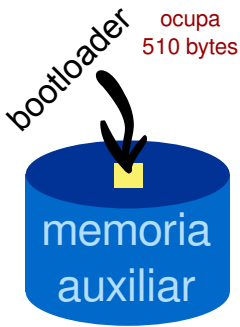
Resumen

- Cuando una computadora arranca solo existe el BIOS (Basic Input/Output system).
- El proceso de booteo comienza ejecutando el código del BIOS, ubicado en la posición `0xFFFF0`, en modo real.
- El BIOS tiene código en ROM que realiza la inicialización (por ejemplo, la placa de video) y una verificación inicial de la máquina: POST (Power on self-test).
- Luego busca algún dispositivo de booteo: Disco Rígido, Floppy, USB, etc...
- Una vez localizado el dispositivo de arranque, carga el primer sector de 512 bytes (CDROM 2048 bytes) en la posición de memoria `0x07C00` y salta a esa dirección.
- Ahora la imagen de arranque es la encargada de cargar el kernel y luego pasarle el control.
- Para que una imagen sea de arranque debe ocupar exactamente 512 bytes (excepto en el CDROM), y estar firmada en los últimos dos bytes con `0x55AA`.
- Una imagen de linux de ejemplo:
  `http://bochs.sourceforge.net/diskimages.html`

Bootloader - Pasos para indicar



¡Todo esto en 510 bytes!

1- Determinar el 'disco' y la partición a bootear

2- Determinar donde esta la imagen del kernel en ese 'disco'

3- Cargar la imagen del kernel en memoria

4- Correr el "kernel"

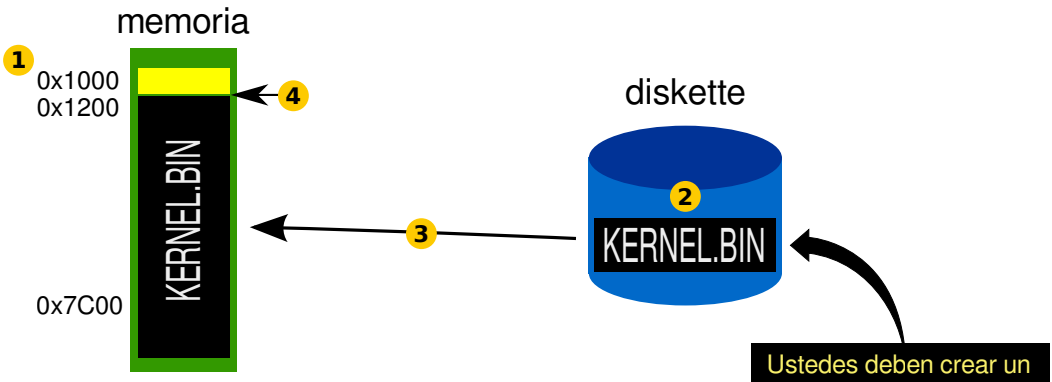
4.1- Pasar a modo protegido

4.2- Preparar las estructuras para administrar la memoria

4.3- Preparar las estructuras del sistema

4.4- ¡Listo!

EL Bootloader de Orga2



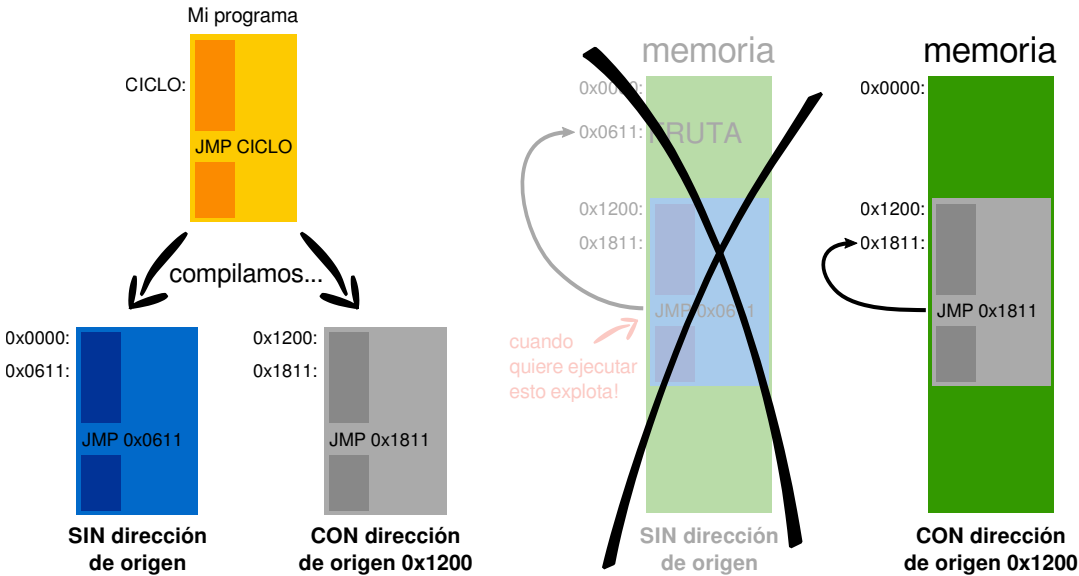
1 - Se copia el Bootloader en la posición **0x1000** de la memoria
2 - Se busca el archivo **KERNEL.BIN** en el diskette
3 - Se copia ese archivo en la posición **0x1200** de la memoria
4 - Se salta y se ejecuta la instrucción en la posición **0x1200** de la memoria

Compilando y Enlazando

- Un compilador construye un programa de forma que pueda correr sobre un **sistema operativo** determinado
- Para resolver direcciones, toma una **dirección de inicio**, por ejemplo `0x00000000`
- Cada etiqueta se traduce a una dirección contando **bytes** desde la dirección de inicio
- Los **destinos** de saltos o llamadas a funciones se debe conocer en tiempo de enlazado
- ¿Y si no estamos en un sistema operativo?

  Ej: el archivo `KERNEL.BIN` se carga en la dirección `0x1200`

Compilando y Enlazando



Compilando y Enlazando

Indicar la dirección de origen

Hay 2 formas de hacerlo, según como se compile:

- De **ensamblador a binario**, usamos la directiva `ORG` al inicio del `archivo.asm` para indicar la direccion de origen

```
ORG 0x1200
```

- De **ensamblador a elf**, usamos el parámetro `-Ttext` en el linker

```
-Ttext 0x1200
```

Compilando y Enlazando

Compilación de ensamblador a binario

- Ensamblado:

```
nasm -fbin archivo.asm -o archivo.bin
```

- Consideraciones:
  - Todo el código ejecutable tiene que estar incluido (No hay bibliotecas)
  - Se ejecuta tal cual se escribió, no hay entry point.

Compilando y Enlazando

Compilación de C en formato elf32 y linkeo

- Compilación:

```
gcc -m32 -fno-zero-initialized-in-bss -fno-stack-protector
-ffreestanding -c -o archivo.elf archivo.c
```

- Linkeo:

```
ld -static -m elf i386 -nostdlib -N -b elf32-i386 -e start
-Ttext 0x1200 -o archivo.elf archivo.o
```

- Lo convertimos en binario:

```
objcopy -S -O binary archivo.elf archivo.bin
```

Compilando y Enlazando

Compilación de assembly en formato elf32 y linkeo

- Compilación:

```
nasm -felf32 archivo.asm -o archivo.o
```

- Linkeo:

```
ld -static -m elf i386 --oformat binary -b elf32-i386 -e start
-Ttext 0x1200 archivo.o -o archivo.bin
```

- Consideraciones:
  - OJO código de 32 bits (en modo protegido)
  - Se pueden usar biblotecas. No se respeta el entry point.
  - El parámetro Ttext da el orígen de la sección .text.
  - Si usan el Bootloader de Orga 2, deben usar 0x1200 como origen de la sección .text.

Compilando y Enlazando

Compilación de un bootloader y creacion de diskette

- Creamos un diskette vacio:

```
dd bs=512 count=2880 if=/dev/zero of=diskette.img
```

- Formateamos la imagen en FAT12:

```
sudo mkfs.msdos -F 12 diskette.img -n ETIQUETA
```

- Escribimos en el sector de booteo:

```
dd if=bootloader.bin of=diskette.img count=1 seek=0 conv=notrunc
```

- Copiado del KERNEL.BIN dentro del diskette

```
mcopy -i diskette.img kernel.bin ::/
```

¡Gracias!

Recuerden leer los comentarios al final de
este video por aclaraciones o fe de erratas.