#include <system.h>

//Main

void main()

{

gdt\_install(); //Setup processor driver

idt\_install(); //Setup Interrupt and Exception access

isr\_install(); //Setup exception handler

irq\_install(); //Handle device interrupts/events

timer\_install(); //Setup timer

keyboard\_install(); //Setup keyboard

//video\_install();

asm volatile("sti");

cls();

PrintLn("Hello World");

for (;;);

}

//////////////

//Memory I/O//

//////////////

//Copies 'count' bytes of data from 'src' to 'dest'

unsigned char \*memcpy(unsigned char \*dest, const unsigned char \*src, int count)

{

const char \*sp = (const char \*)src;

char \*dp = (char \*)dest;

for(; count != 0; count--) \*dp++ = \*sp++;

return dest;

}

unsigned short \*memcpyw(unsigned short \*dest, const unsigned short \*src, int count)

{

const short \*sp = (const short \*)src;

short \*dp = (short \*)dest;

for(; count != 0; count--) \*dp++ = \*sp++;

return dest;

}

//Sets 'count' bytes in 'dest' to 'val'

//memset(String dest, Char val, int count)

unsigned char \*memset(unsigned char \*dest, unsigned char val, int count)

{

char \*temp = (char \*)dest;

for(; count != 0; count--) {\*temp++ = val;}

return dest;

}

//Same as memset just with 16 bits

unsigned short \*memsetw(unsigned short \*dest, unsigned short val, int count)

{

unsigned short \*temp = (unsigned short \*)dest;

for(; count != 0; count--) {\*temp++ = val;}

return dest;

}

unsigned int \*memseti(unsigned int \*dest, unsigned int val, int count)

{

unsigned int \*temp = (unsigned int \*)dest;

for(; count !=0; count--) {\*temp++ = val;}

return dest;

}

///////////

//Strings//

///////////

//Returns the length of the string

int strlen(const char \*str)

{

int i;

for(i = 0; \*str != '\0'; str++) {i++;}

return i;

}

////////////

//Port I/O//

////////////

//Get char from port

unsigned char inportb(unsigned short \_port)

{unsigned char rv; asm volatile("inb %1, %0" : "=a" (rv) : "dN" (\_port)); return rv;}

//Send char to port

void outportb(unsigned short \_port, unsigned char \_data)

{asm volatile("outb %1, %0" : : "dN" (\_port), "a" (\_data));}