**kernel.c (Source code)**

/\* Kernel.c

\* Bina Maria (00366107)

\* MS (CS) Fall - 17

\* Kernel program to build an Operating System from scratch.

\*/

/\* Declaring Function \*/

void printString(char \* s);

void readString(char \* s);

void readSector(char \* buffer, int sector);

void handleInterrupt21(int ax, int bx, int cx, int dx);

int mod(int a, int b);

int div(int a, int b);

/\* The entry point of the program \*/

int main(void)

{

char line[80];

char buffer[512];

printString("Hello World\n\r\0");

makeInterrupt21();

/\*interrupt(0x21,0,0,0,0);\*/

/\*printString("Enter a line: \0");\*/

/\*readString(line);\*/

/\*printString(line);\*/

interrupt(0x21,0,"Enter a line: ",0,0);

interrupt(0x21,1,line,0,0);

interrupt(0x21,0,line,0,0);

interrupt(0x21,2,buffer,30,0);

interrupt(0x21,0,buffer,0,0);

/\*readSector(buffer, 30);

printString(buffer);\*/

while(1);

return 0;

}

void printString(char \* s)

{

char al;

char ah = 0xe;

int ax;

int i;

for(i = 0; s[i] != 0; i++)

{

al = s[i];

ax = ah \* 256 + al;

interrupt(0x10, ax, 0, 0, 0);

}

}

void readString(char \* s)

{

char a;

int i = 0;

char ah = 0xe;

while(i < 77)

{

a = interrupt(0x16, 0, 0, 0, 0);

/\* if enter pressed\*/

if(a == 0xd)

{

break;

}

/\* if backspace pressed\*/

else if(a == 0x8)

{

if(i > 0)

{/\* To print the backspace, and then a whitespace and then again print the backspace to remove the character \*/

interrupt(0x10, ah \* 256 + a, 0, 0, 0);

interrupt(0x10, ah \* 256 + ' ', 0, 0, 0);

interrupt(0x10, ah \* 256 + a, 0, 0, 0);

i--;

}

}

/\* if normal character \*/

else

{

s[i] = a;

interrupt(0x10, ah \* 256 + a, 0, 0, 0);

i++;

}

}

interrupt(0x10, ah \* 256 + 0xa, 0, 0, 0);

interrupt(0x10, ah \* 256 + '\r', 0, 0, 0);

s[i] = 0xa;

s[i+1] = '\r';

s[i+2] = 0x0;

}

void readSector(char \* buffer, int sector)

{

int ah = 2;

int al = 1;

int ch = div(sector, 36);

int cl = mod(sector, 18) + 1;

int dh = mod(div(sector, 18), 2);

int dl = 0;

int ax = ah \* 256 + al;

int cx = ch \* 256 + cl;

int dx = dh \* 256 + dl;

interrupt(0x13, ax, buffer, cx, dx);

}

void handleInterrupt21(int ax, int bx, int cx, int dx)

{

if(ax == 0)

printString(bx);

else if(ax == 1)

readString(bx);

else if(ax == 2)

readSector(bx, cx);

else if(ax >= 3)

printString("Error: handleInterrupt21!!\0");

}

int mod(int a, int b)

{

while(a >= b)

a -= b;

return a;

}

int div(int a, int b)

{

int q = 0;

while((q + 1) \* b <= a)

q += 1;

return q;

}

**compileOS.sh (Source code)**

#compileOS.sh

#Bina Maria (00366107)

#MS (CS) Fall - 17

#Shell script to produce floppya.img file.

nasm bootload.asm #To assemble bootloader using NASM assembler!

dd if=/dev/zero of=floppya.img bs=512 count=2880 #To make image file of floppy disk that is filled with zeros!

dd if=bootload of=floppya.img bs=512 count=1 conv=notrunc #To copy bootload to the beginning of floppya.img!

bcc -ansi -c -o kernel\_c.o kernel.c #To compile your kernel!

as86 kernel.asm -o kernel\_asm.o #To assemble kernel.asm!

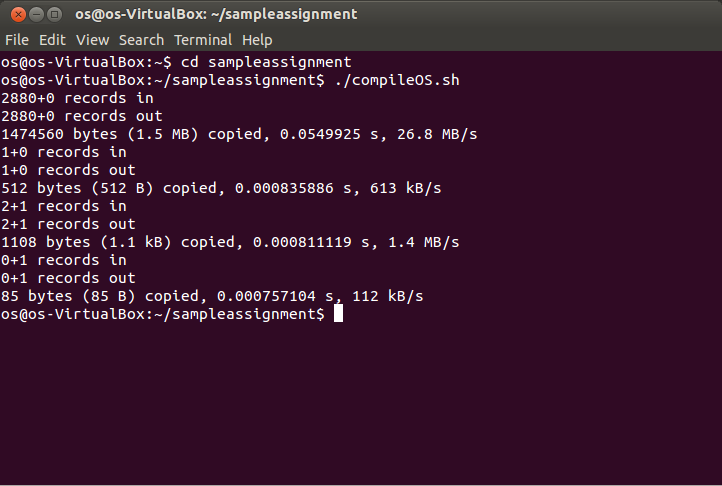
ld86 -o kernel -d kernel\_c.o kernel\_asm.o #To link and produce kernel!

dd if=kernel of=floppya.img bs=512 conv=notrunc seek=3 #To copy kernel to the third sector of floppya.img!

dd if=message.txt of=floppya.img bs=512 count=1 seek=30 conv=notrunc #To put message.txt at sector 30

**Output Screenshots**

When tried running shell script – compileOS.sh.



2. Output when made interrupt 0x21 calls to the functions - printString, readString and readSector and tried running Emumaker86 (Simulator).

