CS 344 Assignment 1 Report

Aditya Trivedi
190101005

1. Creating system call and modifying files

Modifications made :

Added draw function definition in user.h

```
int draw(void*, uint);
```

Added the sys_draw system call as an extern in syscall.c

```
extern int sys_draw(void);
```

Added the sys_draw system call in pointer array syscall.c

```
[SYS_draw]  sys_draw,
```

Added draw as a SYSCALL in usys.S

```
SYSCALL(draw);
```

Assigned SYS_draw macro as 22 in syscall.h

```
#define SYS_draw  22
```

Brief Explanation (edited files can be found in *modified_files* folder)

As our **user defined sys_draw function is inside sysdraw.c**, we must define it in user.h, also we add sys_draw as an extern to make it visible to all the other files in the program. We add it as the 22nd system call in the vector and also define a macro for it. Now we add the sysdraw function as a SYSCALL in usys.S.

Makefile changes are in the next section.

Creating File : sysdraw.c

```
#include "types.h"
#include "defs.h"

const char art[] = "_____________________________$$\n\
_____________________$$$$\n\
___________________$$$$$$\n\
___________________$$$$$$\n\
___________________$$$$\n\
___________________$$\n\
_________$$$$$$$$$$$$_$$$$$$$$$$$$\n\
_____$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
___$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
__$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
_$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
_$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
_$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
_$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
__$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
___$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
____$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
_____$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
______$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
________$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
__________$$$$$$$$$$$$$$$$$$$$$$$$$$\n\
____________$$$$$$$$$$$$$$$$$$$$$$$\n\
_____________$$$$$$$__$$$$$$$\n";

int sys_draw(void){
   int buffer_size;
   char *buffer;

   int sz = sizeof(art);

   // Checking whether the buffer is valid
   int invalidBuffer = argint(1, &buffer_size) < 0;

   // Checking whether the buffer size is sufficient to copy image
   int insufficientSize = buffer_size < sz;

   // Checking whether the pointer provided is correct
   int bufError = argptr(0, &buffer, buffer_size) < 0;

   // If any error, return -1
   if(invalidBuffer || insufficientSize || bufError){
       return -1;
   }

   // copying ASCII ART
   memmove(buffer, art, (uint)sz);

   // returning size
   return sz;
}
```

The art variable stores our art. sz stores the size of the ASCII ART, this will be used to check whether text can be accommodated in buffer. memmove is inbuilt function which is used to copy contents. Rest is explained as comments.

2. User level application drawtest

drawtest.c was created, it is responsible to get the art from the kernel, and prints to the console, it also prints a message in the case there arise any errors.
The first parameter “1” in printf describes that output must be to console.

```
#include "types.h"
#include "user.h"

char myBuffer[1024];


int main(void){
   int n = draw(myBuffer, 1024);
   if(n<0){
       printf(1,"Image Error in drawtest.c\n");
   }else if(write(1, myBuffer, n) != n){
       printf(1,"Write Error in drawtest.c\n");
   }
   exit();
}
```

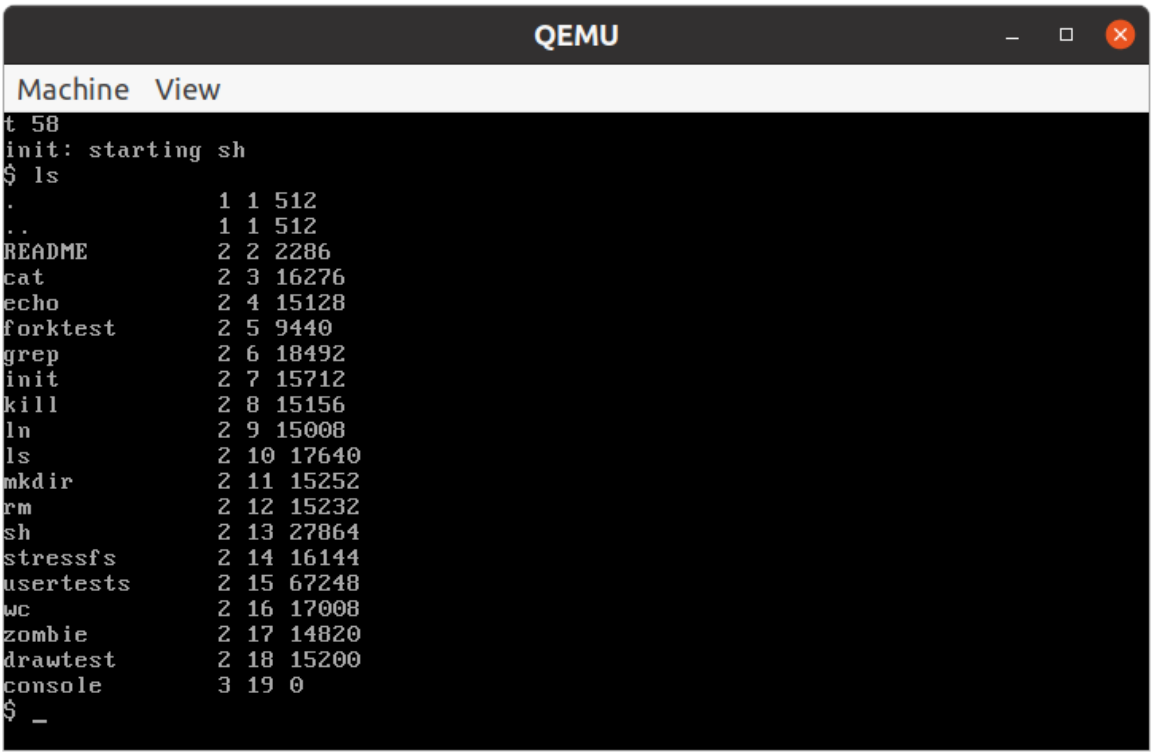
Here the buffer size is taken to be 1024, while the art size is lesser, so there are no image loading errors.
It will call the draw system call which is user-defined by us.

Modifications made to Makefile :

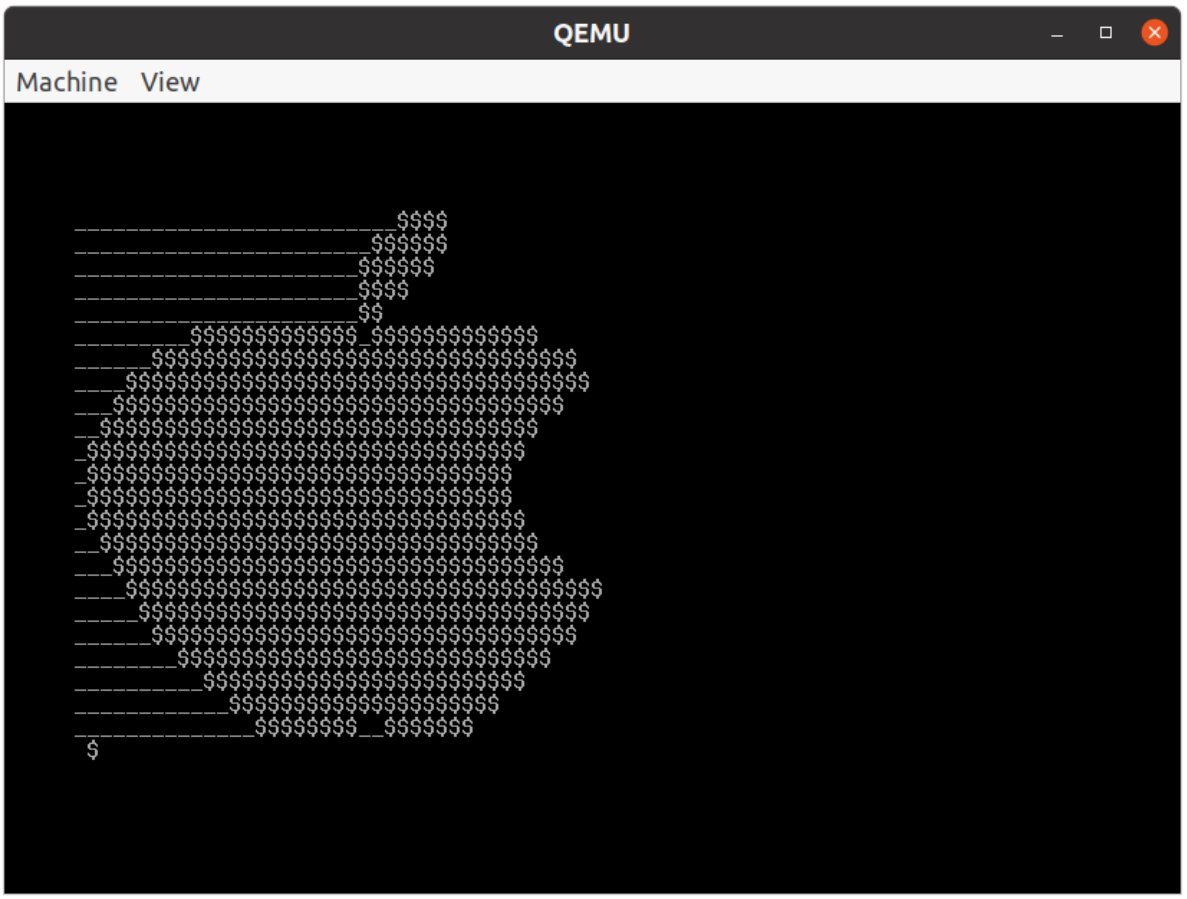
- _drawtest was added to user programs also known as **UPROGS** in Makefile
    - This tells the xv6 OS that it is a user defined program which must be compiled with other programs such as make director (mkdir), cat, echo, etc.
- sysdraw.o was added to the list of OBJS
    - This variable specifies all the object files required to construct the main program.

Successful Output :

After make qemu, running ls command, we get a list of all files, including the drawtest file



Upon running drawtest we get successful printing of art to console:



References

1. https://www.geeksforgeeks.org/xv6-operating-system-adding-a-new-system-call/
2. https://ampleux.wordpress.com/2018/02/22/how-to-add-a-user-program-to-xv6/
3. https://www.cs.usask.ca/staff/oster/makefiles.html
4. https://www.tutorialspoint.com/git/git_patch_operation.htm
5. https://www.geeksforgeeks.org/xv6-operating-system-add-a-user-program/

Note : All modified files are attached in a folder modified_files. This includes :

1. syscall.h
2. syscall.c
3. drawtest.c
4. Makefile
5. sysdraw.c
6. user.h
7. usys.S

Alternatively, copy the patch.txt file in xv6 directory and run following command

git apply patch.txt