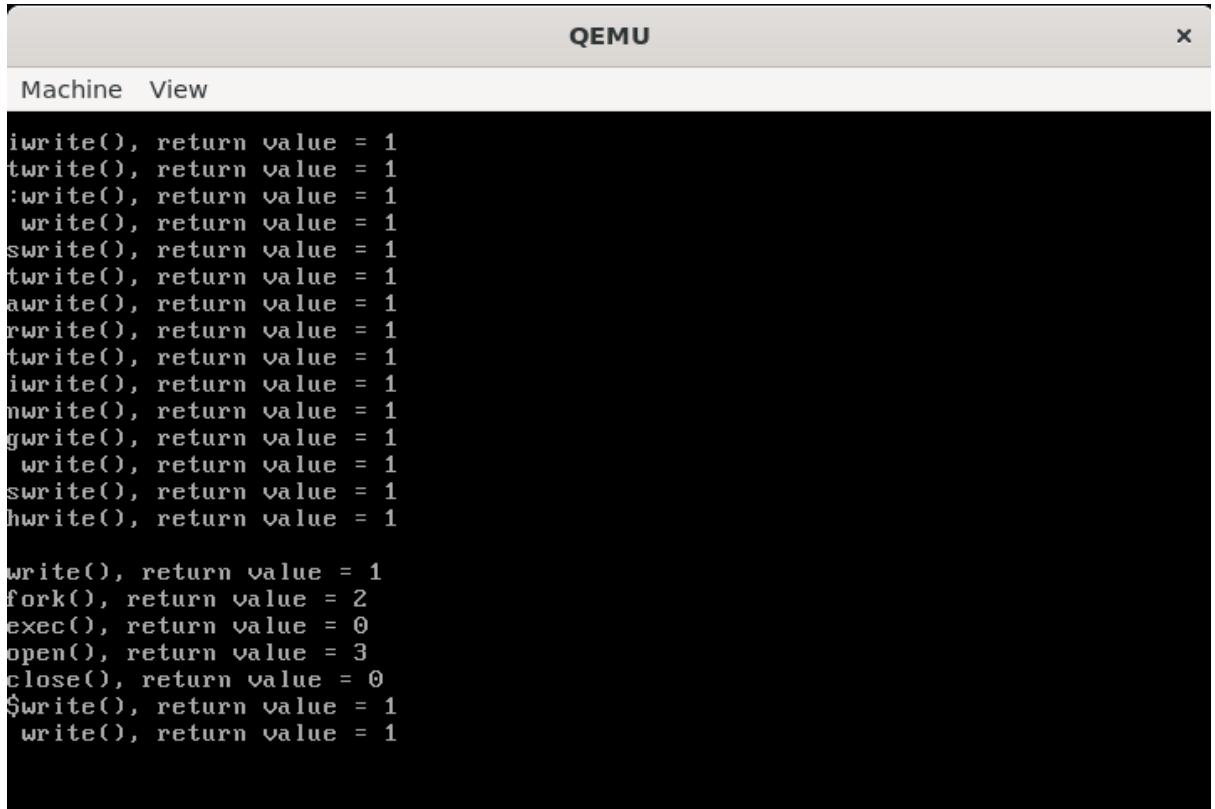


# Assignment – 5

- 1) Print out a line for each system call invocation

Ans- Output



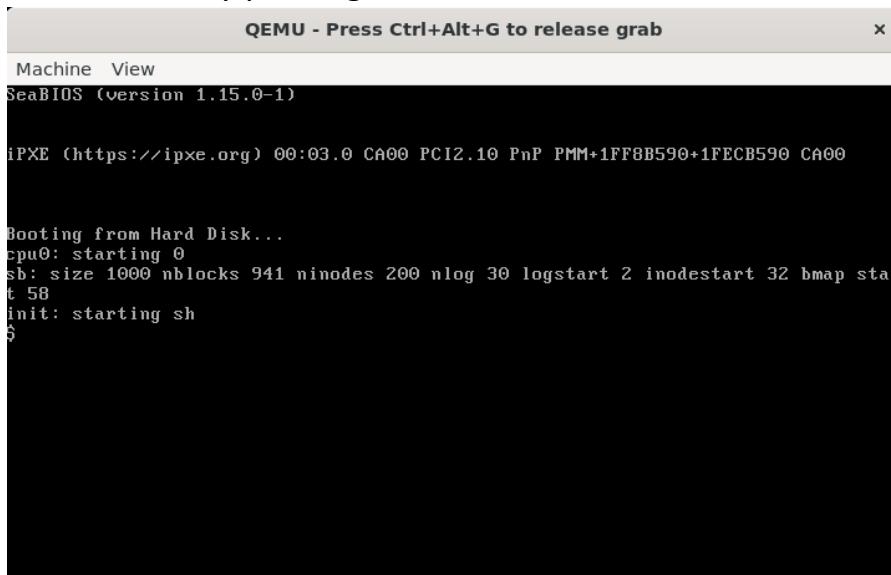
The screenshot shows a terminal window titled "QEMU" with the title bar "Machine View". The terminal output displays numerous system call invocations, all returning a value of 1. The calls listed include iwrite(), twrite(), :write(), \_write(), swrite(), twrite(), awrite(), rwrite(), twrite(), iwrite(), nwrite(), gwrite(), \_write(), swrite(), and hwrite(). Below these, standard file operations like write(), fork(), exec(), open(), close(), \$write(), and \_write() are also shown.

```
iwrite(), return value = 1
twrite(), return value = 1
:write(), return value = 1
_write(), return value = 1
swrite(), return value = 1
twrite(), return value = 1
awrite(), return value = 1
rwrite(), return value = 1
twrite(), return value = 1
iwrite(), return value = 1
nwrite(), return value = 1
gwrite(), return value = 1
_write(), return value = 1
swrite(), return value = 1
hwrite(), return value = 1

write(), return value = 1
fork(), return value = 2
exec(), return value = 0
open(), return value = 3
close(), return value = 0
$write(), return value = 1
_write(), return value = 1
```

This is the output upon running the “make qemu” command in ubuntu.

This is basically printing –



The screenshot shows a terminal window titled "QEMU - Press Ctrl+Alt+G to release grab" with the title bar "Machine View". It displays the SeaBIOS boot process, starting with the message "SeaBIOS (version 1.15.0-1)". It then shows iPXE loading from IPXE.org, followed by kernel boot messages. The kernel logs include "Booting from Hard Disk...", "cpu0: starting 0", "sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58", and "init: starting sh".

```
SeaBIOS (version 1.15.0-1)

iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1FF8B590+1FECB590 CA00

Booting from Hard Disk...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
```

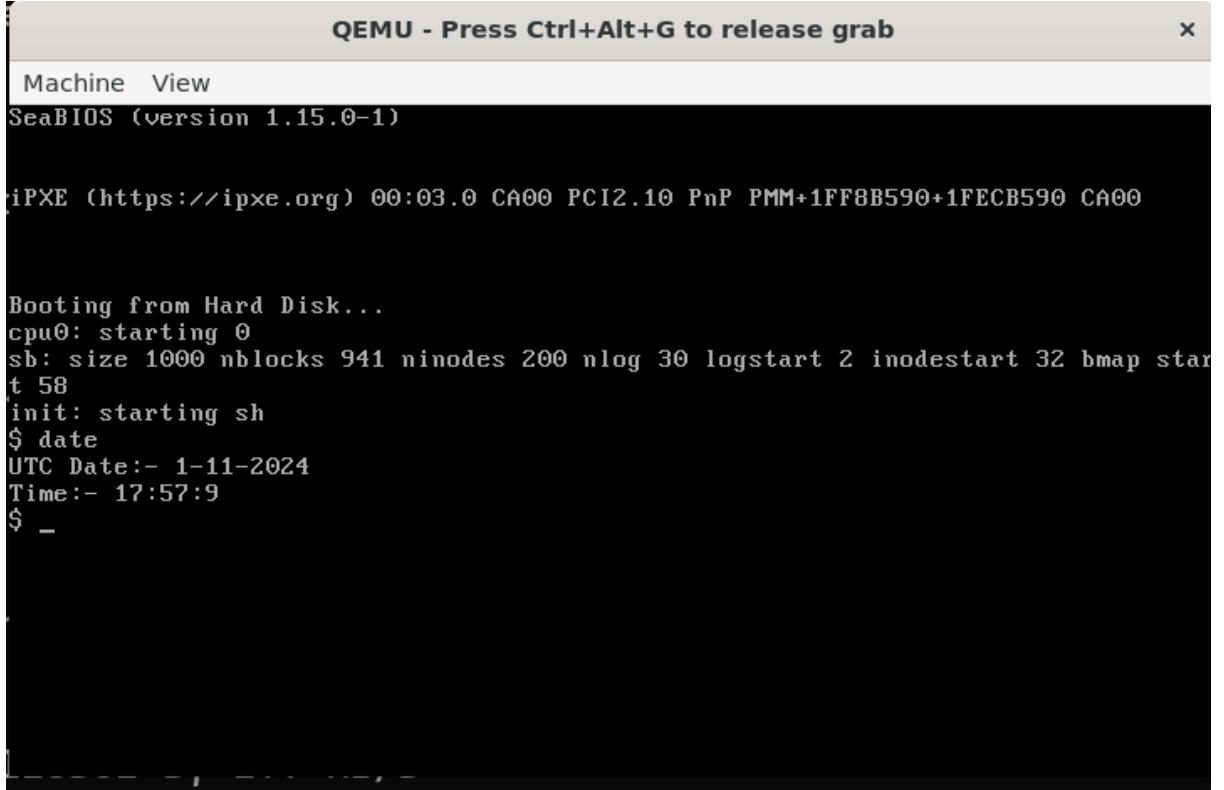
But with the required system call each time it is being called.

- **Code**

To only changes in code are in the syscall.c file. An additional array **syscall\_name[22]** of strings is made which maps the numbers of system calls to its actual names. And an extra line –  
**cprintf("%s(), return value = %d\n", syscall\_name[num], curproc->tf->eax);**  
So that it prints out the name and return value of each system call

## 2) Add a system call to output UTC date

### Ans- Output



The screenshot shows a terminal window titled "QEMU - Press Ctrl+Alt+G to release grab". The window displays the SeaBIOS boot process, including iPXE loading and device detection. It then shows the system booting from a hard disk and starting the init process. Finally, the user runs the "date" command, which outputs the UTC Date and Time.

```
QEMU - Press Ctrl+Alt+G to release grab
Machine View
SeaBIOS (version 1.15.0-1)

iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1FF8B590+1FECB590 CA00

Booting from Hard Disk...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ date
UTC Date:- 1-11-2024
Time:- 17:57:9
$ _
```

Here we can see that calling the “date” command gives out the date.

### - Code

The required files to be changed in this one are:

- syscall.h – defining the number for sys\_date
- syscall.c – adding sys\_date in sys\_calls[] array
- sysproc.c – adding the definition of sys\_date() along with  
extern void cmostime()
- Makefile – adding “\_date\” in UPROGS
- user.h – adding the existence of sys\_date function
- usys.S – adding “SYSCALL(date)”
- Creating a “date.c” file which outputs the date and time as mentioned in assignment pdf.