

# **FACULDADE DE TECNOLOGIA DE GARÇA**

## **ANÁLISE E DESENVOLVIMENTO DE SOFTWARE**

### **SISTEMAS OPERACIONAIS II**

#### **Inicialização do sistema**

Nos primeiros computadores, para inicializar o sistema, era necessário inserir uma fita de papel onde estava contido um programa de boot ou carregar manualmente um programa de boot utilizando os controles de endereço do painel frontal/dados/controle. Os computadores atuais são equipados com recursos que simplificam o processo de boot, embora não necessariamente o deixem mais simples.

Será demonstrada inicialmente uma visualização de alto nível de boot do Linux para que se tenha uma visão geral. Em seguida, serão revisados os processos que ocorrem separadamente em cada uma das etapas. As referências de origem durante esse processo ajudarão a navegar pela árvore de kernel e a compreender melhor esse processo.

Softwares armazenados na ROM

- BIOS: Basic Input/Output System
- POST: Power-On Self Test
- Setup: Configuração

#### **BIOS (Basic Input/Output System –**

(Sistema Básico de Entrada e Saída):

“Ensina” o processador a trabalhar com os periféricos mais básicos do sistema, tais como os circuitos de apoio, unidade de disquetes e o vídeo em modo texto.

#### **POST (Power On Self Test – Auto Teste ao Ligar):**

Um auto teste feito sempre que ligamos o micro que executa as seguintes rotinas, sempre que o micro é ligado:

- Identifica a configuração instalada.
- Inicializa todos os circuitos periféricos de apoio (chipset) da placa-mãe.
- Inicializa o vídeo.
- Testa a memória.
- Testa o teclado.
- Carrega o sistema operacional para a memória.
- Entrega o controle do processador ao sistema operacional.

#### **SETUP (Configuração):**

Programa de configuração de hardware do computador;

– normalmente chamado pressionando um conjunto de teclas durante o POST (geralmente basta apertar a tecla DEL durante a contagem de memória; esse procedimento, contudo, pode variar de acordo com o fabricante).

## Visão Geral

A figura abaixo mostra uma visualização estratégica.

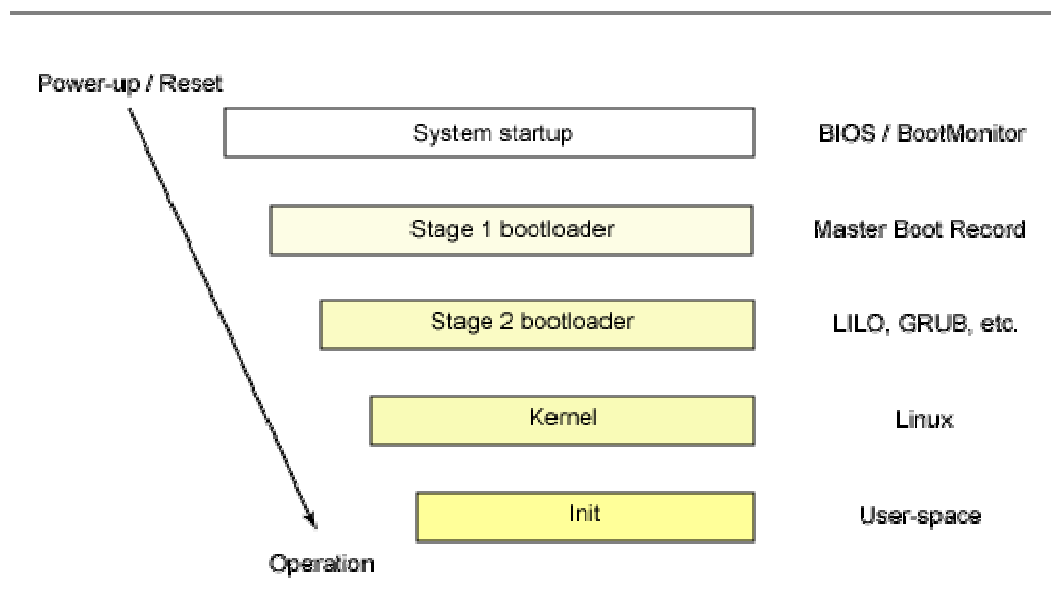


Figura 1. Visualização Estratégica do Processo de Inicialização do Linux

Quando um sistema efetua boot pela primeira vez, o processador executa um código em um local conhecido. Em um Computador Pessoal (PC), esse local é o Sistema Básico de Entrada/Saída (BIOS), que está armazenado na memória flash, na placa-mãe. A Unidade Central de Processamento (CPU) em um sistema embarcado solicita ao vetor de reconfiguração que inicie um programa em um endereço conhecido em flash/ROM. Nos dois casos, o resultado é o mesmo. Como os PCs oferecem bastante flexibilidade, o BIOS precisa determinar quais dispositivos são candidatos ao boot.

Quando um dispositivo de boot é encontrado, o loader de boot de primeiro estágio é carregado na RAM e executado. Esse loader de boot tem menos de 512 bytes (um único setor), e sua tarefa é carregar o loader de boot de segundo estágio.

Quando o loader de boot de segundo estágio está na RAM e em execução, uma tela inicial é geralmente exibida e o Linux e o disco RAM inicial opcional (sistema de arquivo raiz temporário) são carregados na memória. Quando as imagens são carregadas, o loader de boot de segundo estágio passa o controle para a imagem do kernel e o kernel é descompactado e inicializado. Neste estágio, o loader de boot de segundo estágio verifica o hardware do sistema, enumera os dispositivos de hardware anexados, monta o dispositivo raiz e, em seguida, carrega os módulos de kernel necessários. Ao ser concluído, o primeiro programa de espaço de usuário (init) inicia e a inicialização do sistema de alto nível é executada.

Em resumo, é assim que funciona o boot do Linux. Abaixo alguns dos detalhes do processo de boot do Linux.

## Detalhes da inicialização do Sistema

O estágio de inicialização do sistema depende do hardware no qual o Linux está efetuando boot. Em uma plataforma integrada, um ambiente de autoinicialização é utilizado quando o sistema é ligado ou reconfigurado. Como exemplos, temos U-Boot, RedBoot e MicroMonitor da Lucent. As plataformas integradas normalmente vêm de fábrica com um monitor de boot. Esses programas residem em uma região especial da memória flash no hardware de destino e fornecem os meios para fazer download de uma imagem do kernel Linux na memória flash e, em seguida, executá-la. Além da capacidade de armazenar e executar boot em uma imagem do Linux, esses monitores de inicialização executam alguns níveis de teste do sistema e a inicialização do hardware.

Em um PC, o processo de inicialização do Linux começa no BIOS, em 0xFFFF0. A primeira etapa do BIOS é Power-On Self Test (POST). A tarefa do POST é executar uma verificação do hardware. A segunda etapa do BIOS é enumerar e inicializar o dispositivo local.

Dados os diferentes usos das funções do BIOS, temos duas partes: o código POST e os serviços de tempo de execução. Depois de concluído, o POST é apagado da memória, mas os serviços de tempo de execução do BIOS permanecem e ficam disponíveis ao sistema operacional de destino.

Para inicializar um sistema operacional, o tempo de execução do BIOS procura por dispositivos ativos e inicializáveis na ordem de preferência definida pelas configurações de Complementary Metal Oxide Semiconductor (CMOS). Um dispositivo de boot pode ser um disco flexível, um CD-ROM, uma partição de um disco rígido, um dispositivo na rede ou até mesmo um memory stick flash USB.

Normalmente, o Linux tem boot executado a partir de um disco rígido, onde Master Boot Record (MBR) contém o loader de boot primário. O MBR é um setor de 512 bytes, localizado no primeiro setor do disco (setor 1 do cilindro 0, cabeçote 0). Depois que o MBR é carregado na RAM, os campos do BIOS o controlam.

### Loader de Boot de Estágio 1

O loader de boot primário, que reside no MBR, é uma imagem de 512 bytes contendo o código do programa e uma pequena tabela de partição. Os primeiros 446 bytes são o loader de boot primário, que contém o código executável e um texto de mensagem de erro. Os 64 bytes seguintes formam a tabela de partição, que contém um registro para cada uma das quatro partições (cada uma com 16 bytes). O MBR termina com 2 bytes, que são definidos como o número(0xAA55). O número mágico funciona como verificação de validação do MBR.

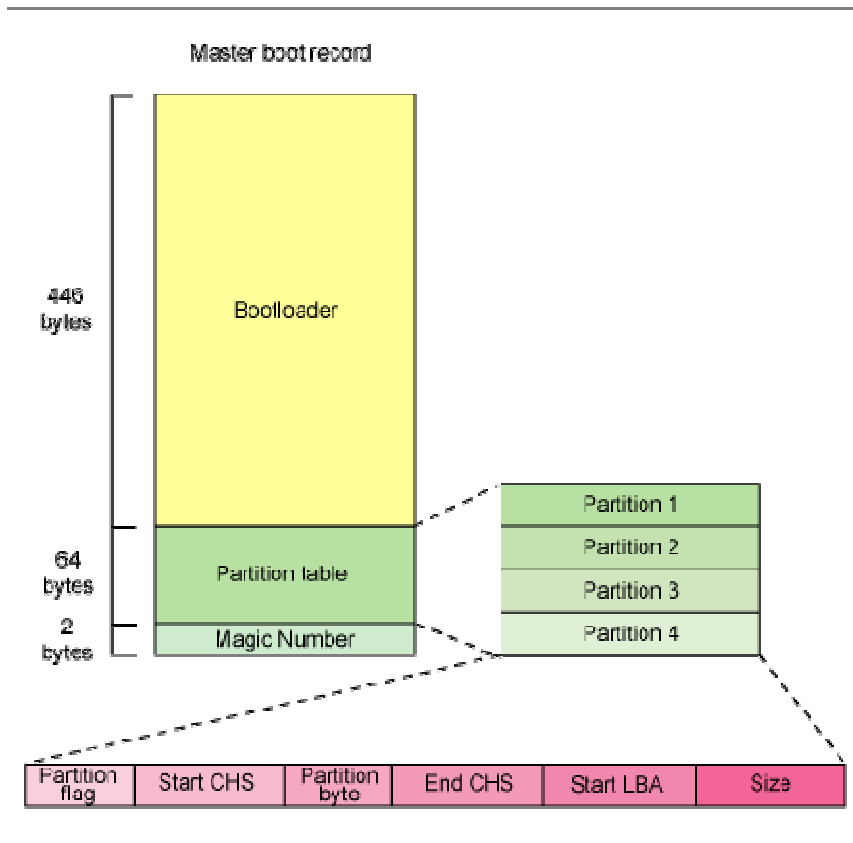


Figura 2. Anatomia do MBR

A função do loader de boot primário é encontrar e carregar o loader de boot secundário (estágio 2). Isso é feito consultando a tabela particionada de uma partição ativa. Ao localizar uma partição ativa, ele faz uma varredura das partições restantes na tabela, a fim de certificar-se de que todas estão inativas. Quando isso é verificado, o registro de boot da partição ativa é lido a partir do dispositivo para a RAM e executado.

## Loader de Boot de Estágio 2

O loader de boot secundário, ou de segundo estágio, pode ser chamado de loader de kernel. Neste estágio, a tarefa é carregar o kernel Linux e o disco RAM inicial opcional.

## Loaders de Boot de Estágio GRUB

O diretório /boot/grub contém os loaders de boot stage1, stage1.5, e stage2, bem como diversos loaders alternativos (por exemplo, os CR-ROMs usam iso9660\_stage\_1\_5).

Os loaders de boot de primeiro e segundo estágios combinados são chamados de Linux Loader (LILO) ou GRand Unified Bootloader (GRUB) no ambiente x86 do PC. O LILO apresenta algumas desvantagens que foram corrigidas no GRUB, desta forma o GRUB é mais adequado para o estudo.

O ponto alto do GRUB é que ele inclui o conhecimento dos sistemas de arquivo Linux. Em vez de utilizar setores brutos no disco, como o LILO faz, o GRUB pode carregar um kernel Linux a partir de um sistema de arquivos ext2 ou ext3. Isso é feito transformando o loader de boot de dois estágios em um de três estágios. O estágio 1 executa boot em um loader de boot de estágio intermediário (1,5), que compreende o sistema de arquivos específico, contendo a imagem do kernel Linux. Como exemplos, temos reiserfs\_stage1\_5 (para carregar a partir de um sistema de arquivos com registro de mudanças Reiser) ou ue2fs\_stage1\_5 (para carregar a partir de um sistema de arquivos ext2 ou ext3). Quando o loader de boot do estágio intermediário estiver carregado e em execução, o loader de boot do estágio 2 poderá ser carregado.

Quando o estágio 2 estiver carregado, o GRUB poderá, mediante solicitação, exibir uma lista dos kernels disponíveis (definidos em /etc/grub.conf, com soft links de /etc/grub/menu.lst e /etc/grub.conf). É possível selecionar um kernel e até aditá-lo com parâmetros adicionais de kernel. Opcionalmente, é possível utilizar shell de linha de comandos para obter um maior controle manual sobre o processo de boot.

Com o loader de boot de segundo estágio na memória, o sistema de arquivos é consultado e a imagem de kernel padrão e a imagem initrd são carregadas na memória. Com as imagens prontas, o loader de boot de estágio 2 chama a imagem de kernel.

---

FONTE:

<http://www.ibm.com/developerworks/br/library/l-linuxboot/>