CS 344 : Operating Systems Lab **Lab Assignment 1**

Name: Vignesh Ravichandra Rao **Roll Number**: 190101109

Commands to reproduce changes:
The patch file **`190101109_sysdraw.patch`** is included in the submission. This patch file works on the entire (unedited) XV6 source directory. Assuming that the XV6 source directory on your machine is **`xv6-public`**, run the following command from the parent directory:

```
patch -uN -d xv6-public < 190101109_sysdraw.patch
```

In case the name of the XV6 source code directory is different, replace **`xv6-public`** in the above command with the appropriate directory name.

Exercise 1)

The following changes were made in order to implement in the **`sys_draw`** system call:

Files created:

1. **`sysdraw.h`** :
   - Header file that contains the function prototype for the system call and the static character array containing the ASCII art.
   - Here the function prototype is: **`int sys_draw(void);`**
     - Kernel space system call functions can not directly accept arguments from the user space. Hence the function sysdraw does not accept any arguments.
     - Instead, the function fetches its argument using the argint and argptr functions which retrieves the arguments from the user program stack.
2. **`sysdraw.c`**:
   - This contains the implementation of the **`sys_draw`** system call.
     - First, the arguments are retrieved from the program stack and checked for their validity.
     - Then the size of user supplied buffer is tested to ensure that the ASCII art fits into it. In case it isn't, -1 is returned.
     - If the buffer is of sufficient size, the ASCII art defined in sysdraw.h is copied into the buffer using the memmove utility function which is predefined.
3. **`syscall.h`**:
   - All system call numbers are defined in this header file. The **`sys_draw`** system call was defined to have number 22 here.
   - This number defines the index of the system call in an array of function pointers in **`syscall.c`**
4. **`syscall.c`**:
   - Linked externally defined **`sys_draw`** function using extern int **`sys_draw(void)`** and added a pointer to this function at index = 22 in the array of system call function pointers
5. **`user.h`**:
   - Defined the function prototype **`int draw(void*,uint)`** for the user side system call function. This function will be used by the user to run the kernel's **`sys_draw`** function.
6. **`usys.S`**:
   - Links user space system call function draw to the kernel space **`sys_draw`** function.

- Added the line **`SYSCALL(draw)`**. Here **`SYSCALL`** is a macro that stores the index value of the system call defined in **`syscall.h`** in the **`%eax`** register. Using this index value, the corresponding system call is executed from **`syscall.c`**.

Exercise 2)

The following files were edited/created:

1. **`Makefile`** – edited
   - The Makefile was edited to include the .o object file from syscall.c into the list of object files required by the kernel.
   - The drawtest function was added to the list of user programs (UPROGS) so that it can be called from the shell console.
2. **`drawtest.c`** – created
   - The drawtest function was implemented in this file. This function makes use of the draw system call defined in user.h to print the ASCII art on the console. In case the user supplied buffer is too small, an error message is displayed.

For the changes of Exercise 2 to reflect on the image file of the OS, we run the following commands:

```
make clean
make qemu-nox
```

Then, we type drawtest in the XV6 console. The output is shown below