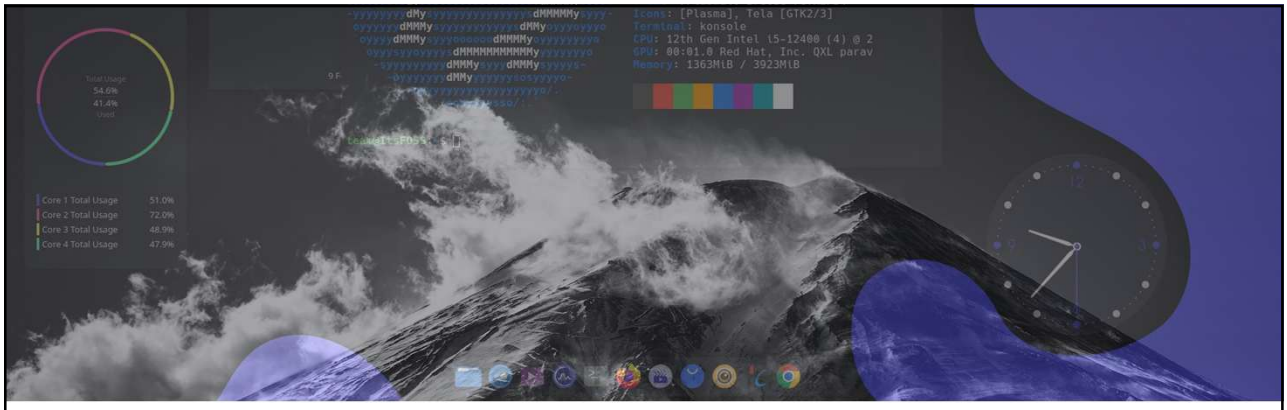


SISTEMAS OPERATIVOS

Conceitos sobre inicializacao da maquina

António Godinho

1

ARRANQUE DO SISTEMA OPERATIVO

O sistema operativo age como um gestor da máquina e tem mais “poderes” do que qualquer aplicação regular (ditas aplicações utilizador).

Porquê? O que lhe dá estes “poderes” privilegiados?

O processador assume características e capacidades diferentes consoante estiver a executar código do SO ou de uma aplicação. A configuração destas características é feita pelo SO;

O sistema operativo é o primeiro programa completo a ser carregado para a máquina e configura o hardware de forma a que quando é a sua vez de se executar, tem todas as capacidades da máquina, e quando é a vez dos restantes programas se executarem, a máquina assume um comportamento de capacidades reduzidas.

Basicamente, o sistema chega à máquina primeiro e "arranja" o cenário de forma a quem os programas tenham apenas os privilégios que o sistema entende dar.

Assunto destes slides: como se processa a inicialização da máquina e o carregamento do sistema operativo?

2

2

ARRANQUE DO SISTEMA OPERATIVO

Conceitos necessários

Firmware, BIOS, POST

Discos, Partições, MBR

Bootstraping, Boot loader, chainloading

UEFI, GPT, Secure boot

3

3

FIRMWARE

Firmware

Software armazenado de forma permanente (não precisa de alimentação permanente), inicialmente ROM, agora memória flash

Contêm rotinas utilitárias para controlar aspectos do equipamento e rotinas para inicializar esse equipamento e colocá-lo num estado inicial coerente: Basic Input Output System (BIOS)

Efectua um teste simplificado ao equipamento: Power On Self Test (POST)

Em equipamentos simples, pode conter a totalidade do software necessário à operação do dispositivo.

Nos computadores habituais

- Contém rotinas de arranque inicial da máquina, rotinas para interacção com dispositivos standard (ex.: discos, teclado, etc.)
- Corre normalmente em modo simplificado (não privilegiado) do processador e as suas capacidades de gestão são limitadas.
- Inclui normalmente o software habitualmente designado por BIOS

4

4

BIOS

BIOS (Basic Input Output System)

Parte habitual do software habitualmente designado de firmware.

Parte da norma estabelecida para os IBM PC e compatíveis.

Apresenta algumas limitações face ao equipamento moderno e está em processo de substituição pela norma UEFI

Contém software para:

- Inicialização da máquina
- Interacção com dispositivos standard normalmente presentes em qualquer máquina.
- Carregar ou iniciar a carga do sistema operativo

5

5

TAREFAS DA BIOS

Inicialização da máquina

Identificação e teste dos componentes principais da máquina (ex., memória, teclado, etc.). Tarefa normalmente designada por POST

(Power On Self Test)

- Enumeração dos dispositivos presentes e configuração de cada um deacordo com parâmetros standard
- Passagem de controlo ao software que inicia a carga do sistema operativo (início do processo de bootstrap)

6

6

TAREFAS DA BIOS

Interação com componentes e dispositivos standard

Materializado sob a forma de um conjunto de rotinas acessíveis ao sistema operativo e a qualquer outro programa

Podem agir como substituo de funções sistema em cenários em que o sistema operativo é muito simples

Exemplo: IBM PC/MSDOS.

Rotinas da BIOS (exemplos): acessíveis via int 10H, int 09H, etc.

Limitadas na sua ação por:

Estarem preparadas para dispositivos standard, e portanto não aproveitam capacidades específicas ou otimizadas de hardware melhor.

Correrem habitualmente com o processador em modo não privilegiado (menor capacidade de ação)

Estas rotinas são essenciais ao processo de arranque e configuração da máquina, sendo apenas dispensáveis se o sistema operativo tiver software que as substitua e apenas depois deste estar presente em memória

7

7

TAREFAS DA BIOS - BOOTSTRAPING

Início do processo de bootstrap

Leitura dos parâmetros de configuração da máquina em memória não volátil para determinação de qual o dispositivo por onde se inicia a carga do sistema (ex:, disco, CD, USB, etc.)

Análise do dispositivo em questão (o disco, CD, o que for) e leitura do *bootloader* para memória

Passagem do controlo da execução para o *bootloader* que foi carregado. Deste ponto em diante a BIOS age essencialmente como repositório de rotinas utilitárias. O processo de *bootstrap* prossegue com a execução do *bootloader*.

8

8

BOOTSTRAPING

Bootloader

- Pequeno programa armazenado em memória secundária que começa o processo de carga do sistema
- Razoavelmente standard quanto à localização e formato mas ainda assim algo dependente do sistema operativo e bastante dependente da arquitetura
  - Tamanho standard típico: menos que 512 bytes.
    - Sendo um tamanho bastante pequeno, o *bootloader* tipicamente carrega outros programas (cada vez mais complexos e dependentes do sistema operativo) e transfere o controlo a esses programas. O efeito é o de uma sequência de peças de dominó a caírem em que cada uma empurra a seguinte.
  - Locais típicos: início do disco / da partição ativa do disco
- A sua localização depende do tipo de dispositivo
  - Os exemplos e descrição seguintes assumem que o dispositivo é um disco rígido (cenário mais habitual)

9

9

DISCOS RIGIDOS

Organização dos discos rígidos

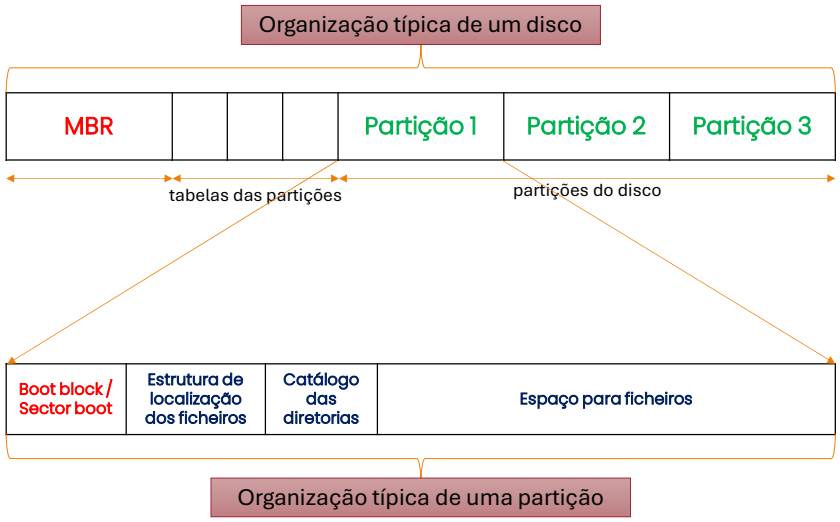
Os discos rígidos estão normalmente organizados em **partições**.

- Cada partição pode ter um sistema de ficheiros diferente
- Cada partição pode hospedar sistema operativo independente
- Alguns sistemas de ficheiros conseguem abarcar várias partições dando a ideia da existência de um só sistema de ficheiros

10

10

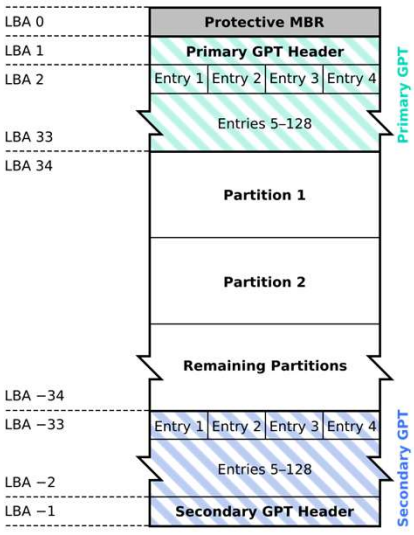
DISCOS RIGIDOS



11

11

ORGANIZAÇÃO DE DISCOS RÍGIDOS - GPT



Discos GPT

• Possíveis em *chipsets* e sistemas compatíveis com UEFI

GPT: substitui a lógica antes atribuída ao MBR, mas essencialmente é a mesma coisa com mais flexibilidade e potencialidades

Passa a poder haver 128 partições

O arranque do sistema pode ser feito de forma mais flexível e através de pequenos programas (*loaders*) existentes na primeira partição (partição de sistema)

Suportam características mais evoluídas tais como certificados de *secure-boot*

12

12

DISCOS RÍGIDOS - PARTIÇÕES

MBR - Master boot record

- Normalmente um sector (pode ser mais), sendo o primeiro sector do disco
- Contém um pequeno programa que é um *bootloader*.
  - Este programa pode, em teoria, carregar o sistema diretamente, mas normalmente passa o controlo para um outro *bootloader* que se encontra no sector *boot* da partição ativa
- Mantém informação acerca da geometria do disco:
  - Quantas partições existem
  - Onde começa e acaba cada partição
  - Qual a partição que está ativa (qual a que contém o S.O. de arranque)

13

13

SISTEMAS DE FICHEIROS - PARTIÇÕES

Sector boot

- Normalmente 1 sector (pode ser mais), sendo o primeiro sector de uma partição
- Contém um pequeno programa que é um *bootloader* e dá inicio ao arranque do sistema (se a partição for a ativa)
  - Normalmente o sistema que está nessa partição, mas pode ser outro
  - Este *bootloader* é, normalmente, específico ao sistema operativo, enquanto que o *bootloader* no MBR é, normalmente, mais genérico.
  - A instalação de *bootloader* deve ser feita, sempre que possível, na partição do sistema operativo em questão, deixando o MBR intacto tanto quanto possível uma vez que esse diz respeito ao disco todo.
- Contém informação variada acerca da partição
  - Localização acerca das outras componentes
  - Tamanho dos blocos lógicos (clusters)

14

14

BOOTSTRAPING

Processo de carga do sistema

1. A BIOS identifica o dispositivo onde deve procurar o sistema operativo (informação armazenada na configuração em memória não volátil)

   Assume-se nestes exemplos de que se trata de um disco rígido
2. A BIOS lê o MBR para memória e transfere a execução para o programa que se encontra dentro do MBR.
3. O programa no MBR determina qual a partição ativa (tabela existente no MBR), lê o sector *boot* dessa partição para memória e transfere-lhe a execução.
4. O programa no sector *boot* prossegue a carga do sistema carregando os ficheiros do sistema presente na sua partição, ou apresentando um menu possibilitando passar para outro sector *boot* de outra partição (dual *boot*)

   O processo de um sector *boot* carregar o sector de outra partição e passar- lhe o controlo é designado de ***chainloading***

Tanto o programa do MBR como o do sector *boot* da partição podem apresentar opções de escolha para *dual boot*. Normalmente esta escolha é feita a nível do sector *boot* pois as alterações no MBR afetam o disco todo (por oposição de afetar apenas uma partição) e portanto são de evitar tanto quanto possível

15

15

DUAL BOOT

Consiste em ter vários sistemas operativos na máquina (normalmente em partições diferentes) e ter a possibilidade de arrancar a máquina com qualquer um desses sistemas operativos (em alternativa)

Método

Configuração do *bootloader* no sector *boot* para apresentar um menu que possibilite:

Carregar os ficheiros do sistema presentes na partição em questão

Ou então

Duplicar o procedimento efetuado pelo programa no MBR e

1. Carregar um outro sector boot para memória
2. Passar a execução para o código desse outro sector *boot*

Este procedimento é conhecido com ***chainloading***

16

16

UEFI

UEFI - Unified Extensible Firmware Interface

Norma que visa substituir a anterior (BIOS)

Sendo mais moderna, permite um conjunto alargado de funcionalidades mais em linha de conta com o hardware recente, por exemplo:

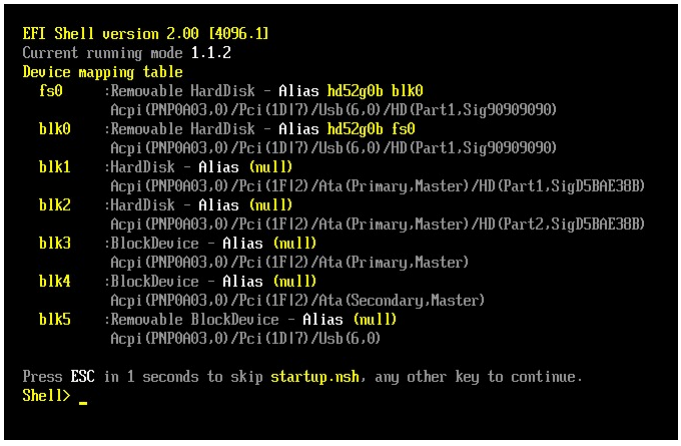
- Discos GPT (**GPT** = **G**UID **P**artition **T**able, GUID = Global Unique IDentfier)
  - São discos normais particionados segundo um esquema diferente daquele possível com o MBR.
- Secure boot
  - Utilização de certificados digitais nos loaders para controlar o acesso de um sistema à máquina. Um sistema que não tenha um certificado válido quando comparado com os que estão presentes na memória da máquina, será impedido de se carregar e executar.
  - Os certificados podem ser adicionados e a opção de secure boot pode (eventualmente) ser desligada (acções do administrador)
- Rotinas de interface com hardware mais evoluídas de carácter semelhante a device drivers em linguagem independente do processador
- Suporte melhorado para placas gráficas recentes

17

17

ORGANIZAÇÃO DE DISCOS RÍGIDOS GPT

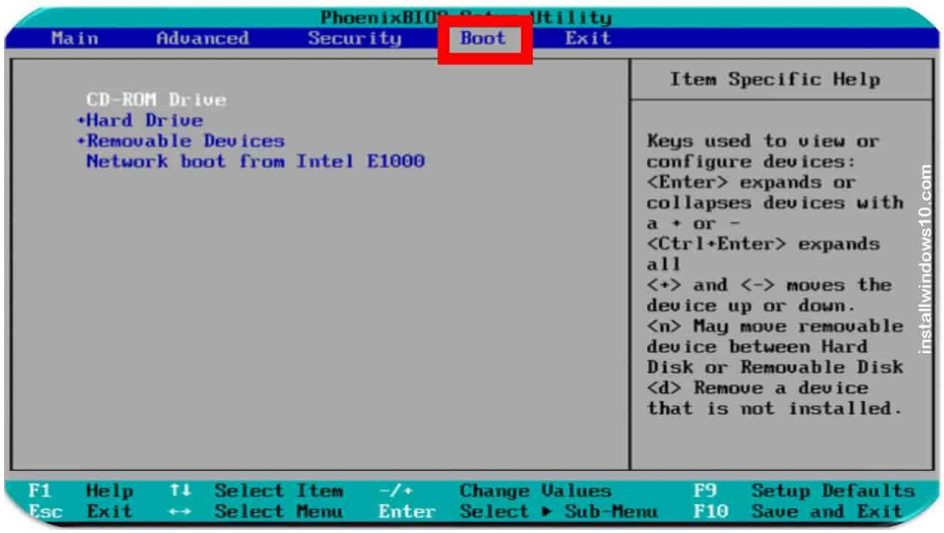
Aspeto de uma Shell de arranque num sistema UEFI



18

18

BIOS



19

19

UEFI

20

Exemplos de implementações de alguns dos principais fabricantes.

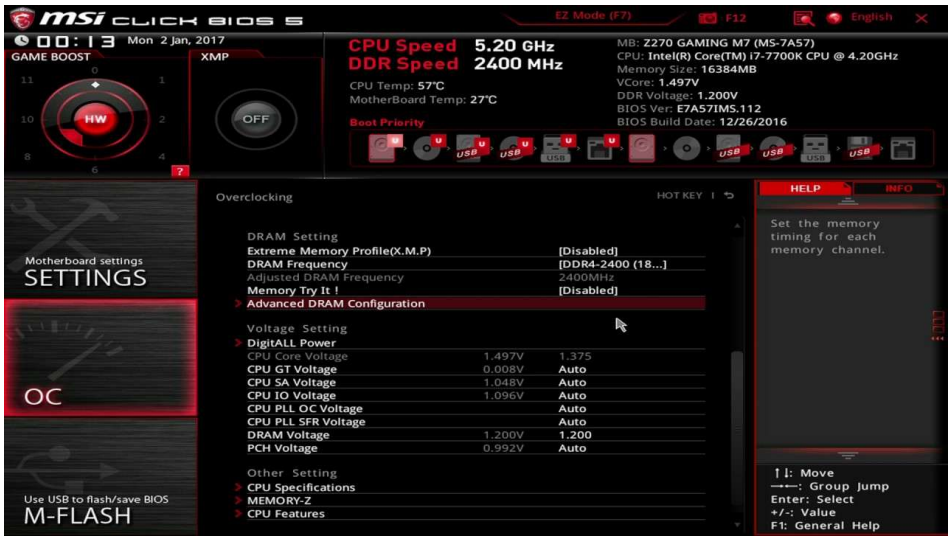
20

UEFI -MSI

21

21

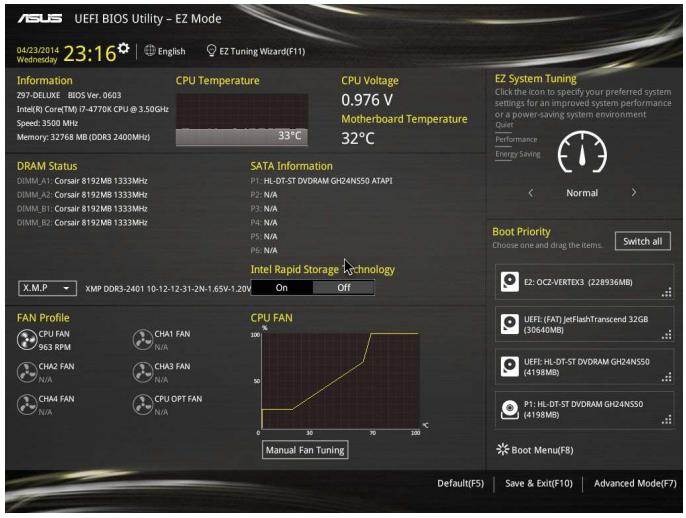
UEFI -MSI



22

22

UEFI - ASUS



23

23

UEFI - ASUS

24

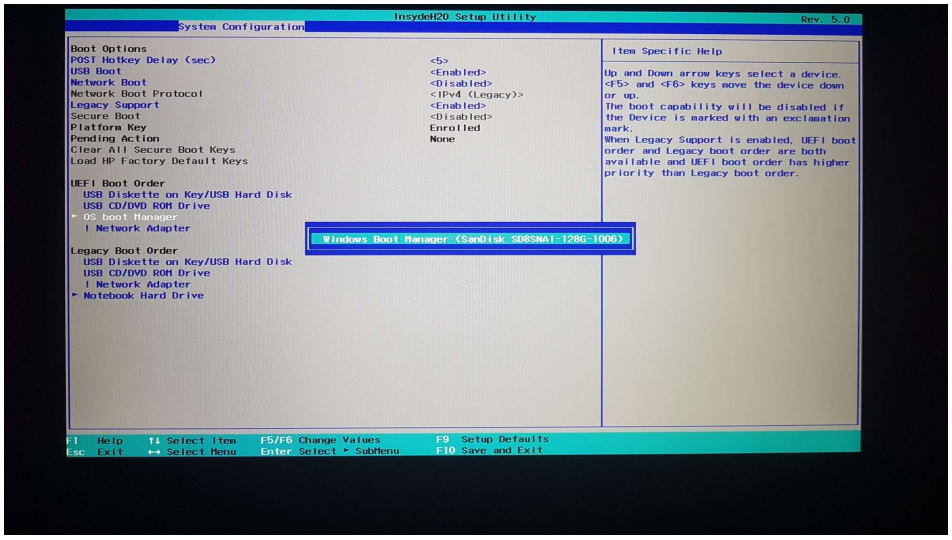
24

UEFI - ASUS

25

25

UEFI - HP



26

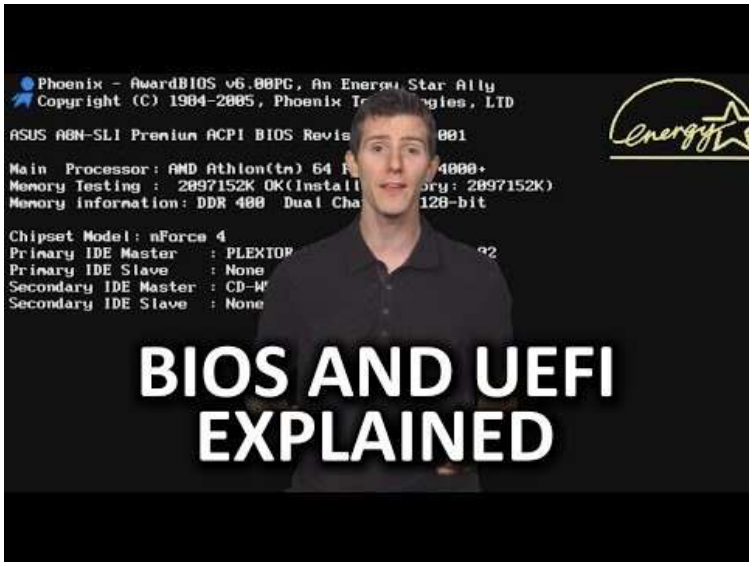
26

UEFI - GIGABYTE

27

UEFI - GIGABYTE

28



https://www.youtube.com/watch?v=zlYkol851dU

29

29