Arranque del Sistema

AISO - Introducción

Manuel Prieto-Matías
High Performance Computing Division
ArTeCS Group
Universidad Complutense de Madrid

Bootstraping

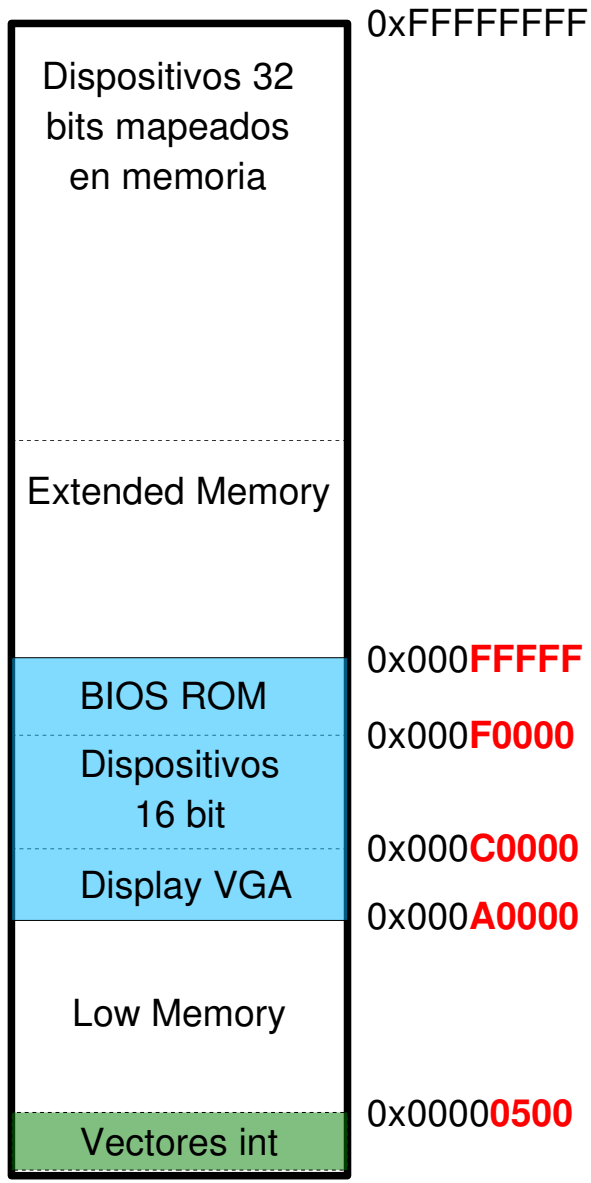
- Cuando arrancamos un computador personal el SO no esta presente en la memoria
  - En sistemas empotrados es frecuente disponer del SO completo cargado en una memoria no volátil
  - La mayor parte de los computadores sólo pueden ejecutar código que este cargado en memoria
- Es necesario utilizar un programa específico, el cargador de arranque – bootloader o bootstrap –
  - El objetivo del bootloader es cargar el SO desde algún dispositivo de almacenamiento secundario/red.
  - El Proceso de arranque es complejo y a menudo se utilizan cargadores con múltiples etapa

Arranque Arquitectura IBM PC (I)

- En la arquitectura IBM PC juega una papel esencial en el proceso de arranque la BIOS: Basic Input/Output Subrutines
  - Memoria no volátil – ROM en los primeros PC, Flash en la actualidad – en la que se incluyen facilidades para:
    - Arranque
    - Acceso básico a los dispositivos de E/S

Arranque Arquitectura IBM PC (II)

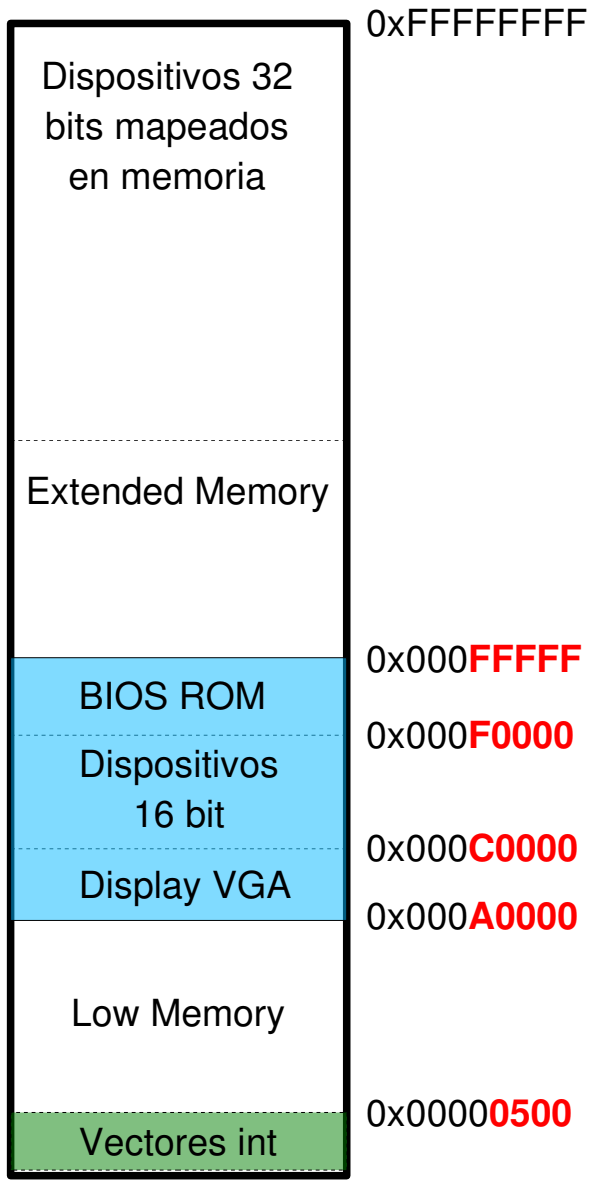
- El i8086 del IBM-PC original solo podía direccionar hasta 1 MB de memoria física (20 bits Bus Dir.)
  - La parte alta de la memoria (0x0A0000-0FFFF) solo lectura
    - La BIOS ocupa el último segmento de 64 KBs.
  - La parte baja de la memoria (0x000000-09FFFF) es la memoria de lectura-escritura donde se aloja:
    - El SO
    - Los programas de usuario (640KBs)
    - En los primeros 1280 bytes (0x0000-0500) la BIOS guarda los vectores de interrupción.
- Con el i80386 el bus de direcciones se extendió a 32 bits y se paso a direccionar 4GB de memoria
  - Por compatibilidad, la región entre los 640KB y 1MB se ha mantenido de solo lectura.



Manuel Prieto-Matías
Universidad Complutense de Madrid

Arranque Arquitectura IBM PC (III)

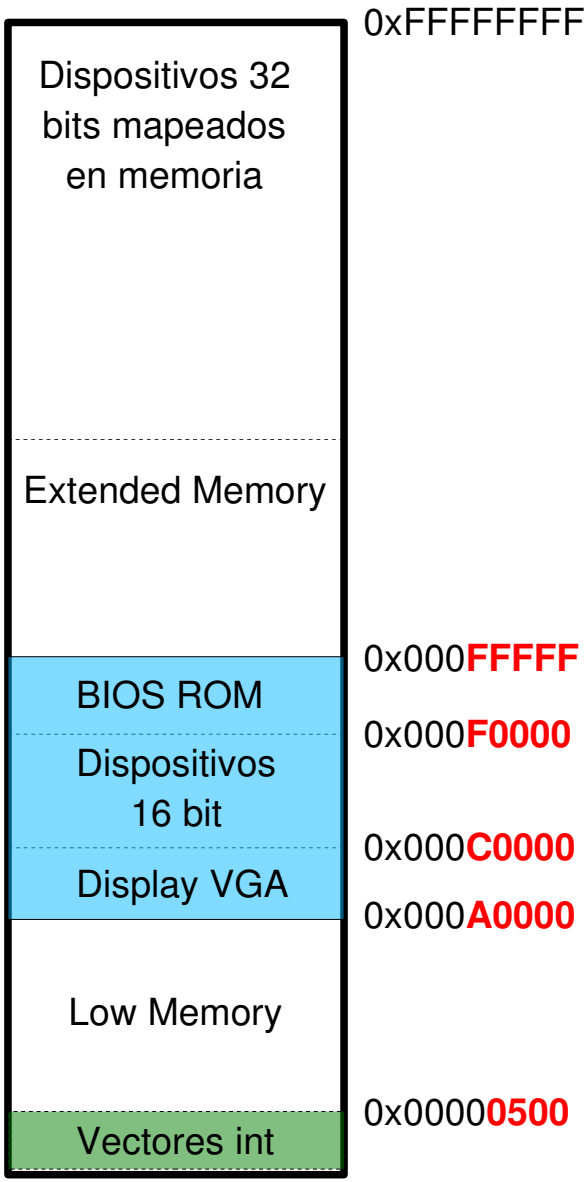
- Cuando arrancamos (reset) un i8086 , el “contador de programa” se inicializa con la dirección lineal 0xFFFF0
  - En el i8086 se utilizaba un espacio de direcciones segmentado en el que las direcciones lógicas vienen dadas por Segmento:Offset. La dirección lineal correspondiente viene dada por:
    - 16xSegmento + Offset (CS =0xF000, IP =0xFFF0 → 0xFFF0)
  - En el i80386s y posteriores, la dirección es 0xFFFFFFF0
    - Las lineas del bus de direcciones A20-A31 se inicializan a 1
- En dicha dirección se incluye una instrucción de salto al programa de arranque de la BIOS
  - Primera Instrucción: jmp far f000:e05b



Manuel Prieto-Matías
Universidad Complutense de Madrid

Arranque BIOS (I)

- Qué hace el programa de arranque de la BIOS?
  - Test del sistema: *power-on self test* (POST).
    - Si hay errores, se detiene el arranque y se emite un código de diagnóstico por el altavoz
  - Saltar a los "programas de inicialización" de distintos dispositivos
    - Controladora de vídeo (0xC000): suele mostrar información en el monitor sobre la controladora de vídeo
    - Discos duros IDE/ATA (0xC800)
    - …
  - Muestra Información de la BIOS/PC por la pantalla
    - Fabricante de la BIOS y del PC (logos), versión, fecha, tecla que se debe pulsar para saltar al programa de configuración (BIOS)

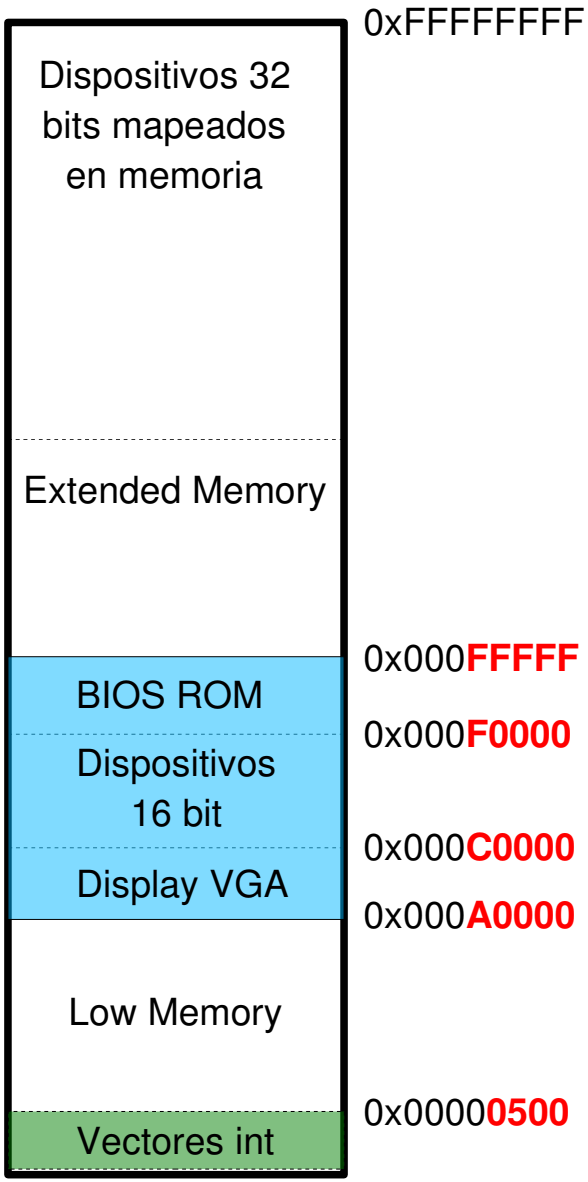


Manuel Prieto-Matías
Universidad Complutense de Madrid

Arranque BIOS (II)

- ...
- Diagnostico de la memoria RAM
- Inventario de periféricos, asignación de recursos (*Plug and Play*, …)
- Se muestra una tabla descriptiva de la configuración del PC
- …

- Finalmente, se busca un dispositivo de arranque de acuerdo a una secuencia de dispositivos predeterminada/configurable
  - Se carga en memoria (0x7C00-0x7DFF) el *bootsector* (512B)
  - Se comprueba que es un *bootsector* válido (número mágico)
    - Se cede el control si hay éxito o ...
    - … o "*No boot disk or disk error, Replace and stroke a key*",



Manuel Prieto-Matías
Universidad Complutense de Madrid

El Master Boot Record (I)

- La principal dificultad que presentan los bootloaders está en las limitaciones de tamaño.
  - En los discos duros, el bootloader se encuentra almacenado en el primer sector del disco, el Master Boot Record (MBR). En los 512 bytes del MBR se incluye:
    - 446 bytes para el código del programa MBR.
    - Tabla de particiones primaria (4 entradas de 16 bytes)
    - un número mágico de 2 bytes (0xAA55) que identifica al sector como MBR
  - En arranque desde CD surgió con posterioridad y se aprovecho para rediseñar el proceso de arranque
    - Los CD-ROMs utilizan un sector de 2048 Bytes y la BIOS pueda cargar en memoria un bootloader más complejo antes de transferirle el control (El Torito Bootloader)

El Master Boot Record (II)

- Qué es lo que puede hacer/hace el programa MBR?
  - Examinar la tabla de particiones y buscar la partición activa.
  - Interactuar con el usuario para configurar parámetros de arranque.
  - Cambiar el modo de ejecución – cambiar de modo real a modo protegido –
  - En último término el programa MBR debe cargar en memoria la siguiente etapa del proceso de arranque. Los detalles dependen del cargador y el sistema operativo que consideremos.
    - En la arquitectura IBM PC ha sido frecuente que los cargadores hagan uso de las facilidades de la BIOS para acceso a disco de bajo nivel: Servicio INT 13.

Grub Legacy (I)

- El cargador que suele utilizar los sistemas Linux es Grub (GRand Unified Bootloader)
- El cargador que se aloja en el área de código del MBR se conoce en GRUB como GRUB stage 1.
  - Su misión es cargar la siguiente etapa del del proceso de arranque, GRUB stage 1.5 o GRUB stage 2, dependiendo de la instalación.
  - En la instalación de GRUB, se fija el sector del disco donde comienza la siguiente etapa

Grub Legacy (II)

- GRUB Stage 1.5
  - Es habitual alojar Grub stage 1.5 en el primer sector de la partición activa del disco (el volume boot sector), aunque existen otras alternativas.
  - Otra posibilidad es utilizar el resto de sectores del primer cilindro del disco a continuación del MBR (zona de compatibilidad con DOS).
    - La imagen de MS-DOSdebía estar alineada al comienzo de un cilindro. Para una geometría de disco estándar de 64 sectores por pista, la zona de compatibilidad DOS ocupa 32 KB – 63 sectores de 512 KB – .
  - Existe una etapa GRUB 1.5 específica para cada tipo de sistema de ficheros soportado por GRUB
    - GRUB 1.5 puede localizar directamente en el sistema de ficheros de la partición activa la siguiente etapa (no es fijar el sector en la instalación de Grub)

Grub Legacy (III)

- GRUB Stage 2
  - En esta etapa final es donde reside la *lógica* del GRUB.
  - Es el encargado de presentar el menu de opciones al usuario y ejecutar los correspondientes comandos que se indiquen en el fichero de script *menu.lst*.
  - Cómo GRUB Stage 1.5 puede manejar directamente el sistema de ficheros de la partición activa y puede localizar en el disco el kernel directamente a partir de una ruta (configurable)

Grub 2.0

- La estructura de GRUB ha cambiado notablemente en la versión 2.0
  - En lugar de constar de las 3 imágenes utilizadas en GRUB Legacy (Stage 1, Stage 1.5 y Stage 2) se compone de una serie de módulos
    - Se pueden combinar de diferentes formas. Ejemplo:
      - Stage1: boot.img
      - Stage 1.5 (core image): diskboot.img + kernel.img + + ext2.mod
      - Stage 2: normal.mod (grub.cfg)+ chain.mod

AISO Introducción

Versión 0.1

© **Manuel Prieto Matias**

Este documento (o uno muy similar) esta disponible en
https://cv2.sim.ucm.es/moodle/course/view.php?id=3235