ASSIGNMENT – 0

Name: Siddam Shetty Sahithi Shresta

Roll No: 200123054

Assignment0A -

1.

```c
#include <stdio.h>
int main(int argc, char **argv)
{
        int x = 1;
        printf("Hello x = %d\n", x);
        __asm__ ( "incl %0" : "+r"(x) );
        printf("Hello x = %d after increment\n", x);
        if(x == 2){
                printf("OK\n");
        }
        else{
                printf("ERROR\n");
        }
}
```

```
sahithishresta@sahithishresta:~/Desktop/os$ gcc ex1.c
sahithishresta@sahithishresta:~/Desktop/os$ ./a.out
Hello x = 1
Hello x = 2 after increment
OK
sahithishresta@sahithishresta:~/Desktop/os$
```

The value of x is incremented by 1 using an inline assembly code.

2. “si” instruction in gdb is used to execute a single machine instruction.

```
(gdb) source .gdbinit
Redefine command "hook-stop"? (y or n) [answered Y; input not from terminal]
+ target remote localhost:25501
warning: No executable has been specified and target does not support
determining executable automatically.  Try using the "file" command.
The target architecture is set to "i8086".
[f000:fff0]    0xffff0: ljmp   $0x3630,$0xf000e05b
0x0000fff0 in ?? ()
+ symbol-file kernel
(gdb) si
[f000:e05b]    0xfe05b: cmpw   $0xffc8,%cs:(%esi)
0x0000e05b in ?? ()
(gdb) si
[f000:e062]    0xfe062: jne    0xd241d0f1
0x0000e062 in ?? ()
(gdb) si
[f000:e066]    0xfe066: xor    %edx,%edx
0x0000e066 in ?? ()
(gdb) si
[f000:e068]    0xfe068: mov    %edx,%ss
0x0000e068 in ?? ()
(gdb) si
[f000:e06a]    0xfe06a: mov    $0x7000,%sp
0x0000e06a in ?? ()
(gdb) si
[f000:e070]    0xfe070: mov    $0xfc63,%dx
0x0000e070 in ?? ()
(gdb) si
[f000:e076]    0xfe076: jmp    0x5576cf6e
0x0000e076 in ?? ()
(gdb) si
[f000:cf6c]    0xfcf6c: cli
0x0000cf6c in ?? ()
(gdb) si
[f000:cf6d]    0xfcf6d: cld
0x0000cf6d in ?? ()
(gdb) si
[f000:cf6e]    0xfcf6e: mov    %ax,%cx
0x0000cf6e in ?? ()
(gdb)
```

4.

The output obtained:

```
1: a = 0x7ffcfd0b2cc0, b = 0x55a600d566b0, c = 0x7ffcfd0b2ce7
2: a[0] = 200, a[1] = 101, a[2] = 102, a[3] = 103
3: a[0] = 200, a[1] = 300, a[2] = 301, a[3] = 302
4: a[0] = 200, a[1] = 400, a[2] = 301, a[3] = 302
5: a[0] = 200, a[1] = 128144, a[2] = 256, a[3] = 302
6: a = 0x7ffcfd0b2cc0, b = 0x7ffcfd0b2cc4, c = 0x7ffcfd0b2cc1
```

```
sahithishresta@sahithishresta:~/xv6-public$ objdump -h kernel

kernel:     file format elf32-i386

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00007238  80100000  00100000  00001000  2**4
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .rodata       0000101f  80107240  00107240  00008240  2**5
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  2 .data         00002516  80109000  00109000  0000a000  2**12
                  CONTENTS, ALLOC, LOAD, DATA
  3 .bss          0000afb0  8010b520  0010b520  0000c516  2**5
                  ALLOC
  4 .debug_line   00006af7  00000000  00000000  0000c516  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  5 .debug_info   00010edc  00000000  00000000  0001300d  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  6 .debug_abbrev 000044b8  00000000  00000000  00023ee9  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  7 .debug_aranges 000003b0  00000000  00000000  000283a8  2**3
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  8 .debug_str    00000e02  00000000  00000000  00028758  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  9 .debug_loclists 000050b1  00000000  00000000  0002955a  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
 10 .debug_rnglists 00000845  00000000  00000000  0002e60b  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
 11 .debug_line_str 0000013a  00000000  00000000  0002ee50  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
 12 .comment      00000026  00000000  00000000  0002ef8a  2**0
                  CONTENTS, READONLY
```

```
sahithishresta@sahithishresta:~/xv6-public$ objdump -h bootblock.o

bootblock.o:     file format elf32-i386

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         000001c3  00007c00  00007c00  00000074  2**2
                  CONTENTS, ALLOC, LOAD, CODE
  1 .eh_frame     000000b0  00007dc4  00007dc4  00000238  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  2 .comment      00000026  00000000  00000000  000002e8  2**0
                  CONTENTS, READONLY
  3 .debug_aranges 00000040  00000000  00000000  00000310  2**3
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  4 .debug_info   00000585  00000000  00000000  00000350  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  5 .debug_abbrev 0000023c  00000000  00000000  000008d5  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  6 .debug_line   00000283  00000000  00000000  00000b11  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  7 .debug_str    0000020e  00000000  00000000  00000d94  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  8 .debug_line_str 00000049  00000000  00000000  00000fa2  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
  9 .debug_loclists 0000018d  00000000  00000000  00000feb  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
 10 .debug_rnglists 00000033  00000000  00000000  00001178  2**0
                  CONTENTS, READONLY, DEBUGGING, OCTETS
```

The values of VMA and LMA are same when we run objdump -h bootblock.o which indicates that it does loading and executing from same address.

Assignment-0B:

1 and 2. To define a system call in xv6, we need to make changes in the following files:

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

i. syscall.h: As the file already contains 21 system calls, so as to create a new system call, we need to add the following line –

    #define SYS_draw 22

ii. syscall.c: To add pointer to the system call in this file, we need to add the following line –

    [SYS_draw] sys_draw,

    extern int sys draw(void);

iii. sysproc.c: Add the following code to this –

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

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

  char text[] = ".d88b.  d8888b. d88888b d8888b.  .d8b.  d888888b d888888b d8b   db  d888b       .d8888. db    db .d8888. d888888b d88888b .88b  d88. \n
                .8P  Y8. 88   8D 88      88   8D d8   8b  ~~88~~     88    888o  88 88  Y8b      88   YP  8b  d8  88   YP  ~~88~~  88      88 YbdP 88\n
                88    88 88oodD  88ooooo 88oobY  88ooo88    88       88    88V8o 88 88           8bo.     8bd8    8bo.       88     88ooooo 88  88  88\n
                88    88 88~~~   88~~~~~ 88`8b   88~~~88    88       88    88 V8o88 88  ooo        Y8b.    88       Y8b.     88     88~~~~~ 88  88  88\n
                8b  d8   88      88.     88 `88. 88   88    88      .88.   88  V888 88. ~8~      db   8D    88     db   8D    88     88.     88  88  88\n
                 Y88P    88      Y88888P 88   YD YP   YP    YP    Y888888P VP   V8P  Y888P       `8888Y'    YP     `8888Y'    YP     Y88888P YP  YP  YP\n";
  if(sizeof(text)>size)
    return -1;


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

iv. To add interface, add this to usys.S –

SYSCALL(draw)

v. To user.h

int draw(void*,uint);

2.

Then run “make clean”, “make”, “make qemu-nox” and then type Drawtest, then the text will be displayed as shown below.