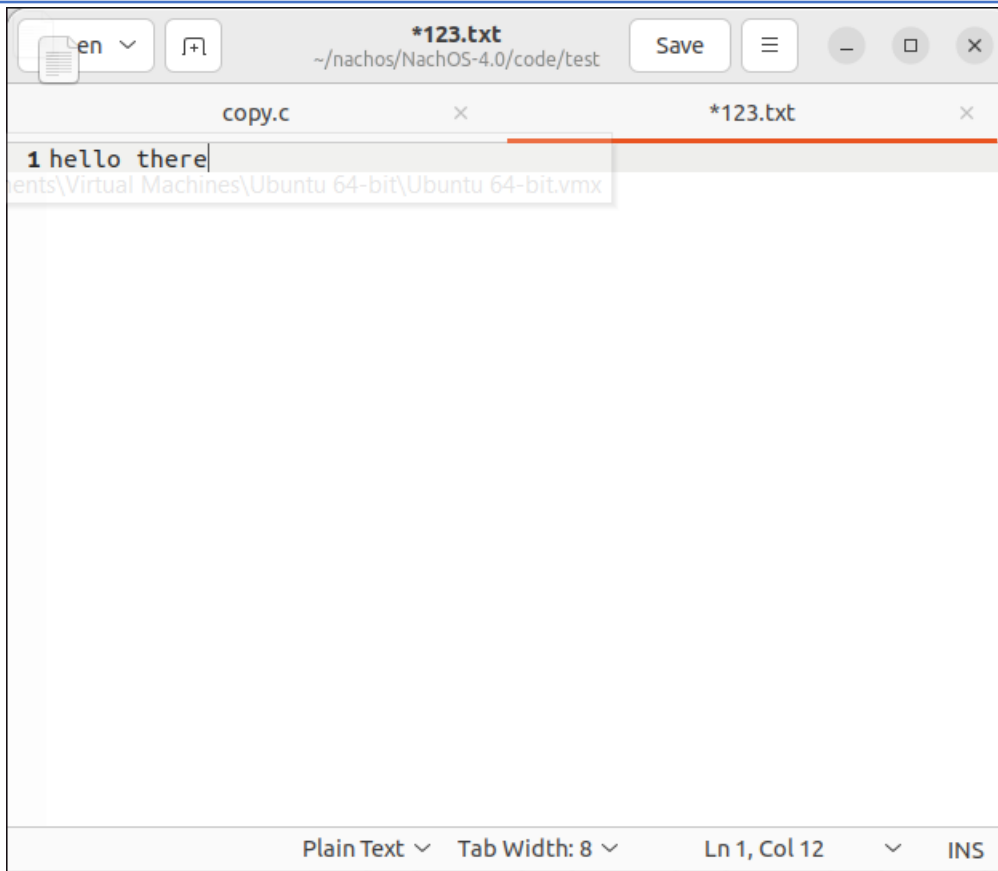
Kết quả chạy chương trình:

```
kiet@kiet-virtual-machine:~/nachos/NachOS-4.0/code/test$ chmod +x ../build.linux/nachos
kiet@kiet-virtual-machine:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x copy
Type source file: abc.txt
Type destination file: 123.txt

Successfully copy file 'abc.txt' to file '123.txt'

Machine halting!

Ticks: total 588604747, idle 588584362, system 20230, user 155
Disk I/O: reads 0, writes 0
Console I/O: reads 16, writes 591
Paging: faults 0
Network I/O: packets received 0, sent 0
kiet@kiet-virtual-machine:~/nachos/NachOS-4.0/code/test$ S
```

4.4 Delete:

Chương trình thực hiện đọc tên file từ console và mở file đó với chế độ READWRITE bằng Syscall Open. Chương trình xóa file bằng Syscall Remove

Kết quả chạy chương trình:

```
kiet@kiet-virtual-machine:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x delete
File name to delete: abc.txt
Successfully delete file 'abc.txt'

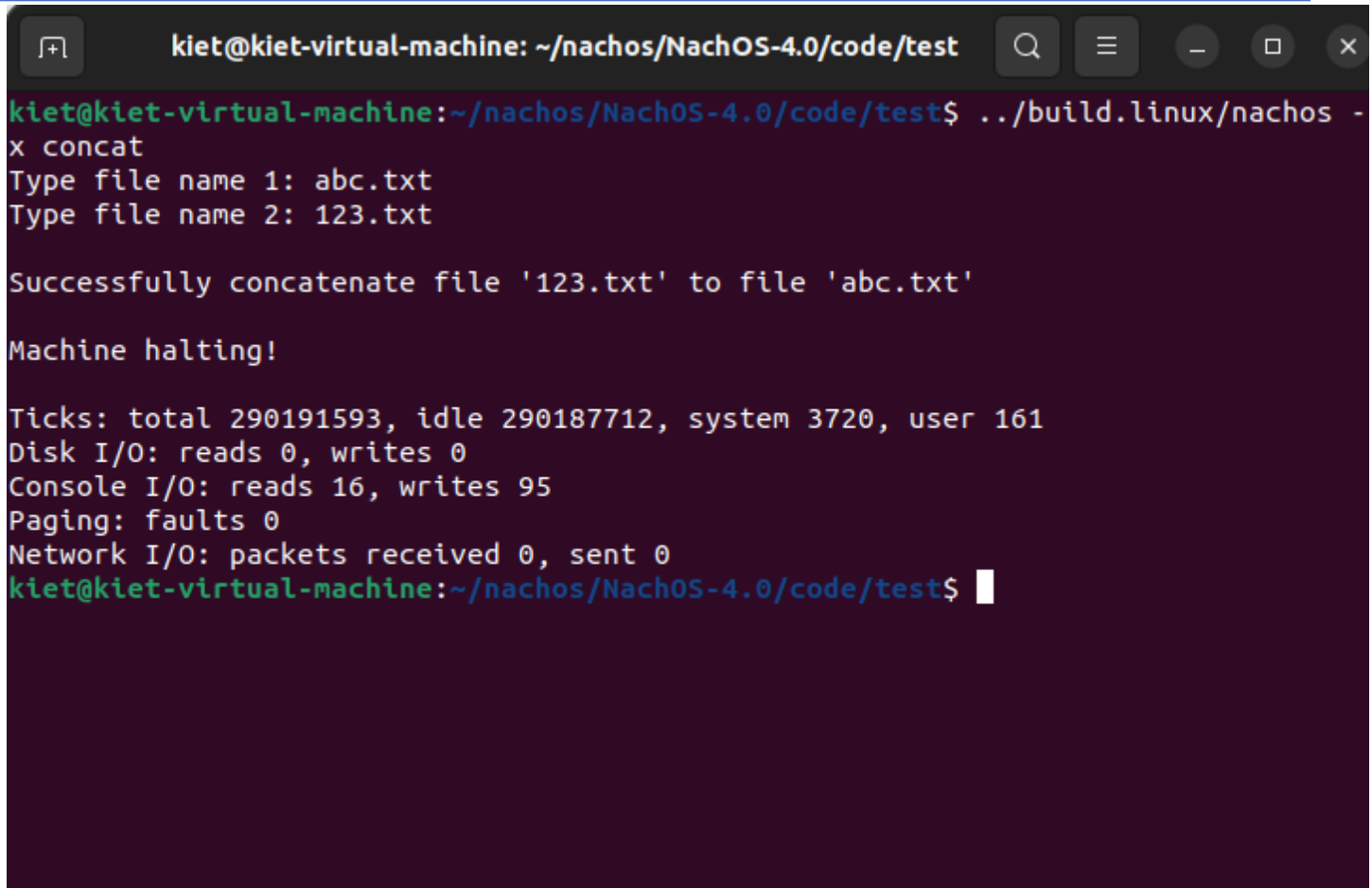
Machine halting!

Ticks: total 110111402, idle 110109212, system 2130, user 60
Disk I/O: reads 0, writes 0
Console I/O: reads 8, writes 56
Paging: faults 0
Network I/O: packets received 0, sent 0
kiet@kiet-virtual-machine:~/nachos/NachOS-4.0/code/test$
```

4.5 Concat:

Chương trình thực hiện đọc 2 tên file từ console và mở file đó với chế độ READWRITE bằng Syscall Open. Chương trình đọc 2 file bằng Syscall Read, sao chép nội dung file 2 và thêm vào file 1

Kết quả chạy chương trình:



```
kiet@kiet-virtual-machine: ~/nachos/NachOS-4.0/code/test
kiet@kiet-virtual-machine:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -
x concat
Type file name 1: abc.txt
Type file name 2: 123.txt

Successfully concatenate file '123.txt' to file 'abc.txt'

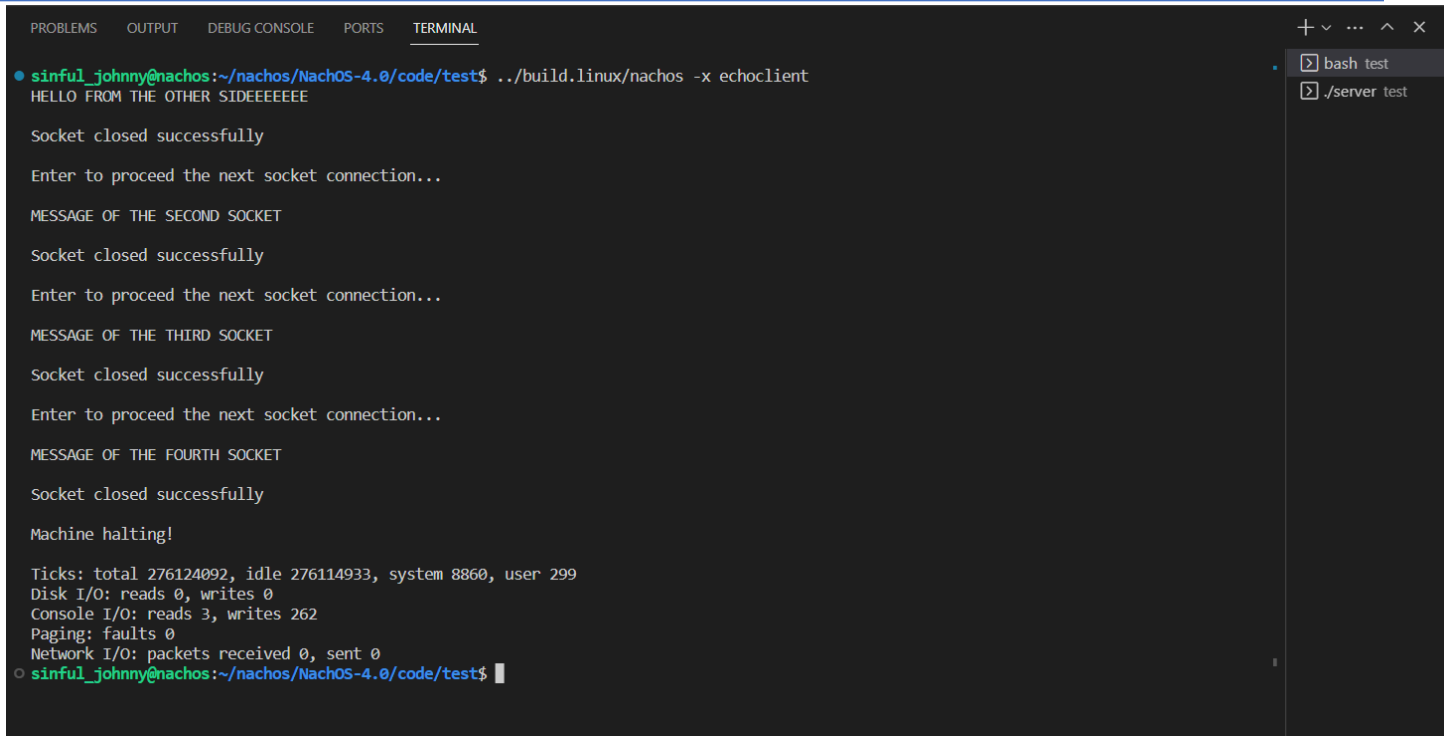
Machine halting!

Ticks: total 290191593, idle 290187712, system 3720, user 161
Disk I/O: reads 0, writes 0
Console I/O: reads 16, writes 95
Paging: faults 0
Network I/O: packets received 0, sent 0
kiet@kiet-virtual-machine:~/nachos/NachOS-4.0/code/test$
```

4.6 Echo:

Client dùng Syscall Connect để kết nối và gửi thông điệp qua 4 socket, đọc thông điệp bằng Syscall Read rồi đóng socket lại

- Kết quả chương trình:



The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL
• sinful_johnny@nuchos:~/nuchos/NachOS-4.0/code/test$ ../build.linux/nuchos -x echoclient
HELLO FROM THE OTHER SIDEEEEEEE

Socket closed successfully

Enter to proceed the next socket connection...

MESSAGE OF THE SECOND SOCKET

Socket closed successfully

Enter to proceed the next socket connection...

MESSAGE OF THE THIRD SOCKET

Socket closed successfully

Enter to proceed the next socket connection...

MESSAGE OF THE FOURTH SOCKET

Socket closed successfully

Machine halting!

Ticks: total 276124092, idle 276114933, system 8860, user 299
Disk I/O: reads 0, writes 0
Console I/O: reads 3, writes 262
Paging: faults 0
Network I/O: packets received 0, sent 0
○ sinful_johnny@nuchos:~/nuchos/NachOS-4.0/code/test$
```

On the right side of the terminal window, there is a sidebar with two tabs: "bash test" and "/server test".

4.7 File Transfer:

Client đọc file và chuyển file lên Server bằng Syscall Read và Connect thông qua socket, Server nhận thông điệp, chuyển thông điệp thành chữ in hoa và gửi lại cho Client

Kết quả chương trình:

HỆ ĐIỀU HÀNH – PROJECT: SYSTEM CALLS & FILE – NETWORK OPERATION

```
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL
● sinful_johnny@nachos:~/nachos/NachOS-4.0/code/test$ ../build.linux/nachos -x filetransfer

Start
Read buffer: hello there$

File read
HELLO THERE$

Echo
File written

Machine halting!

Ticks: total 10378, idle 7590, system 2570, user 218
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 76
Paging: faults 0
Network I/O: packets received 0, sent 0
○ sinful_johnny@nachos:~/nachos/NachOS-4.0/code/test$
```

```
code > test > abc.txt
1 hello there
```

```
code > test > 123.txt
1 HELLO THERE$FF
```