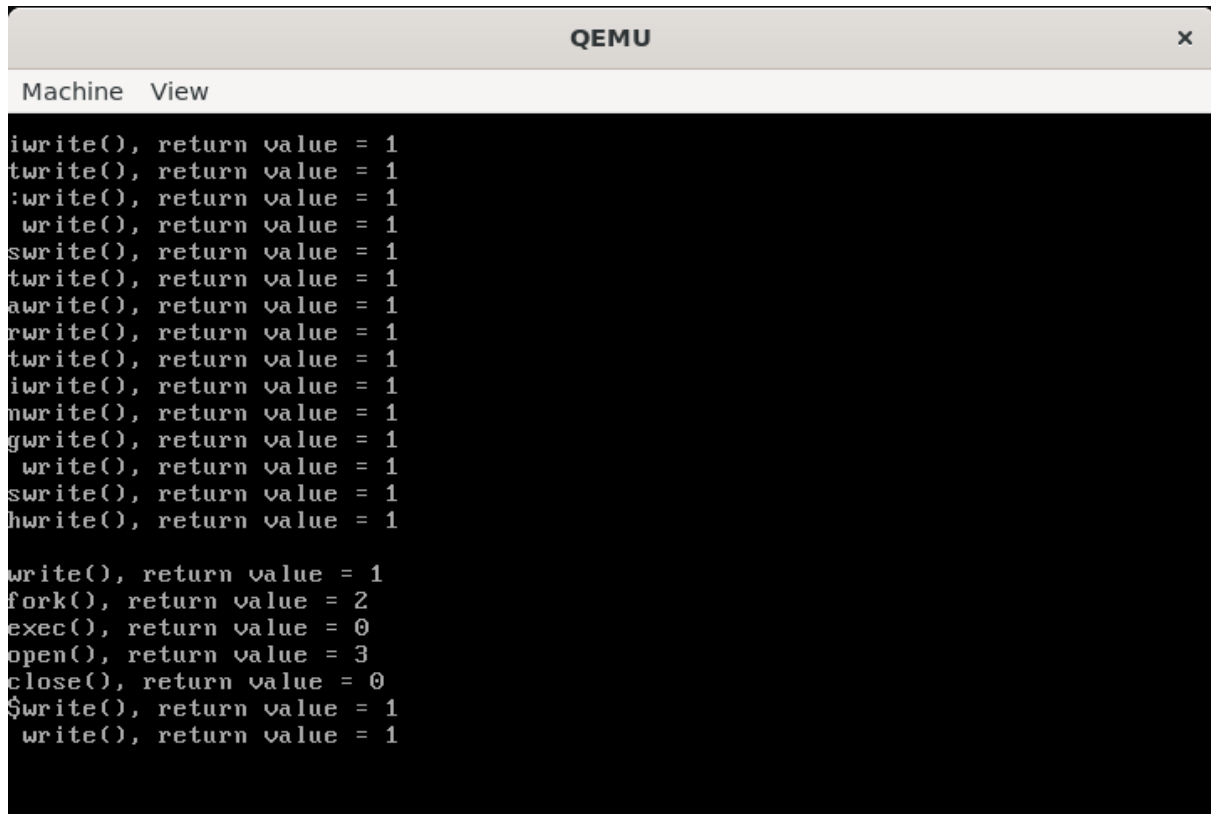


Assignment – 5

1) Print out a line for each system call invocation

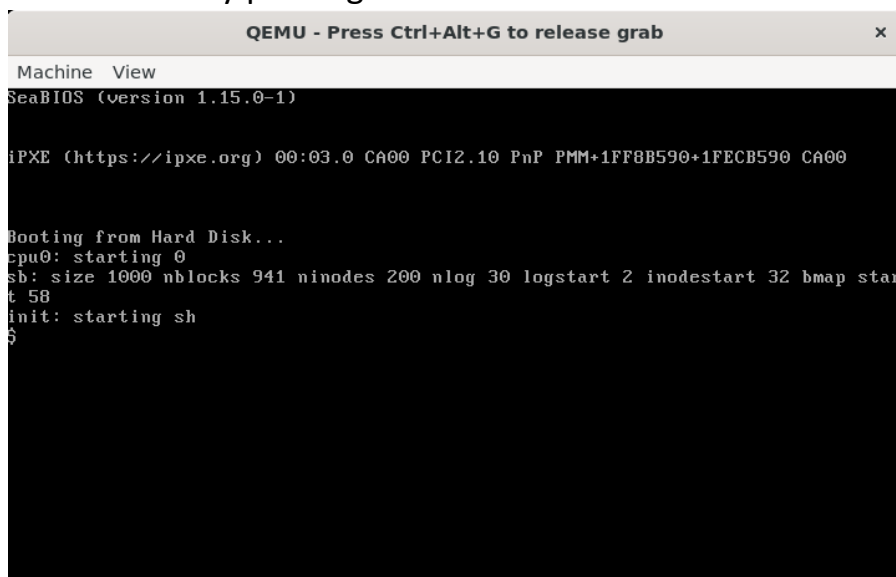
Ans- Output

A screenshot of a QEMU terminal window. The title bar says "QEMU" with a close button. Below the title bar is a menu bar with "Machine" and "View". The terminal output shows a list of system calls and their return values, such as "iwrite(), return value = 1", "twrite(), return value = 1", "write(), return value = 1", "swrite(), return value = 1", "fork(), return value = 2", "exec(), return value = 0", "open(), return value = 3", "close(), return value = 0", "\$write(), return value = 1", and "write(), return value = 1".

```
Machine View
iwrite(), return value = 1
twrite(), return value = 1
write(), return value = 1
swrite(), 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 –

A screenshot of a QEMU terminal window. The title bar says "QEMU - Press Ctrl+Alt+G to release grab" with a close button. Below the title bar is a menu bar with "Machine" and "View". The terminal output shows the boot process, including "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 star", "t 58", "init: starting sh", and "\$".

```
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 star
t 58
init: starting sh
$
```

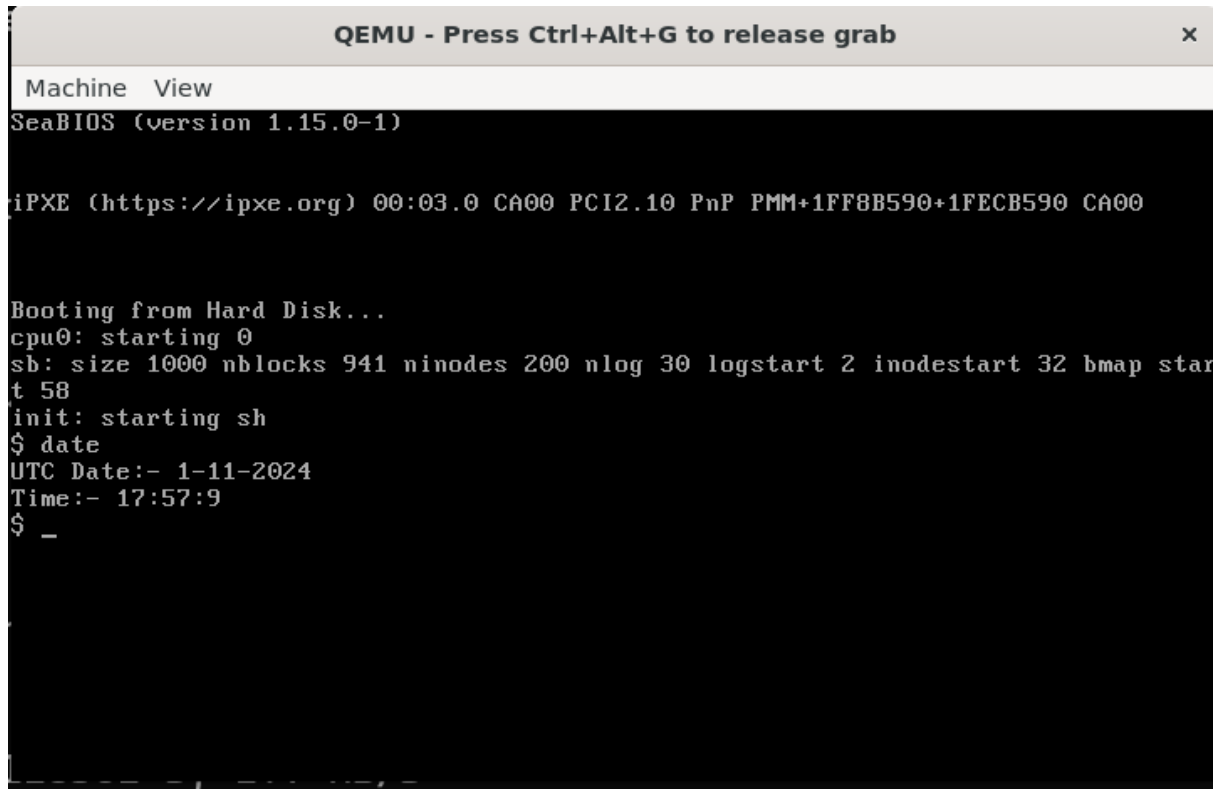
But with the required system call each time it is being called.

- **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



```
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 star
t 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.