Operating Systems, Lab Report 0B

Soham Roy, Mathematics and Computing
Roll No. 200123055

Exercise 1. Create a system call int sys_draw(void *buf, uint size), which copies an ASCII art image (Use a buffer of any picture, google it for more information) of a picture to a user-supplied buffer, provided that the buffer is large enough. You are welcome to use an ASCII art generator, or draw your own by hand. (One such example is given with the assignment) If the buffer is too small, or not valid, return a negative value. If the call succeeds, return the number of bytes copied. You may find it helpful to review how other system calls are implemented and compiled into the kernel, such as read.

An operating system supports two modes; the kernel mode and the user mode. When a program in user mode requires access to RAM or a hardware resource, it must ask the kernel to provide access to that particular resource. This is done via a system call. When a program makes a system call, the mode is switched from user mode to kernel mode. In order to define our own system, call in xv6, changes need to be made to 5 files. Namely, these files are as follows.

(i) syscall.h
(ii) syscall.c
(iii) sysproc.c
(iv) usys.S
(v) user.h

We would start the procedure by editing syscall.h in which a number is given to every system call. This file already contains 21 system calls. In order to add the custom system call, the following line needs to be added to this file.

#define SYS draw 22

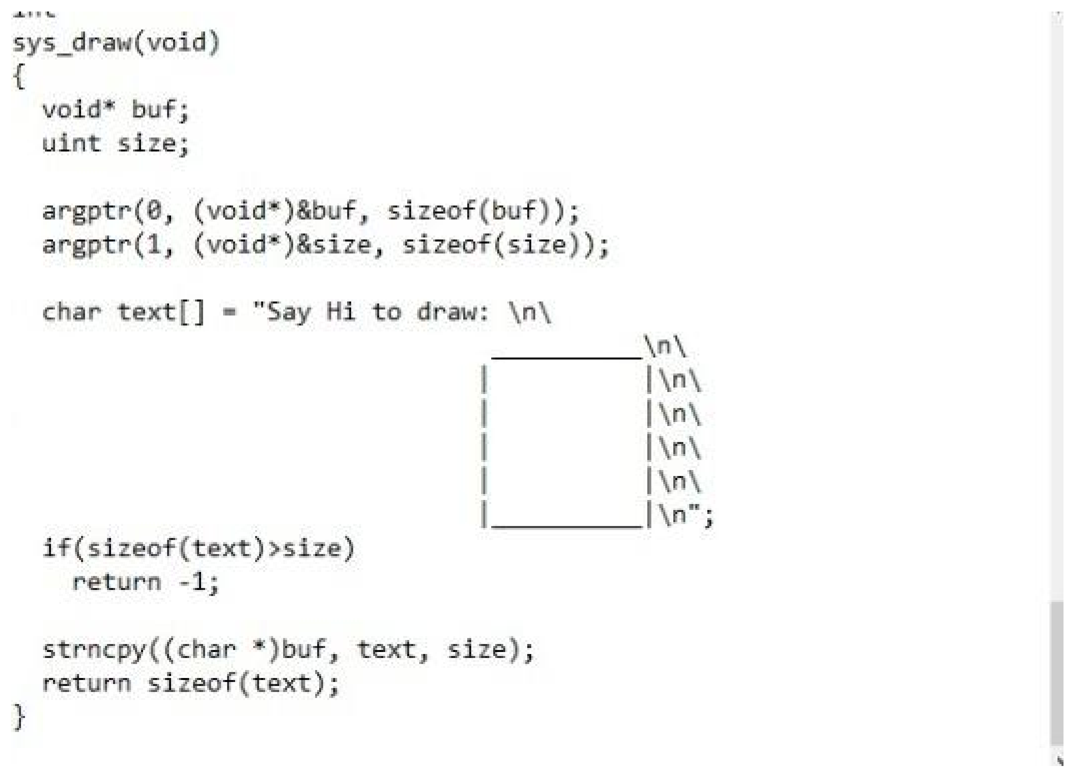
Next, we need to add a pointer to the system call in the syscall.c file. This file contains an array of function pointers which uses the above-defined numbers (indexes) as pointers to system calls which are defined in a different location. In order to add our custom system call, add the following line to this file.

[SYS_draw] sys_draw

When the system call with number 22 is called by a user program, the function pointer sys_wolfie which has the index SYS_wolfie or 22 will call the system call function. Hence, our next objective is to implement a system call function. But we do not implement the system call function in syscall.c. We only add a prototype as shown below.

Extern int sys_draw(void);

We implement the system call function in sysproc.c.



```
int
sys_draw(void)
{
  void* buf;
  uint size;

  argptr(0, (void*)&buf, sizeof(buf));
  argptr(1, (void*)&size, sizeof(size));

  char text[] = "Say Hi to draw: \n\
                               __________\n\
                              |          |\n\
                              |          |\n\
                              |          |\n\
                              |          |\n\
                              |__________|\n";
  if(sizeof(text)>size)
    return -1;

  strncpy((char *)buf, text, size);
  return sizeof(text);
}
```

Now, to add an interface for a user program to call the system call, we add

SYSCALL(draw)

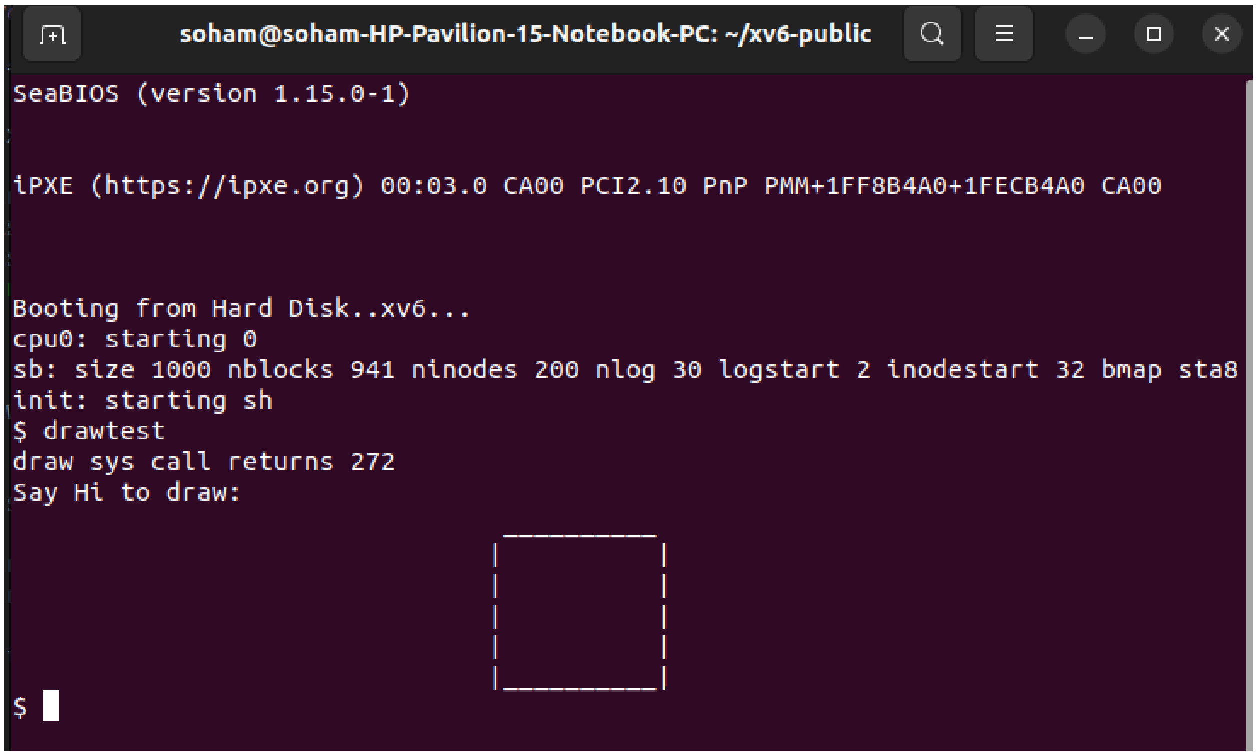
to usys.S and

Int draw(void*,uint);

to user.h

<u>Exercise 2</u>. Write a user-level application, called Drawtest.c, that gets the image from the kernel, and prints it to the console. When the OS runs, your program's binary should be included in fs.img and listed if someone runs ls at the xv6 shell's command prompt. Study Makefile to figure out how to compile a usermode program and add it to fs.img

Now the only task left is to add a user program to call the system call that we just made above. For this, I made a file drawtest.c inside xv6 folder and wrote the following code in it.

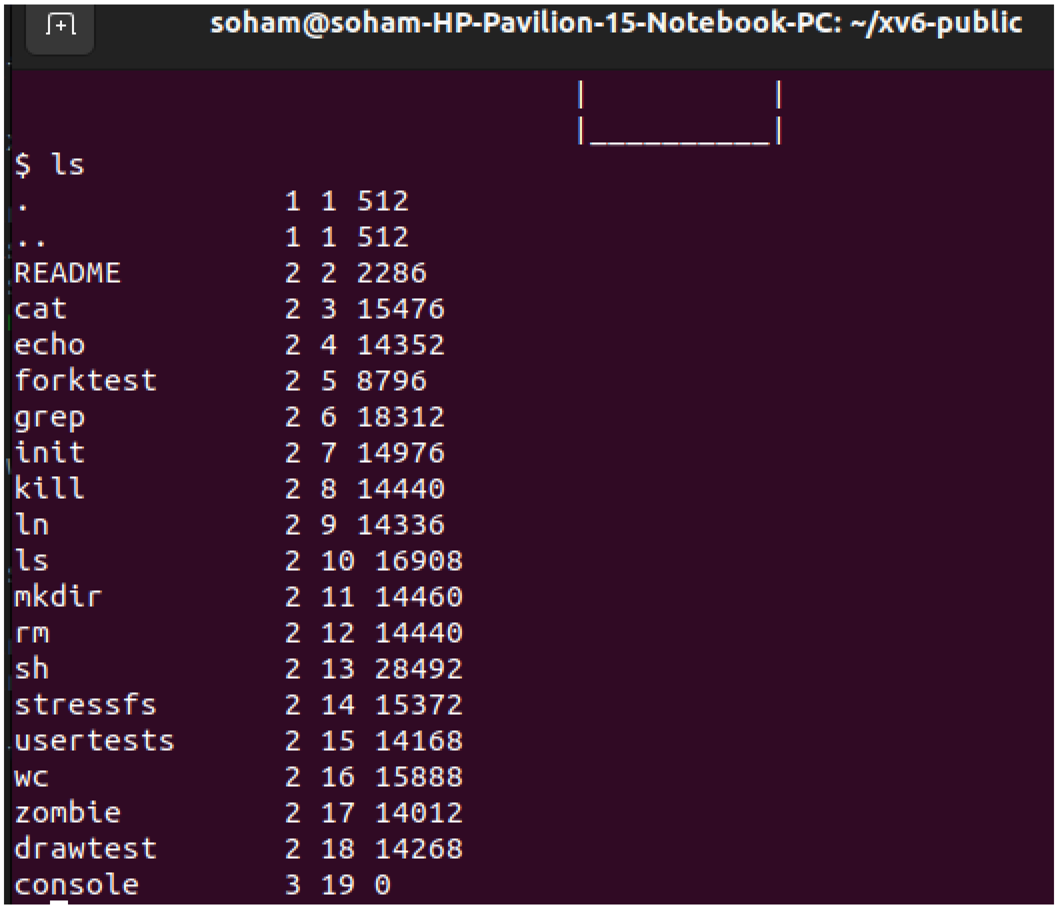


```
SeaBIOS (version 1.15.0-1)


iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1FF8B4A0+1FECB4A0 CA00



Booting from Hard Disk..xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap sta8
init: starting sh
$ drawtest
draw sys call returns 272
Say Hi to draw:
                                __________
                               |          |
                               |          |
                               |          |
                               |          |
                               |__________|
$ 
```



soham@soham-HP-Pavilion-15-Notebook-PC: ~/xv6-public

```
                                |          |
                                |__________|
$ ls
.              1 1 512
..             1 1 512
README         2 2 2286
cat            2 3 15476
echo           2 4 14352
forktest       2 5 8796
grep           2 6 18312
init           2 7 14976
kill           2 8 14440
ln             2 9 14336
ls             2 10 16908
mkdir          2 11 14460
rm             2 12 14440
sh             2 13 28492
stressfs       2 14 15372
usertests      2 15 14168
wc             2 16 15888
zombie         2 17 14012
drawtest       2 18 14268
console        3 19 0
```