CS 344: OPERATING SYSTEMS LABS
GROUP 12
ASSIGNMENT_0B-1

220123029 KAVURI VEDA VARSHA
220123030 KODIBOYINA LEELA VARSHINI
220123031 KOJJA VAMSI KRISHNA
220123052 RAMINENI HARDHIKA

EXERCISE 1:

As we have to create a system call we modified these files in xv6 public folder:

- syscall.h
- syscall.c
- sysproc.c
- user.h
- usys.S
- Makefile

Syscall.c

```
extern int sys_chdir(void);
extern int sys_close(void);
extern int sys_dup(void);
extern int sys_exec(void);
extern int sys_exit(void);
extern int sys_fork(void);
extern int sys_fstat(void);
extern int sys_getpid(void);
extern int sys_kill(void);
extern int sys_link(void);
extern int sys_mkdir(void);
extern int sys_mknod(void);
extern int sys_open(void);
extern int sys_pipe(void);
extern int sys_read(void);
extern int sys_sbrk(void);
extern int sys_sleep(void);
extern int sys_unlink(void);
extern int sys_wait(void);
extern int sys_write(void);
extern int sys_uptime(void);
extern int sys_draw(void);
```

```c
static int (*syscalls[])(void) = {
[SYS_fork]    sys_fork,
[SYS_exit]    sys_exit,
[SYS_wait]    sys_wait,
[SYS_pipe]    sys_pipe,
[SYS_read]    sys_read,
[SYS_kill]    sys_kill,
[SYS_exec]    sys_exec,
[SYS_fstat]   sys_fstat,
[SYS_chdir]   sys_chdir,
[SYS_dup]     sys_dup,
[SYS_getpid]  sys_getpid,
[SYS_sbrk]    sys_sbrk,
[SYS_sleep]   sys_sleep,
[SYS_uptime]  sys_uptime,
[SYS_open]    sys_open,
[SYS_write]   sys_write,
[SYS_mknod]   sys_mknod,
[SYS_unlink]  sys_unlink,
[SYS_link]    sys_link,
[SYS_mkdir]   sys_mkdir,
[SYS_close]   sys_close,
[SYS_draw]    sys_draw,
};

void
syscall(void)
{
```

We modified syscall.c by adding a line “extern int sys_draw(void);” and “[SYS_draw] sys_draw”

Syscall.h

```c
// System call numbers
#define SYS_fork    1
#define SYS_exit    2
#define SYS_wait    3
#define SYS_pipe    4
#define SYS_read    5
#define SYS_kill    6
#define SYS_exec    7
#define SYS_fstat   8
#define SYS_chdir   9
#define SYS_dup    10
#define SYS_getpid 11
#define SYS_sbrk   12
#define SYS_sleep  13
#define SYS_uptime 14
#define SYS_open   15
#define SYS_write  16
#define SYS_mknod  17
#define SYS_unlink 18
#define SYS_link   19
#define SYS_mkdir  20
#define SYS_close  21
#define SYS_draw   22
```

Here we modified syscall.h by adding “define SYS_draw 22”

User.h

```
Makefile   *syscall.c   syscall.h   user.h
int fork(void);
int exit(void) __attribute__((noreturn));
int wait(void);
int pipe(int*);
int write(int, const void*, int);
int read(int, void*, int);
int close(int);
int kill(int);
int exec(char*, char**);
int open(const char*, int);
int mknod(const char*, short, short);
int unlink(const char*);
int fstat(int fd, struct stat*);
int link(const char*, const char*);
int mkdir(const char*);
int chdir(const char*);
int dup(int);
int getpid(void);
char* sbrk(int);
int sleep(int);
int uptime(void);
int draw(void *buf, uint size);
// ulib.c
int stat(const char*, struct stat*);
char* strcpy(char*, const char*);
void *memmove(void*, const void*, int);
```

Here we modified user.h by adding “int draw(void *buf,uint size);”

usys.S

```
  movl $SYS_ ## name, %eax; \
  int $T_SYSCALL; \
  ret

SYSCALL(fork)
SYSCALL(exit)
SYSCALL(wait)
SYSCALL(pipe)
SYSCALL(read)
SYSCALL(write)
SYSCALL(close)
SYSCALL(kill)
SYSCALL(exec)
SYSCALL(open)
SYSCALL(mknod)
SYSCALL(unlink)
SYSCALL(fstat)
SYSCALL(link)
SYSCALL(mkdir)
SYSCALL(chdir)
SYSCALL(dup)
SYSCALL(getpid)
SYSCALL(sbrk)
SYSCALL(sleep)
SYSCALL(uptime)
SYSCALL(draw)
```

Here we modified usys.S by adding “SYSCALL(draw)”

Makefile

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

fs.img: mkfs README $(UPROGS)
        ./mkfs fs.img README $(UPROGS)

-include *.d

clean:
        rm -f *.tex *.dvi *.idx *.aux *.log *.*.ilg \
        *.o *.d *.asm *.sym vectors.S bootblocentryother \
        initcode initcode.out kernel xv6.img fs.img kernelmemfs \
```

Here we modified makefile by adding “_testdraw\”

Testdraw.c

```

#include "types.h"
#include "user.h"
#include "stat.h"

int main(int argc, char *argv[]) {
    char buf[256];
    int result;

    result = draw(buf, sizeof(buf));

    if (result < 0) {
        printf(1, "draw failed\n");
        exit();
    }

    printf(1, "draw succeeded, %d bytes copied:\n", result);
    printf(1, "%s\n", buf);

    exit();
}

```

Here we added a new file testdraw.c to xv6 folder

After adding the system call

```
guest@iitg-Vostro-3910: ~/Downloads/xv6/xv6-public-master
stdraw.o testdraw.c
ld -m    elf_i386 -N -e main -Ttext 0 -o _testdraw testdraw.o ulib.o usys.o prin
tf.o umalloc.o
objdump -S _testdraw > testdraw.asm
objdump -t _testdraw | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > testdraw.sym
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32
-Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -no-pie   -c -o zo
mbie.o zombie.c
ld -m    elf_i386 -N -e main -Ttext 0 -o _zombie zombie.o ulib.o usys.o printf.o
 umalloc.o
objdump -S _zombie > zombie.asm
objdump -t _zombie | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > zombie.sym
./mkfs fs.img README _cat _echo _forktest _grep _init _kill _ln _ls _mkdir _rm _
sh _stressfs _usertests _wc _testdraw _zombie
nmeta 59 (boot, super, log blocks 30 inode blocks 26, bitmap blocks 1) blocks 94
1 total 1000
balloc: first 674 blocks have been allocated
balloc: write bitmap block at sector 58
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=
raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap star
t 58
init: starting sh
$ testdraw
draw succeeded, 73 bytes copied:

  _____
 /     \
| () () |
 \  ^  /
  |||||
  |||||

$
```

After running the command testdraw it prints the ascii_art

EXERCISE 2:

To print the ascii_art image in the console we have added the file drawtest.c and also modified makefile

Drawtest.c

user.h usys.S testdraw.c drawtest.c

```c
// user/Drawtest.c
#include "types.h"
#include "user.h"

#define BUFFER_SIZE 256

int main(int argc, char *argv[]) {
    char buf[BUFFER_SIZE];
    int result;

    // Call the sys_draw system call
    result = draw(buf, sizeof(buf));

    if (result < 0) {
        printf(2, "draw failed\n");
        exit();
    }

    printf(1, "draw succeeded, %d bytes copied:\n", result);
    printf(1, "%s\n", buf);

    exit();
}
```

Here we added a new file drawtest.c to xv6 folder

Makefile

```
Makefile   *syscall.c   syscall.h   user.h
UPROGS=\
        _cat\
        _echo\
        _forktest\
        _grep\
        _init\
        _kill\
        _ln\
        _ls\
        _mkdir\
        _rm\
        _sh\
        _stressfs\
        _usertests\
        _wc\
        _testdraw\
        _drawtest\
        _zombie\

fs.img: mkfs README $(UPROGS)
        ./mkfs fs.img README $(UPROGS)

-include *.d

clean:
        rm -f *.tex *.dvi *.idx *.aux *.log *.ind *.ilg \
        *.o *.d *.asm *.sym vectors.S bootblock entryother \
        initcode initcode.out kernel xv6.img
```

Here we modified makefile by adding “_drawtest\”

```
objdump -S _wc > wc.asm
objdump -t _wc | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > wc.sym
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -no-pie   -c -o testdraw.o testdraw.c
ld -m    elf_i386 -N -e main -Ttext 0 -o _testdraw testdraw.o ulib.o usys.o printf.o umalloc.o
objdump -S _testdraw > testdraw.asm
objdump -t _testdraw | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > testdraw.sym
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -no-pie   -c -o drawtest.o drawtest.c
ld -m    elf_i386 -N -e main -Ttext 0 -o _drawtest drawtest.o ulib.o usys.o printf.o umalloc.o
objdump -S _drawtest > drawtest.asm
objdump -t _drawtest | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > drawtest.sym
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -no-pie   -c -o zombie.o zombie.c
ld -m    elf_i386 -N -e main -Ttext 0 -o _zombie zombie.o ulib.o usys.o printf.o umalloc.o
objdump -S _zombie > zombie.asm
objdump -t _zombie | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > zombie.sym
./mkfs fs.img README _cat _echo _forktest _grep _init _kill _ln _ls _mkdir _rm _sh _stressfs _usertests _wc _testdraw _drawtest _zombie
nmeta 59 (boot, super, log blocks 30 inode blocks 26, bitmap blocks 1) blocks 941 total 1000
balloc: first 704 blocks have been allocated
balloc: write bitmap block at sector 58
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ ls
.              1 1 512
..             1 1 512
README         2 2 2286
cat            2 3 15516
echo           2 4 14396
forktest       2 5 8844
grep           2 6 18360
init           2 7 15016
kill           2 8 14480
ln             2 9 14376
ls             2 10 16948
mkdir          2 11 14504
rm             2 12 14484
sh             2 13 28544
stressfs       2 14 15412
usertests      2 15 62916
wc             2 16 15940
testdraw       2 17 14592
drawtest       2 18 14584
zombie         2 19 14060
console        3 20 0
$ ./drawtest
draw succeeded, 73 bytes copied:
   ____
  /    \
 | ()  () |
  \   ^   /
   |||||
   |||||

$
```

After adding the file drawtest.c we can print the ascii_art in the console