CS_344 Assignment 0B

Akshat Mittal, 200101011

Aug 12, 2022

Task: Adding a System Call

Exercise 1:

To achieve this task, I have made **changes in 6 files** (all updated files are present in folder, along with **patchfile**).

1. sys_draw function in sysproc.c

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

  // problem in allocation of integer or buffer
  if(argint(1, (void*) &size) < 0)
    return -1;

  if(argptr(0, (void*) &buf, size) < 0)
    return -1;

  // ASCII art
  const char ascii_art[] = "\
        _                   _              _               _      _      _                 _         \n\
       / /\\                /\\_\\           / /\\            / /\\   / /\\   / /\\                /\\ \\      \n\
      / /  \\              / / /  _       / /  \\          / / /  / / /  / /  \\               \\_\\ \\    \n\
     / / /\\ \\            / / /  /\\_\\    / / /\\ \\__      / /_/  / / /  / / /\\ \\              /\\__ \\   \n\
    / / /\\ \\ \\          / / /__/ / /   / / /\\ \\___\\    / /\\ \\__/ / /  / / /\\ \\ \\            / /_ \\ \\  \n\
   / / /  \\ \\ \\        / /\\_____/ /    \\ \\ \\ \\/___/   / /\\ \\___\\/ /  / / /  \\ \\ \\          / / /\\ \\ \\ \n\
  / / /___/ /\\ \\      / /\\_______/      \\ \\ \\        / / /\\/___/ /  / / /___/ /\\ \\        / / /  \\/_/ \n\
 / / /_____/ /\\ \\    / / /\\ \\ \\     _    \\ \\ \\      / / /   / / /  / / /_____/ /\\ \\      / / /         \n\
/ /_________/\\ \\ \\  / / /  \\ \\ \\   /_/\\__/ / /     / / /   / / /  / /_________/\\ \\ \\    / / /          \n\
/ / /_       __\\ \\_\\/ / /    \\ \\ \\  \\ \\/___/ /     / / /   / / /  / / /_       __\\ \\_\\/_/ /          \n\
\\_\\___\\     /____/_/\\/_/      \\_\\_\\  \\_____\\/      \\/_/    \\/_/   \\_\\___\\     /____/_/\\_\\/           \n\
";

  // size of ascii art
  int art_size = sizeof(ascii_art);

  // if size of ascii art is more that available
  if(size < art_size)
    return -1;

  // copy art to buffer
  for(int i = 0; i < art_size; i ++)
    buf[i] = ascii_art[i];

  // return number of bytes copied, i.e., 1*(size of art)
  return art_size;
```

As these functions do not take an argument directly, we need to use ***argint*** and ***argptr*** function from ***syscall.c***. I returned **-1** if either of these are unsuccessful. After that, I calculated the size of the **ascii art** string. If it is more than the buffer size, **return -1**. Otherwise, transfer the art characters to the buffer character by character. At end, return size of ascii art, i.e., the number of bytes copied to buffer.

2. Including sys_draw in syscall.c

```c
extern int sys_write(void);
extern int sys_uptime(void);
extern int sys_draw(void);
```

First define the function prototype of ***sys_draw*** using ***extern*** keyword.

```c
[SYS_mkdir]   sys_mkdir,
[SYS_close]   sys_close,
[SYS_draw]    sys_draw,
};
```

Then, in the array of void function pointers, add this function. This enables ***syscall*** function to **track *sys_draw*** when system call is been made.

3. syscall.h

```
#define SYS_close  21
#define SYS_draw   22
```

Assign a system call number to **SYS_draw**, i.e., basically the position of this in the array of system calls.

4. Function definition in defs.h

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

// swtch.S
void            swtch(struct context**, struct context*);
```

Include the function definition, i.e.,

int draw(void*, uint);

5. user.h

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

Include user level function prototype in the **user.h** file, i.e., ***int draw(void*, uint);***

6. usys.S

```
30   SYSCALL(sleep)
31   SYSCALL(uptime)
32   SYSCALL(draw)
```

Declaring the **assembly** of draw function as a **pre-processor directive** in **usys.S**.

This all steps add the system call of **sys_draw** function.

Exercise 2:

In this part, we have to create a user-level application that gets the image from kernel and prints it to console. A new file **drawtest.c** was added which contains the code that has to be executed.

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

int main(int argc, char const *argv[])
{
    const int size = 2000;
    void *buf = malloc(size);
    int art_size = draw(buf,size);
    if(art_size == -1){
        printf(1, "Error: Could not allocate buffer!\n");
    }
    else{
        printf(1,"%s\n", (char*)buf);
    }
    exit();
}
```

- First, include the header files required.
- Then, declare the buffer and store the return value of **draw** system call.
- If some problem occurs, it will **return -1** and hence, we can print the error.
- Else, we can print our buffer which will contain the **ascii art**.

```
UPROGS=\
	_cat\
	_echo\
	_forktest\
	_grep\
	_init\
	_kill\
	_ln\
	_ls\
	_mkdir\
	_rm\
	_sh\
	_stressfs\
	_usertests\
	_wc\
	_zombie\
	_drawtest\
```

- Next the **Makefile** was updated to include **drawtest** in **UPROGS** so that is it available as a **command line instruction** in kernel.
- After this, the commands ***make clean***, ***make*** and ***make qemu-nox*** were run to start the qemu.
- The output of drawtest command can be seen below.

```
Booting from Hard Disk..xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ drawtest
```

```
$ ls
.              1 1 512
..             1 1 512
README         2 2 2286
cat            2 3 16280
echo           2 4 15132
forktest       2 5 9444
grep           2 6 18496
init           2 7 15716
kill           2 8 15160
ln             2 9 15012
ls             2 10 17644
mkdir          2 11 15256
rm             2 12 15236
sh             2 13 27876
stressfs       2 14 16148
usertests      2 15 67252
wc             2 16 17012
zombie         2 17 14824
drawtest       2 18 15264
console        3 19 0
```

- Also, when **ls** command was executed, drawtest was included in the list with other commands.

Note: I have attached all files changed with the report.
Next, I created a patch file using this command:

diff -ruN xv6-public xv6-public-assign_0B > patchfile

xv6-public: original directory.
xv6-public-assign_0B: directory with updated files.