UNIVERSITY OF SCIENCE

VIETNAM NATIONAL UNIVERSITY

PROJECT 1

EXCEPTIONS AND SYSTEM CALLS

Lecturer

Le Giang Thanh

Nguyen Thanh Quan

Ho Chi Minh City, April 3rd, 2022

TABLE OF CONTENTS

1. INFORMATION

Lecturer

- Le Giang Thanh
- Nguyen Thanh Quan

Class: 20CLC08

Student

- 20127039 – Tran Dam Gia Huy
- 20127043 – Nguyen Thoai Dang Khoa
- 20127666 – Huynh Tan Vinh

The percentage of completion: 100%

Task	Question	Note	Point	Accomplish
1		Understand the NachOS code	1	X
2		Understand the design	0.5	X
3	1	Handle exceptions	0.5	X
	2	Increase PC variable	0.5	X
	3	ReadNum	0.5	X
	4	PrintNum	0.5	X
	5	ReadChar	0.5	X
	6	PrintChar	0.5	X
	7	RandomNum	0.5	X
	8	ReadString	0.5	X
	9	PrintString	0.5	X
	10,14	Create, Remove	0.5	X
	11	Open, Close	0.5	X
	12	Read	0.5	X
	13	Seek	0.5	X
	15,16,17	Help, ascii, sort	0.5	X
	18,19,20,21	Createfile, cat, copy, delete	0.5	X
		Do not allow users crash the OS	0.5	X
4		Report	0.5	X

2. CONTENT

2.1.Exception

- Rebuild the file *exception.cc* to handle all of exceptions that listed in *../code/machine/machine.h*

2.2.IncreasePC

- All of system calls need to increase program counter before they return the result
- If not, Nachos will be in infinite loop
- To implement, we save the current PC to previous PC, increase the current PC and save the current PC that increases one more time to next PC

2.3.ReadNum

Read an 32-bit integer number by syscall.

<u>Idea</u>

- Read a string by user input in kernel space
- Check condition
- Convert the string to an integer and return it to user space.

<u>Implement</u>

- Read each character by using GetChar() of synchConsoleIn and save to buffer in loop until match one of below cases:
  - Space, enter, end of file, bell, backspace, tab
- Invalid conditions:
  - String has more than 11 characters.
  - String has 10 characters and greater than 2147483647 (String > 2^31 - 1)
  - String has 11 characters and greater than 2147483648 (String < -2^31)
- Convert the string to an integer
  - Ensure that each character in string in range ‘0’ and ‘9’.
  - Convert string to int by mutiply the number by 10 and add with each character-48 in loop.
- Return the integer number by writing the value in register 2.

<u>Result</u>

- The result will be displayed with PrintNum syscall

2.4.PrintNum

Print an integer by syscall

Idea

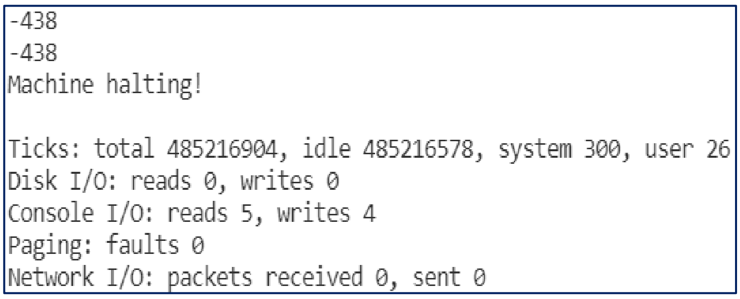
- Read an integer in register
- Convert an integer to string
- Display each character in the string consecutively by loop.

Implement

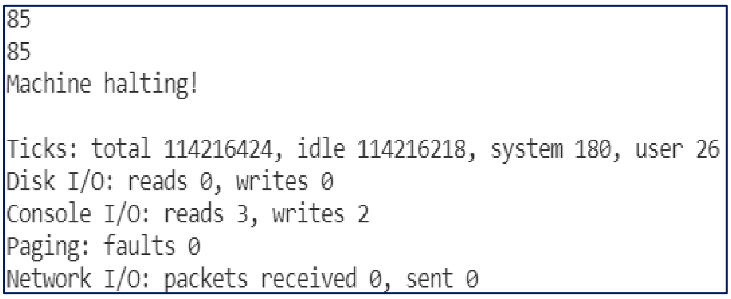
- Read an integer in register 4
- Convert an integer to string:
  - Put the last digit in the number into buffer by absolute moding 10 and add with 48 in each loop
  - Remove the last digit of the number by dividing 10.
  - Put '-' into last element in buffer if the number is negative.
- Loop in reverse of buffer and use PutChar(character) of synchConsoleOut to display each element in buffer.

Result

- **Test case 1**: Negative and positive number (Output 0 if invalid)

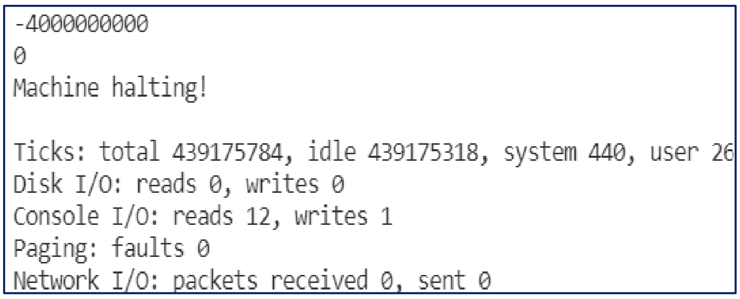


```
-438
-438
Machine halting!

Ticks: total 485216904, idle 485216578, system 300, user 26
Disk I/O: reads 0, writes 0
Console I/O: reads 5, writes 4
Paging: faults 0
Network I/O: packets received 0, sent 0
```

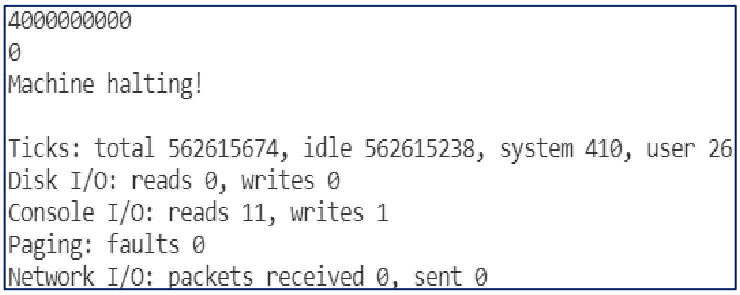


```
85
85
Machine halting!

Ticks: total 114216424, idle 114216218, system 180, user 26
Disk I/O: reads 0, writes 0
Console I/O: reads 3, writes 2
Paging: faults 0
Network I/O: packets received 0, sent 0
```

- **Test case 2**: Large negative and postive number (Output 0 if invalid)

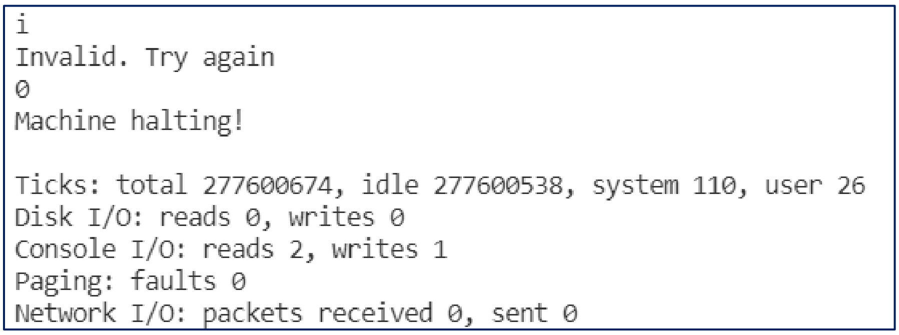


```
-4000000000
0
Machine halting!

Ticks: total 439175784, idle 439175318, system 440, user 26
Disk I/O: reads 0, writes 0
Console I/O: reads 12, writes 1
Paging: faults 0
Network I/O: packets received 0, sent 0
```



```
4000000000
0
Machine halting!

Ticks: total 562615674, idle 562615238, system 410, user 26
Disk I/O: reads 0, writes 0
Console I/O: reads 11, writes 1
Paging: faults 0
Network I/O: packets received 0, sent 0
```

- **Test case 3**: Character (Output 0 if invalid)



```
i
Invalid. Try again
0
Machine halting!

Ticks: total 277600674, idle 277600538, system 110, user 26
Disk I/O: reads 0, writes 0
Console I/O: reads 2, writes 1
Paging: faults 0
Network I/O: packets received 0, sent 0
```

2.5.ReadChar

Read a character by syscall

Idea

- Read a character in kernel space
- Return it to user space

Implement

- Use GetChar() of synchConsoleIn to read an character
- Return the character by writing the value in register 2

Result

- The result will be displayed with PrintChar

2.6.PrintChar

Print a character by syscall

Idea

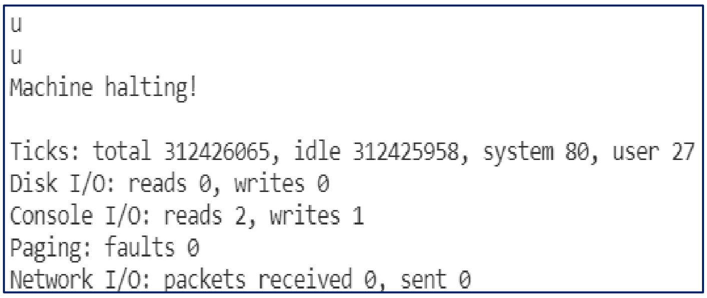
- Read a character in register
- Display a character

Implement

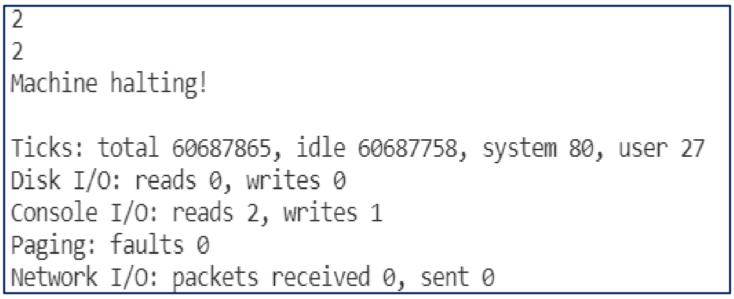
- Read a character in register 4
- Use PutChar(character) of synchConsoleOut to display it

Result

- **Test case 1**: Enter a character

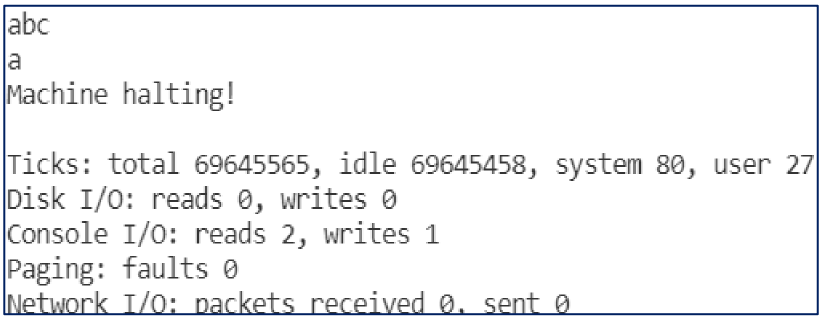


```
u
u
Machine halting!

Ticks: total 312426065, idle 312425958, system 80, user 27
Disk I/O: reads 0, writes 0
Console I/O: reads 2, writes 1
Paging: faults 0
Network I/O: packets received 0, sent 0
```



```
2
2
Machine halting!

Ticks: total 60687865, idle 60687758, system 80, user 27
Disk I/O: reads 0, writes 0
Console I/O: reads 2, writes 1
Paging: faults 0
Network I/O: packets received 0, sent 0
```

- **Test case 2:** Enter more than 1 character



```
abc
a
Machine halting!

Ticks: total 69645565, idle 69645458, system 80, user 27
Disk I/O: reads 0, writes 0
Console I/O: reads 2, writes 1
Paging: faults 0
Network I/O: packets received 0, sent 0
```

2.7.RandomNum

Create a random number by syscall

Idea

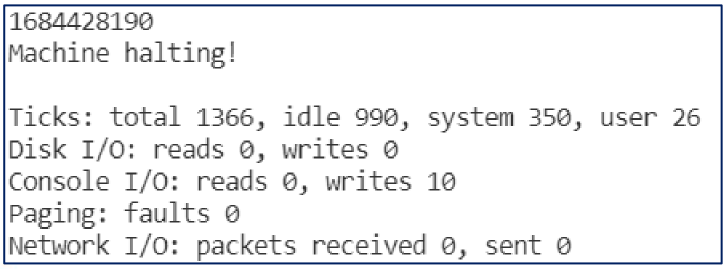
- Random a number
- Return it to user space

Implement

- Use rand() library to create a random number
- Return the random number by writing it in register 2

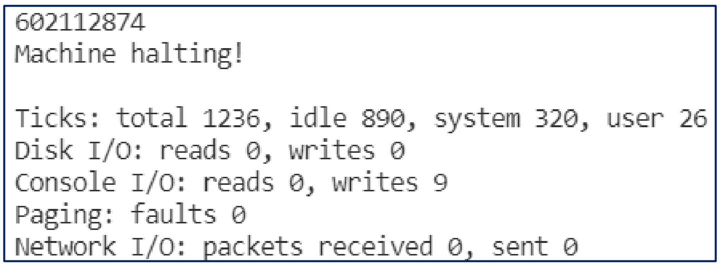
Result

- **Test case 1**



```
1684428190
Machine halting!

Ticks: total 1366, idle 990, system 350, user 26
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 10
Paging: faults 0
Network I/O: packets received 0, sent 0
```

- **Test case 2**



```
602112874
Machine halting!

Ticks: total 1236, idle 890, system 320, user 26
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 9
Paging: faults 0
Network I/O: packets received 0, sent 0
```

2.8.ReadString

Read a string by syscall

Idea

- Read an array of character in kernel space
- Return the array to user space

Implement

- Use the buffer to store a number of characters that user puts into by GetChar() of synchConsoleIn with loop
- Add '\0' in the last element of buffer to make an ending signal.
- Because of the buffer in user space, but when we input the string, the data will be stored in kernel space so that we need to make a function to move data from kernel space to user space
    - The function System2User(…) uses kernel->machine->WriteMem(…) to move data from string to the address of buffer that read in register 4
    - Write '\0' in the last address of buffer that read in register 4 to end the string.
- Delete the data in buffer to avoid leaking memory.

Result

- The result will be displayed with PrintString

2.9.PrintString

Print a string by syscall

Idea

- Read an integer storing the address of string

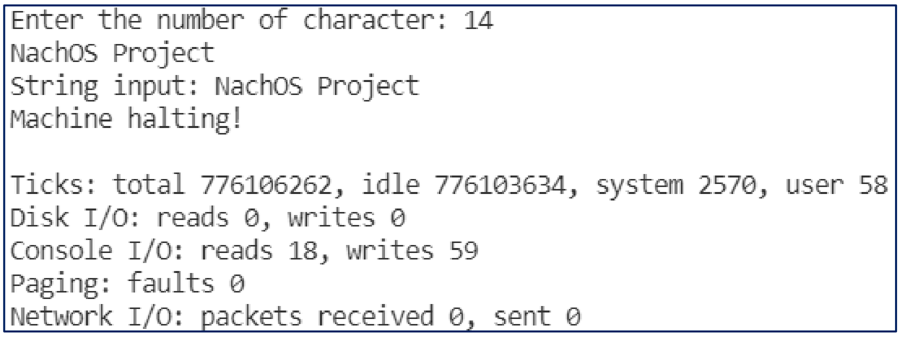
- Move data to kernel space and display it

Implement

- Read an integer stores the address of string in register 4
- Move data from the address of character buffer in register 4 to kernel space
  - The function User2System(…) uses kernel->machine->ReadMem(…) to read data from the address of buffer and moves it to buffer in kernel space.
- Using PutChar(character) of synchConsoleOut to display each element in buffer.

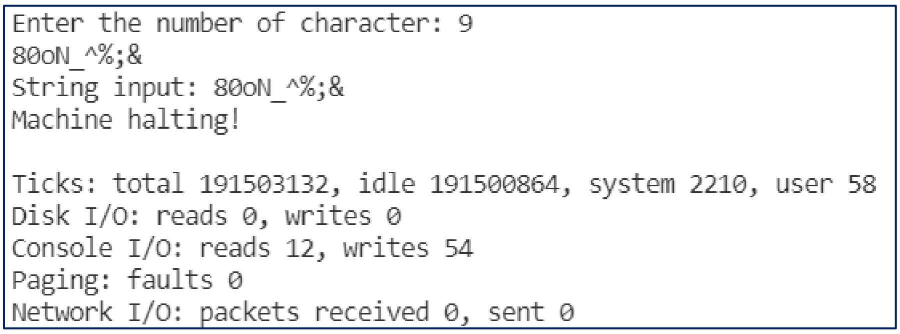
Result

- **Test case 1**



```
Enter the number of character: 14
NachOS Project
String input: NachOS Project
Machine halting!

Ticks: total 776106262, idle 776103634, system 2570, user 58
Disk I/O: reads 0, writes 0
Console I/O: reads 18, writes 59
Paging: faults 0
Network I/O: packets received 0, sent 0
```

- **Test case 2**



```
Enter the number of character: 9
80oN_^%;&
String input: 80oN_^%;&
Machine halting!

Ticks: total 191503132, idle 191500864, system 2210, user 58
Disk I/O: reads 0, writes 0
Console I/O: reads 12, writes 54
Paging: faults 0
Network I/O: packets received 0, sent 0
```

2.10. Create, Remove

Create

Create a empty file by file name by syscall

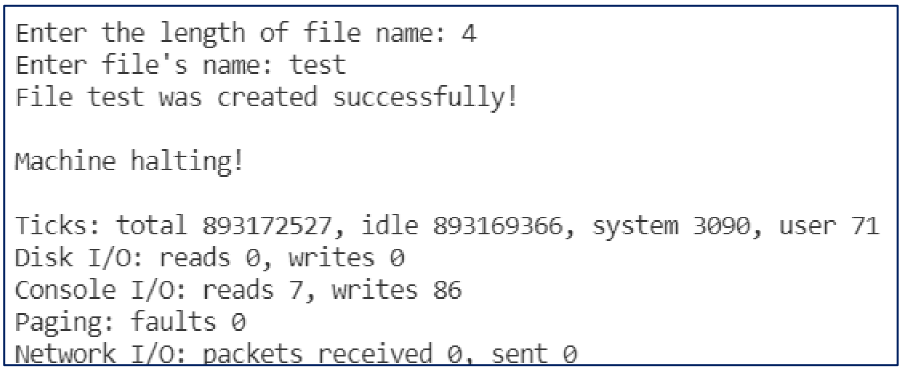
Idea

- Read file name from user
- Create a file with name

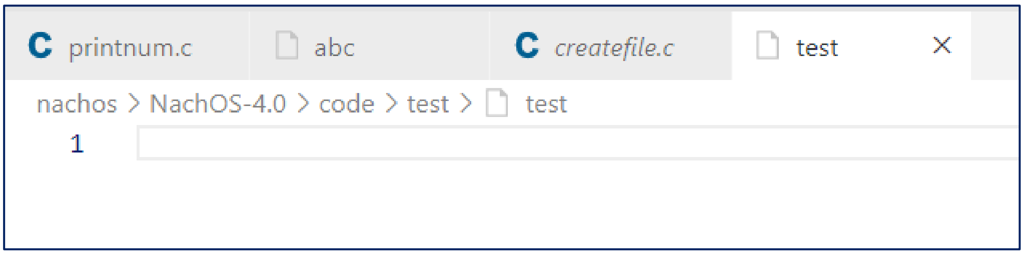
Implement

- Read an integer stores the address of string in register 4
- Move data from the address of character buffer in register 4 to kernel space
  - The function User2System(…) uses kernel->machine->ReadMem(…) to read data from the address of buffer and moves it to buffer in kernel space.
- Use Create(fileName) of kerner->fileSystem to create an empty file.
- Return 0 (Write to register 2) if file is created successfully, return -1 in case of failure
- Delete buffer allocated to avoid memory leak

Result



```
Enter the length of file name: 4
Enter file's name: test
File test was created successfully!

Machine halting!

Ticks: total 893172527, idle 893169366, system 3090, user 71
Disk I/O: reads 0, writes 0
Console I/O: reads 7, writes 86
Paging: faults 0
Network I/O: packets received 0, sent 0
```



Remove

Remove a file by file name by syscall

Idea

- Read file name from user
- Remove file

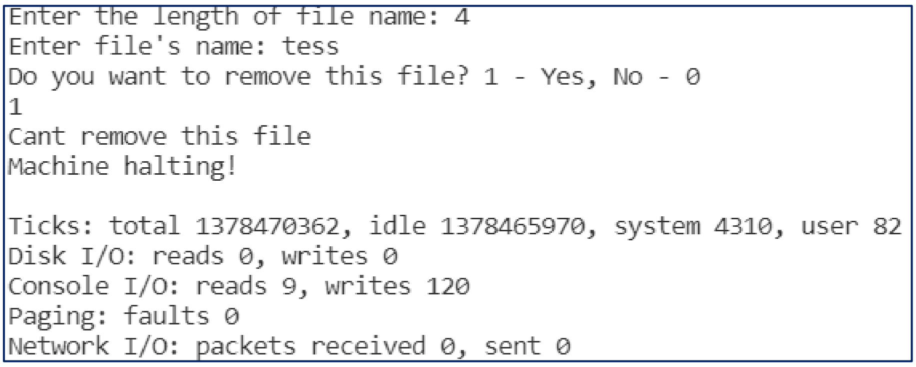
Implement

- Read an integer stores the address of string in register 4
- Move data from the address of character buffer in register 4 to kernel space
  - The function User2System(…) uses kernel->machine->ReadMem(…) to read data from the address of buffer and moves it to buffer in kernel space.
- Use Remove(fileName) of kernel->fileSystem to remove file

- Delete buffer allocated to avoid memory leak

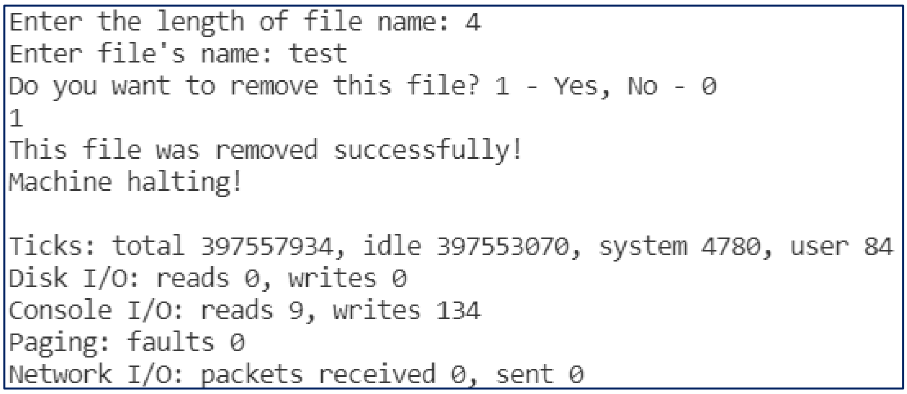
Result

- **Test case 1**: File name does not exist



```
Enter the length of file name: 4
Enter file's name: tess
Do you want to remove this file? 1 - Yes, No - 0
1
Cant remove this file
Machine halting!

Ticks: total 1378470362, idle 1378465970, system 4310, user 82
Disk I/O: reads 0, writes 0
Console I/O: reads 9, writes 120
Paging: faults 0
Network I/O: packets received 0, sent 0
```

- **Test case 2**: File name is valid



```
Enter the length of file name: 4
Enter file's name: test
Do you want to remove this file? 1 - Yes, No - 0
1
This file was removed successfully!
Machine halting!

Ticks: total 397557934, idle 397553070, system 4780, user 84
Disk I/O: reads 0, writes 0
Console I/O: reads 9, writes 134
Paging: faults 0
Network I/O: packets received 0, sent 0
```

2.11. Open, Close

OpenFileID

Open file by file name by syscall

Idea

- Read file name from user
- Create a file with name
- Return id of file name

Implement

- Read an integer stores the address of string in register 4

- Move data from the address of character buffer in register 4 to kernel space
  - The function User2System(…) uses kernel->machine->ReadMem(…) to read data from the address of buffer and moves it to buffer in kernel space.
- User openFile(fileName) of kerner->fileSystem to open file and return id of file.
- id=-1 if it is fail in some cases:
  - File table is full
  - Can not create sector so that can not create file
- Write id of file to register 2 to return it to user
- Delete buffer allocated to avoid memory leak

Result

- The result will be displayed with Seek

CloseFileID

Close file by id of file by syscall

Idea

- Read id from user
- Close file

Implement

- Read an integer id of file in register 4
- Use Close(id) of kernel->fileSystem to close file
- Write the state of close file to register 2 (state=0 if it’s success, state=-1 if it’s fail)

Result

- The result will be displayed with Seek

2.12.Read, Write

Read

Read a file by file name to string by syscall

Idea

- Read fileName, number of bytes read and id by user
- Check condition if it’s avaible to read
- Read file and save to array character

Implement

- Read an integer storing the address of string in register 4
- Read an integer storing the number of bytes in register 5
- Read an integer storing the id of file in register 6
- Check condition:
  - If (id=0) it means using ConsoleInput, input the string and save to buffer
  - Else if (id=1) it means using ConsoleOutput, do nothing after that
  - Else, use Read(buffer,numBytes,id) of kernel->fileSystem to read file and save to buffer
- In case of id=0 or (id!=1 id!=0):
  - Return the number of bytes read to user
  - Because of the buffer in user space, but when we input the string, the data will be stored in kernel space so that we need to make a function to move data from kernel space to user space
    - The function System2User(…) uses kernel->machine->WriteMem(…) to move data from string to the address of buffer that read in register 4
- Delete buffer allocated to avoid memory leak

Result

- The result will be displayed with Write

Write

Write a string to file by file name by syscall

Idea

- Read fileName, number of bytes read and id by user
- Check condition if it's avaible to write
- Write to file from string

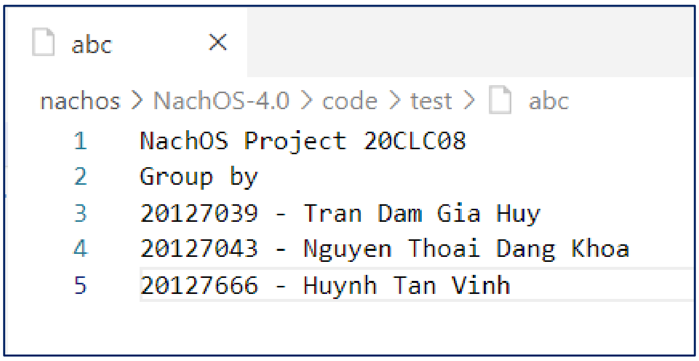
Implement

- Read an integer storing the address of string in register 4
- Move data from the address of character buffer in register 4 to kernel space
  - The function User2System(…) uses kernel->machine->ReadMem(…) to read data from the address of buffer and moves it to buffer in kernel space.
- Read an integer storing the number of bytes in register 5
- Read an integer storing the id of file in register 6
- Check condition:
  - If (id=1) it means using ConsoleOutput, display the string and return the number of bytes display to user

  - Else if (id=0) it means using ConsoleInput, do nothing after that
  - Else, use Write(buffer,numBytes,id) of kernel->fileSystem to write to file.
- In case of or (id!=1 and id!=0) :
  - Write to file and return the number of bytes wrote to user
- Delete buffer allocated to avoid memory leak

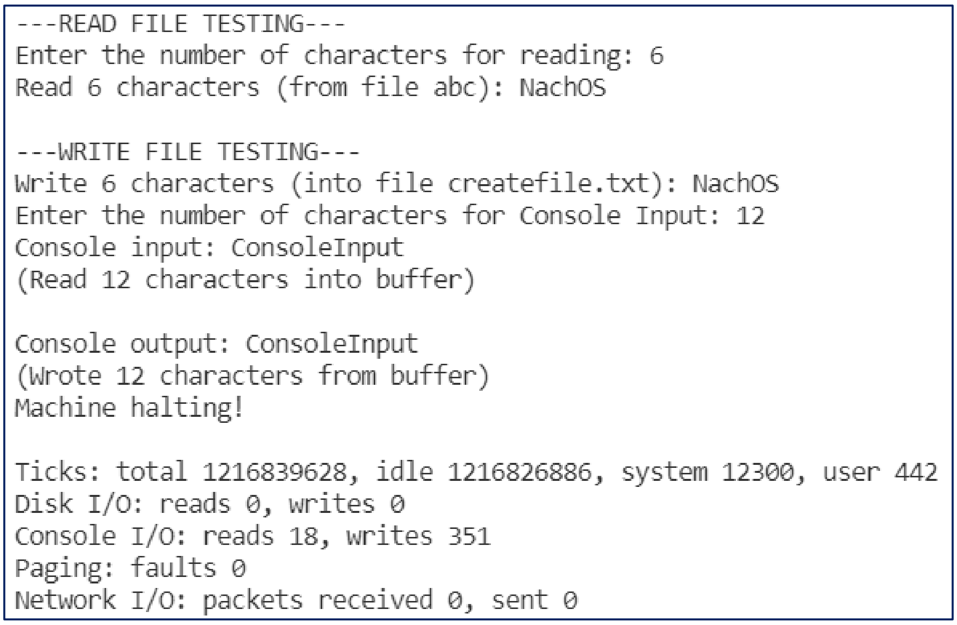
Result

The program we designed that read "abc.txt"

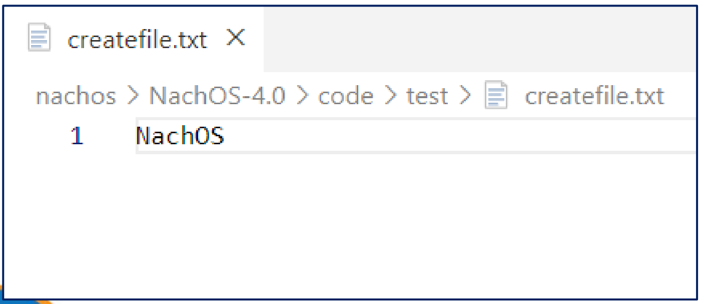


```
NachOS Project 20CLC08
Group by
20127039 - Tran Dam Gia Huy
20127043 - Nguyen Thoai Dang Khoa
20127666 - Huynh Tan Vinh
```

The result including handle ConsoleIO at 0, 1 in file table



```
---READ FILE TESTING---
Enter the number of characters for reading: 6
Read 6 characters (from file abc): NachOS

---WRITE FILE TESTING---
Write 6 characters (into file createfile.txt): NachOS
Enter the number of characters for Console Input: 12
Console input: ConsoleInput
(Read 12 characters into buffer)

Console output: ConsoleInput
(Wrote 12 characters from buffer)
Machine halting!

Ticks: total 1216839628, idle 1216826886, system 12300, user 442
Disk I/O: reads 0, writes 0
Console I/O: reads 18, writes 351
Paging: faults 0
Network I/O: packets received 0, sent 0
```



```
NachOS
```

2.13. Seek

Move the pointer of file to the position that user input by syscall

Idea

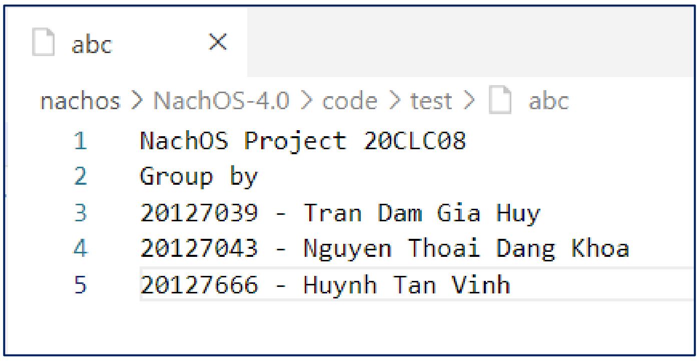
- Read position, id from user
- Check condition if it’s avaible to move
- Move the pointer and return the current postion of pointer to user

Implement

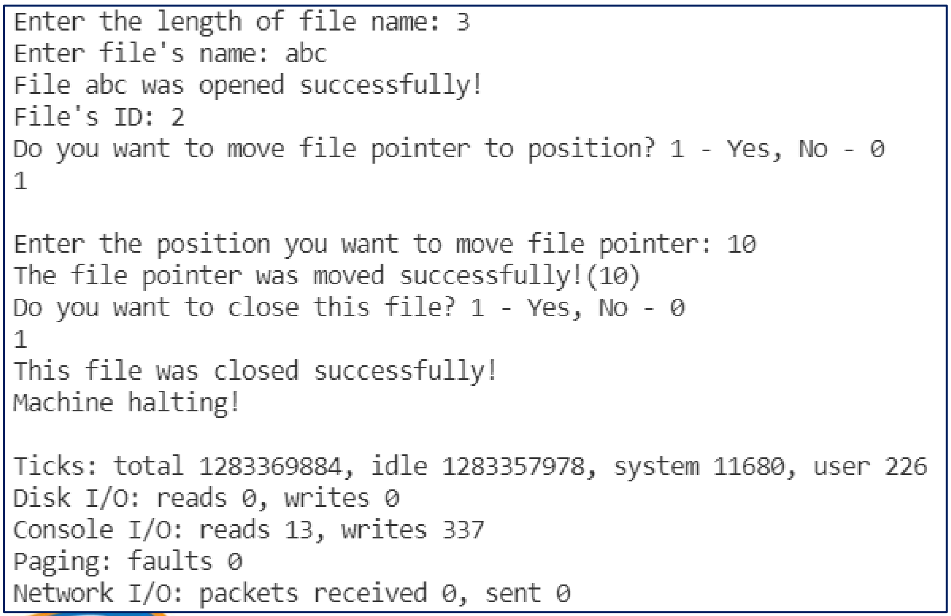
- Read an integer storing the position in register 4
- Read an integer storing the id of file in register 5
- Check condition:
  - If id=1 or id=0, it means ConsoleIO that can not seek and return -1 to user
  - Else, using Seek(position, id) of kernel->fileSystem to move pointer to position and return the current postion of pointer to user

Result

- **Test case 1**: Open file “abc.txt” and move pointer at position 10 and close this file

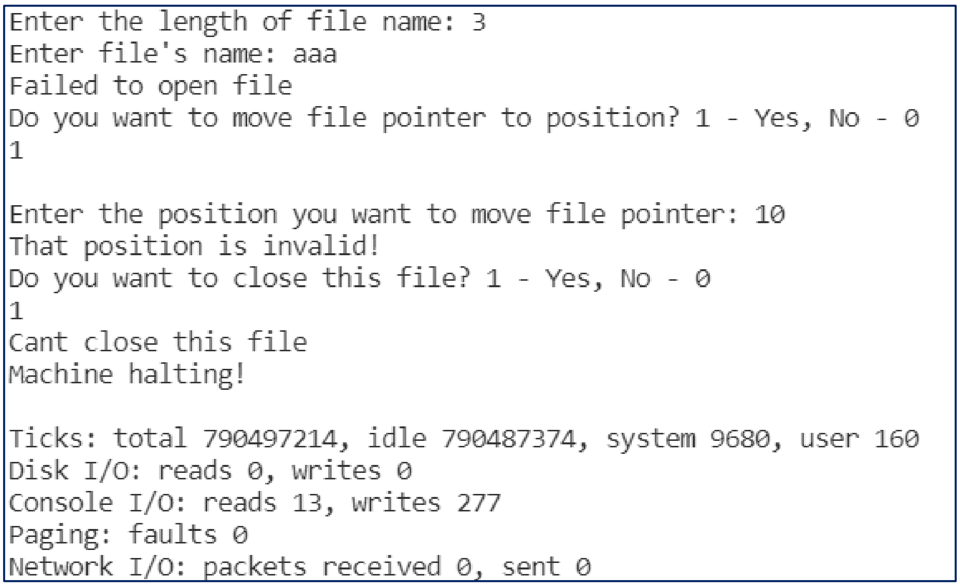


```
NachOS Project 20CLC08
Group by
20127039 - Tran Dam Gia Huy
20127043 - Nguyen Thoai Dang Khoa
20127666 - Huynh Tan Vinh
```



```
Enter the length of file name: 3
Enter file's name: abc
File abc was opened successfully!
File's ID: 2
Do you want to move file pointer to position? 1 - Yes, No - 0
1

Enter the position you want to move file pointer: 10
The file pointer was moved successfully!(10)
Do you want to close this file? 1 - Yes, No - 0
1
This file was closed successfully!
Machine halting!

Ticks: total 1283369884, idle 1283357978, system 11680, user 226
Disk I/O: reads 0, writes 0
Console I/O: reads 13, writes 337
Paging: faults 0
Network I/O: packets received 0, sent 0
```

- **Test case 2**: Open file “aaa.txt” that does not exist



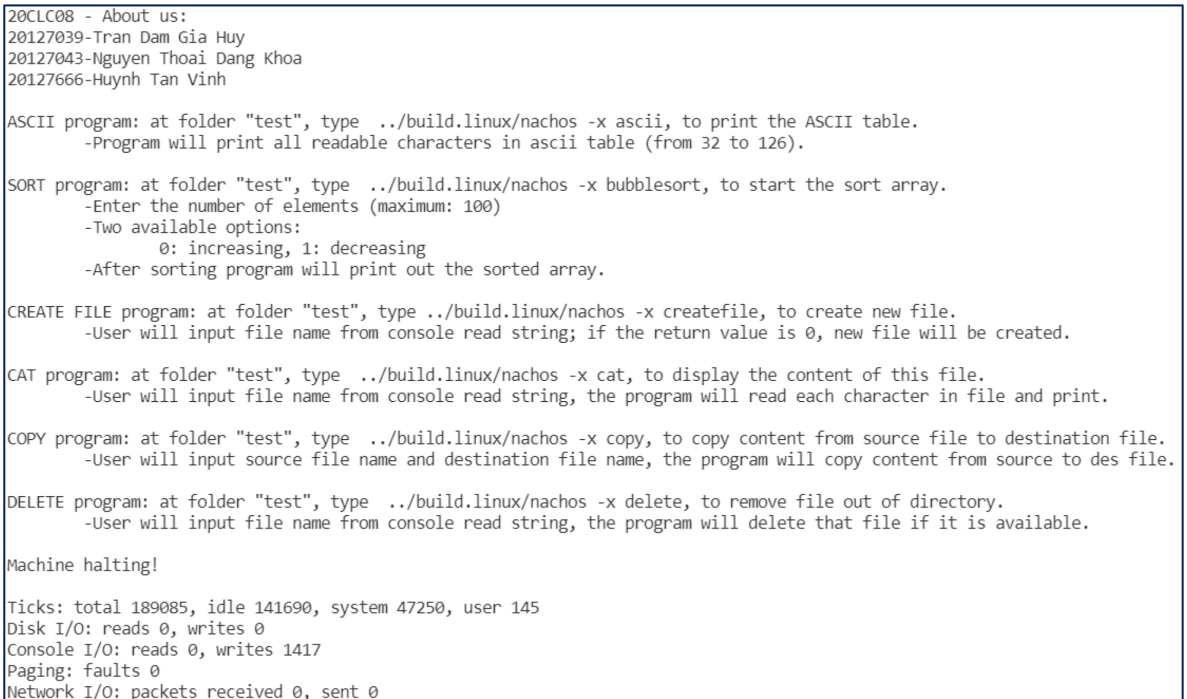
```
Enter the length of file name: 3
Enter file's name: aaa
Failed to open file
Do you want to move file pointer to position? 1 - Yes, No - 0
1

Enter the position you want to move file pointer: 10
That position is invalid!
Do you want to close this file? 1 - Yes, No - 0
1
Cant close this file
Machine halting!

Ticks: total 790497214, idle 790487374, system 9680, user 160
Disk I/O: reads 0, writes 0
Console I/O: reads 13, writes 277
Paging: faults 0
Network I/O: packets received 0, sent 0
```

2.14. Help, ascii, sort

Help

- Use PrintString(string) to print out the group of students and the brief description of ascii, sort

Result

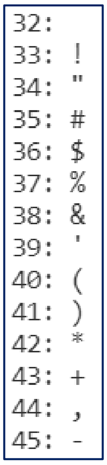


```
20CLC08 - About us:
20127039-Tran Dam Gia Huy
20127043-Nguyen Thoai Dang Khoa
20127666-Huynh Tan Vinh

ASCII program: at folder "test", type  ../build.linux/nachos -x ascii, to print the ASCII table.
        -Program will print all readable characters in ascii table (from 32 to 126).

SORT program: at folder "test", type  ../build.linux/nachos -x bubblesort, to start the sort array.
        -Enter the number of elements (maximum: 100)
        -Two available options:
                0: increasing, 1: decreasing
        -After sorting program will print out the sorted array.

CREATE FILE program: at folder "test", type ../build.linux/nachos -x createfile, to create new file.
        -User will input file name from console read string; if the return value is 0, new file will be created.

CAT program: at folder "test", type  ../build.linux/nachos -x cat, to display the content of this file.
        -User will input file name from console read string, the program will read each character in file and print.

COPY program: at folder "test", type  ../build.linux/nachos -x copy, to copy content from source file to destination file.
        -User will input source file name and destination file name, the program will copy content from source to des file.

DELETE program: at folder "test", type  ../build.linux/nachos -x delete, to remove file out of directory.
        -User will input file name from console read string, the program will delete that file if it is available.

Machine halting!

Ticks: total 189085, idle 141690, system 47250, user 145
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 1417
Paging: faults 0
Network I/O: packets received 0, sent 0
```

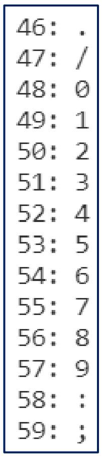
Ascii

- Use PrintNum(number) to print numbers from 32 to 126
- Use PrintChar(character) to print all readable characters in ascii table

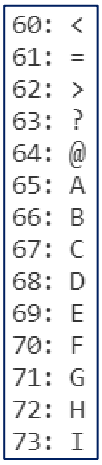
Result



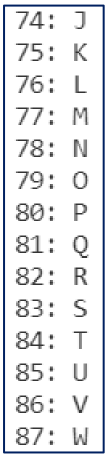
```
32:
33: !
34: "
35: #
36: $
37: %
38: &
39: '
40: (
41: )
42: *
43: +
44: ,
45: -
```



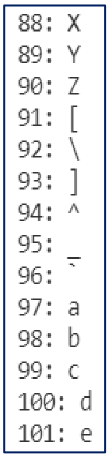
```
46: .
47: /
48: 0
49: 1
50: 2
51: 3
52: 4
53: 5
54: 6
55: 7
56: 8
57: 9
58: :
59: ;
```



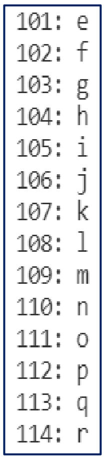
```
60: <
61: =
62: >
63: ?
64: @
65: A
66: B
67: C
68: D
69: E
70: F
71: G
72: H
73: I
```



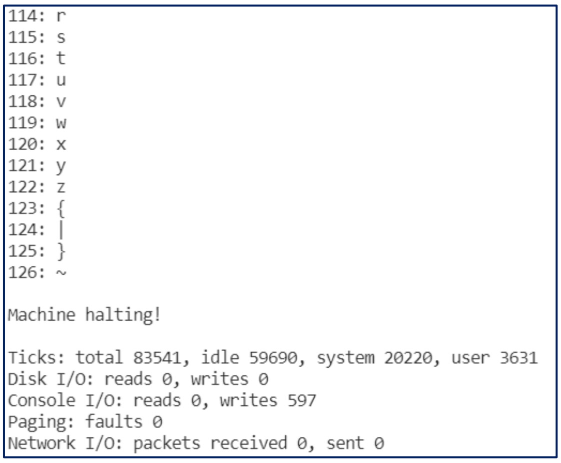
```
74: J
75: K
76: L
77: M
78: N
79: O
80: P
81: Q
82: R
83: S
84: T
85: U
86: V
87: W
```



```
88: X
89: Y
90: Z
91: [
92: \
93: ]
94: ^
95: _
96: `
97: a
98: b
99: c
100: d
101: e
```



```
101: e
102: f
103: g
104: h
105: i
106: j
107: k
108: l
109: m
110: n
111: o
112: p
113: q
114: r
```



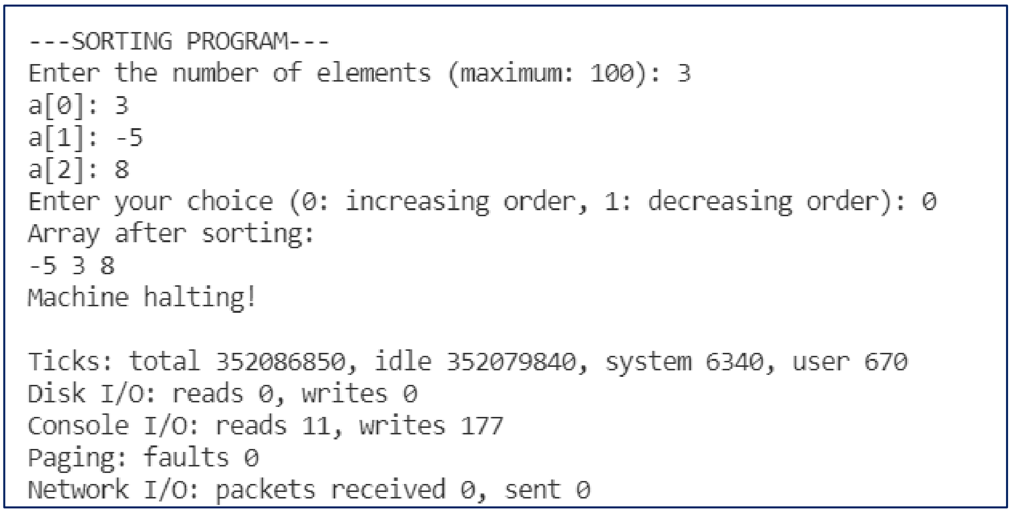
```
114: r
115: s
116: t
117: u
118: v
119: w
120: x
121: y
122: z
123: {
124: |
125: }
126: ~

Machine halting!

Ticks: total 83541, idle 59690, system 20220, user 3631
Disk I/O: reads 0, writes 0
Console I/O: reads 0, writes 597
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Sort

- ReadNum the number of elements and the value of each element
- ReadNum choice, choice=0 is increase, choice=1 is decrease
- Implement bubble sort algorithm

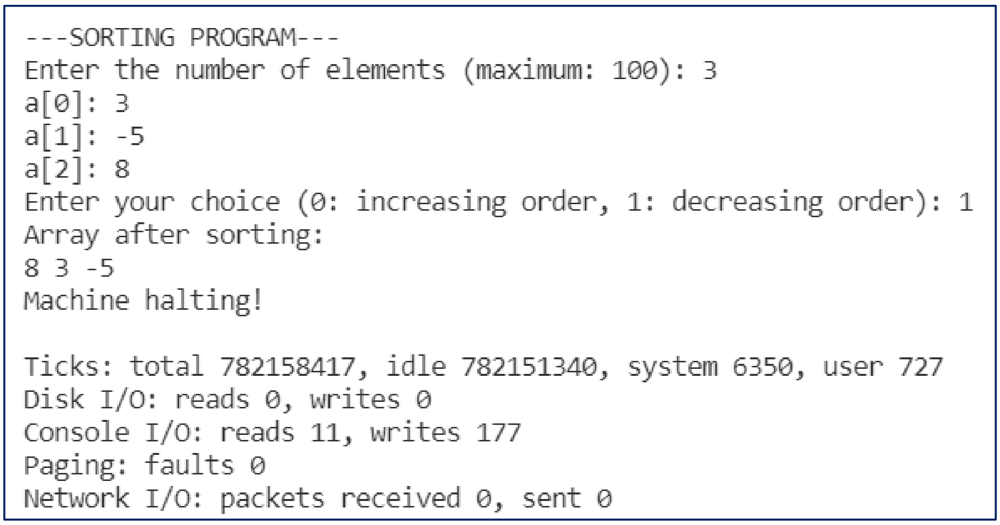
Result

Increasing



```
---SORTING PROGRAM---
Enter the number of elements (maximum: 100): 3
a[0]: 3
a[1]: -5
a[2]: 8
Enter your choice (0: increasing order, 1: decreasing order): 0
Array after sorting:
-5 3 8
Machine halting!

Ticks: total 352086850, idle 352079840, system 6340, user 670
Disk I/O: reads 0, writes 0
Console I/O: reads 11, writes 177
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Decreasing



```
---SORTING PROGRAM---
Enter the number of elements (maximum: 100): 3
a[0]: 3
a[1]: -5
a[2]: 8
Enter your choice (0: increasing order, 1: decreasing order): 1
Array after sorting:
8 3 -5
Machine halting!

Ticks: total 782158417, idle 782151340, system 6350, user 727
Disk I/O: reads 0, writes 0
Console I/O: reads 11, writes 177
Paging: faults 0
Network I/O: packets received 0, sent 0
```

2.15. Createfile, cat, copy

Create file

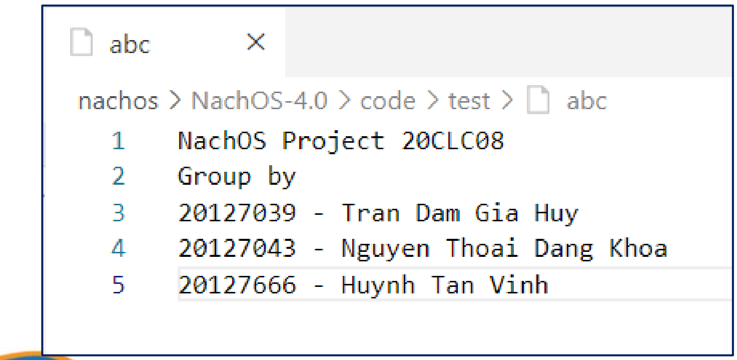
- Input length of file name and file name from user
- Create(fileName), if success display(fileName + "was created successfully"), else display("Error create file")

Cat

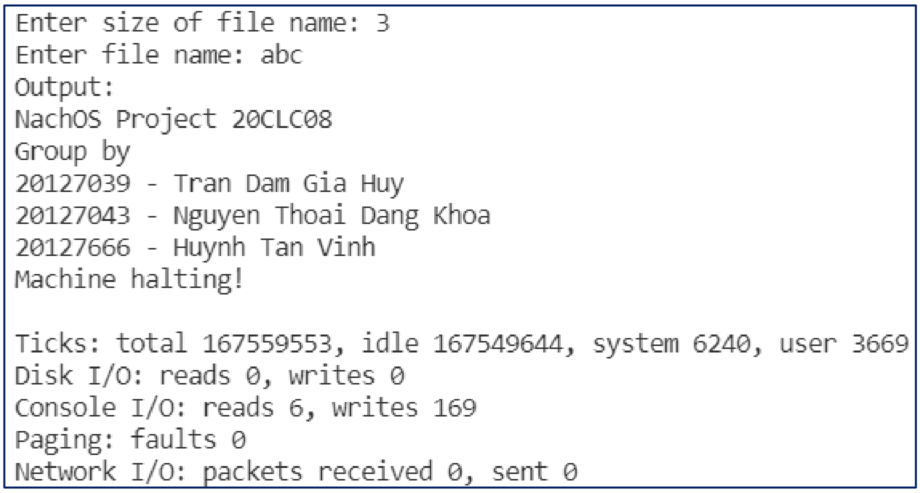
- Input length of file name and file name from user
- Calculate the length of file by move the pointer to end of file
- Read each character in file and display it by loop until reaches the length of file

Result

Read file "abc.txt" and display it



```
NachOS Project 20CLC08
Group by
20127039 - Tran Dam Gia Huy
20127043 - Nguyen Thoai Dang Khoa
20127666 - Huynh Tan Vinh
```



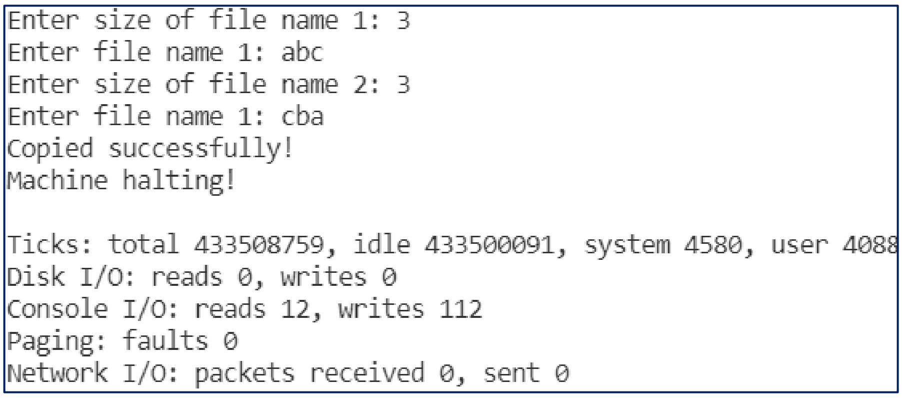
```
Enter size of file name: 3
Enter file name: abc
Output:
NachOS Project 20CLC08
Group by
20127039 - Tran Dam Gia Huy
20127043 - Nguyen Thoai Dang Khoa
20127666 - Huynh Tan Vinh
Machine halting!

Ticks: total 167559553, idle 167549644, system 6240, user 3669
Disk I/O: reads 0, writes 0
Console I/O: reads 6, writes 169
Paging: faults 0
Network I/O: packets received 0, sent 0
```

Copy

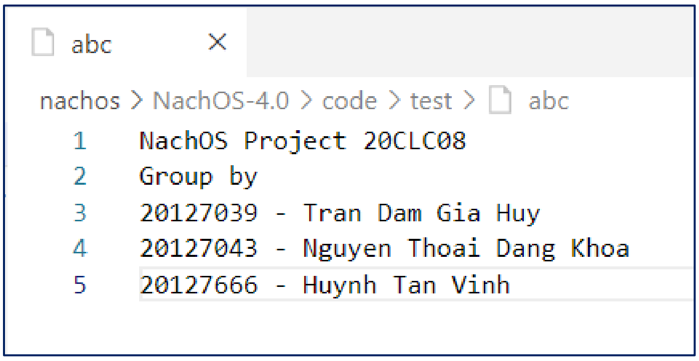
- Input length of file name and file name of source file and destination file from user
- Calculate the length of source file by move the pointer to end of source file
- Read each character in source file and write it to destination file by loop until reaches the length of source file

Result

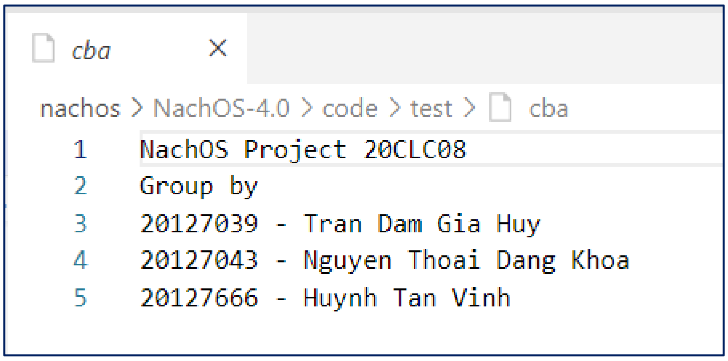
Read file “abc.txt” and copy to file “cba.txt”



```
Enter size of file name 1: 3
Enter file name 1: abc
Enter size of file name 2: 3
Enter file name 1: cba
Copied successfully!
Machine halting!

Ticks: total 433508759, idle 433500091, system 4580, user 4088
Disk I/O: reads 0, writes 0
Console I/O: reads 12, writes 112
Paging: faults 0
Network I/O: packets received 0, sent 0
```



```
NachOS Project 20CLC08
Group by
20127039 - Tran Dam Gia Huy
20127043 - Nguyen Thoai Dang Khoa
20127666 - Huynh Tan Vinh
```



```
NachOS Project 20CLC08
Group by
20127039 - Tran Dam Gia Huy
20127043 - Nguyen Thoai Dang Khoa
20127666 - Huynh Tan Vinh
```

3. REFERENCES

- References that lecturers provide on moodle
- https://github.com/nguyenthanhchungfit/Nachos-Programing-HCMUS