Laboratorio

Bootloader

Ciencias de la Computación VII

El objetivo de este laboratorio es adquirir conocimiento sobre el proceso de booteo de una computadora, la manipulación de espacios de memoria.

Requerimientos del Laboratorio:

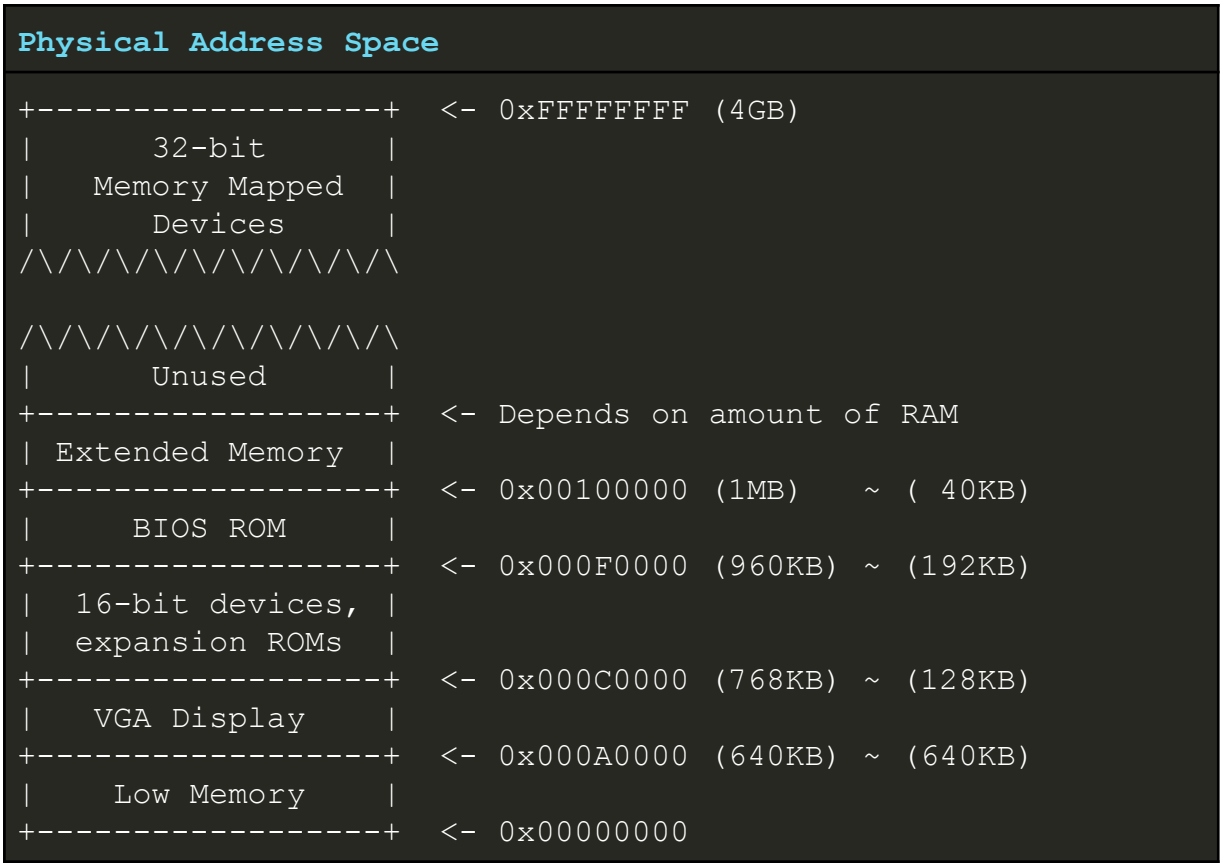
- Instalar
  - Docker y tener versión 16.04 (Lab 1)
  - apt-get update
  - apt-get install bochs
  - apt-get install bochs-x
  - apt-get install bochs-sdl
  - apt-get install libgtk2.0-dev libreadline-dev
- Configurar
  - .bochsrc (dentro de la carpeta zip)

Documentación

Un procesador **Intel 8088** de 16 bits direcciona hasta **1MB** de memoria física. Por lo tanto, el espacio de direcciones de memoria que podían ser accesibles por una de estas primeras PC comenzaba desde 0x00000 y terminaba en la dirección 0xFFFFF, actualmente un procesador **Intel i9-7900X** puede direccionar hasta 128GB de memoria física en una arquitectura de 64 bits.

Intel finalmente rompió la barrera de 1MB con el lanzamiento de los procesadores **80286** y **80386** los cuales llegaron a soportar un espacio físico de **16MB** y **4GB** respectivamente, también llegando a 32 bits con el último. Estos procesadores heredaron el diseño de la memoria de sus antecesores, dejando el esquema original de una sección **"Low Memory"** de 1MB con el objetivo de mantener compatibilidad con las aplicaciones desarrolladas para las arquitecturas anteriores.

Las PC's modernas tienen un agujero en el espacio físico desde la dirección 0x0A0000 hasta 0x100000, dividiendo la memoria RAM en dos secciones: **Memoria Convencional** (los primeros 640KB) y la **Memoria Extendida** (el resto de memoria). En el nuevo esquema, se planteó la posibilidad de utilizar la parte más alta de la memoria para que el BIOS pudiese administrar la mayoría de dispositivos externos.



- **Low Memory** Región de 640KB, era el único espacio de RAM que estas computadoras podían utilizar.
- **VGA Display Devices & Extension** Región de 320KB que va desde la dirección 0xA0000 hasta 0xF0000, estaba reservado para uso exclusivo del hardware de vídeo. En este espacio de memoria se almacenan los buffers de pantalla y el firmware de aquellos primeros monitores.
- **BIOS** Región de 64KB, que van desde la dirección 0xF0000 hasta 0xFFFFF. Una de las secciones más importantes reservada para el Sistema Básico de Entrada/Salida. Las primeras PC almacenaban el BIOS en una ROM, sin embargo las computadoras actuales almacenan este sistema un una memoria flash, facilitando su actualización. El BIOS es el responsable de realizar las tareas básicas de inicialización de todo el sistema, como por ejemplo, activar la tarjeta de video, chequear la cantidad de memoria instalada, etc. Luego de la fase de inicialización el BIOS se encarga de cargar las primeras rutinas del Sistema Operativo desde alguna memoria "booteable", como puede ser desde el disco duro, un CD-ROM o incluso desde la red. Cuando el BIOS termina este proceso de carga a memoria, cede el control a estas

rutinas, las cuales se encargan de continuar el proceso de inicialización del OS.

Una de las principales tareas de los Sistemas Operativos modernos es gestionar todos los aspectos relacionados con la organización de hardware y los complicados esquemas de memoria física que han ido evolucionando a través del tiempo.

El Boot Loader

Como hemos visto en clase, la mayoría de discos duros están divididos en secciones de 512 bytes llamadas Sectores. Un Sector es la unidad mínima direccionable, por lo que cada operación de lectura o escritura implica uno o más sectores y el acceso debe estar alineado al tamaño y ubicación de un sector. Si una unidad de almacenamiento es "Bootable", entonces el primer Sector de dicha unidad recibe el nombre de “Boot Sector”, debido a que por definición este Sector contiene el código necesario para iniciar el proceso de carga de un Sistema Operativo. Cuando el BIOS encuentra un "Bootable Sector", este se encarga de cargar los 512 bytes del sector y lo ubica en la dirección física 0x7C00 de la memoria RAM, luego la última instrucción que ejecuta es un "jump" a la dirección CS:IP to 0000: 7C00, si el proceso de carga fue exitoso, en esta dirección encontraríamos la primera instrucción especificada en el "Bootable Sector". De esta manera el BIOS entrega el control de la computadora al programa residente en la unidad como "Boot Loader".

La capacidad de "bootear" desde un CD-ROM apareció ya mucho tiempo después, y los arquitectos de estos sistemas ya tenían más experiencia y conocimientos para poder hacer una reingeniería en el proceso de "booteo". Como resultado, el proceso para arrancar un OS desde un CD-ROM es un tanto más complicado, pero más poderoso. Los CD-ROM utilizan sectores de 2048 bytes en vez de 512, por lo tanto el BIOS puede cargar a memoria una imagen más grande del boot loader.

Compilación

En la carpeta adjunta encontrarán el Archivos de Bootloader y algunos otros archivos de ayuda para compilar, recuerde que la clave para resolver todo es nunca dejar de leer documentación.

El archivo bochsrc en la sección Display Library debe agregar la siguiente línea:

```
display_library: sdl
```

Toda la información de la configuración de la máquina virtual del Bochs la pueden encontrar en el archivo .bochsrc

También se adjunta un script para instalar script GCC i386-elf pero no es el único método, lo motivamos a investigar.

Para realizar el Make puede ir a la carpeta Files, la salida será parecido a lo siguiente:

```
$ cd Files
$ make
     + as boot/boot.S
     + cc -Os boot/main.c
     + ld boot/boot
     boot block is 414 bytes (max 510)
     + mk obj/kern/bochs.img
```

Para levantar el Bochs, ejecuten el siguiente comando desde la carpeta Files:

```
Files $ bochs.
```

Preguntas e Implementación

- Describa el funcionamiento del código en la carpeta Boot. (boot.S y main.c)

- Modifique el programa **boot.S**, agregando dos funciones:
  - **ReadLine:** Lee del teclado una cadena de caracteres hasta que se presione la tecla Enter.
  - **WriteLine:** Imprime la cadena de caracteres ingresada por el usuario.
  - Si es necesario (lo es) cree otras funciones para que pueda administrar mejor su código.

- ¿Por qué no podemos implementar la impresión de la cadena "hello world" con un printf en main.c?

- ¿Cómo podemos cambiar el orden de "booteo" en la configuración del Bochs?, escriba el ejemplo de cómo quedaría la configuración si deseáramos "bootear" primero desde el floppy y luego desde el HD.

Material de Apoyo

Ver carpeta "x86 Assembly" en la sección de Material de Apoyo del GES.

Entrega

- El Laboratorio debe entregarse por medio del GES
- Debe entregar un ZIP con todos los archivos, no se aceptarán enlaces de lo contrario no tendrá nota.
- Las respuestas deben de venir en un PDF.