Prática de Laboratório de Sistemas Operacionais

Profa. Gisele S. Craveiro

Giovane Melo Andrade, 8516688

**SYSTEM STARTUP**

**Plano de Aula**

**Tema: System Startup - Visão Geral**

**Objetivos:**

Realizar uma visão geral de todo processo de inicialização do computador desde do momento que é apertado o botão de POWER até o momento que é apresentado a área de trabalho pronta para ser utilizado pelo usuário. Passando pela parte de Hardware até chegar na abertura do sistema operacional.

**Introdução**

System startup ou processo de inicialização ou mas conhecido como BOOT são processos que juntos inicializam o computado e o preparam para ser utilizado.

Para entendermos melhor todo processo de inicialização do S.O é necessário entender desde do momento que apertamos o botão de POWER (Ligar).

Ao ligar o computador, a primeira parte a ser ligada é a BIOS(Basic Input/Output System — Sistema Básico de Entrada/Saída em português). A Bios é um dos três programas que ficam no chip de memória rom, sendo os outros dois o POST e o SETUP.

**Histórico**

*“Na época do MS-DOS era bem conhecida a divisão entre memória real (os primeiros 640 KB da memória RAM) e memória extendida (do primeiro MB em diante, englobando quase toda a memória instalada). O MS-DOS rodava em modo real, onde o processador trabalha simulando um 8088 (o processador usado no XT) que era capaz de acessar apenas 640 KB de memória. Mesmo os processadores modernos conservam este modo de operação, mas os sistemas operacionais modernos rodam inteiramente em modo protegido, onde são usados todos os recursos da máquina.*

*O espaço entre os primeiros 640 KB, onde termina a memória real e os 1024 KB onde começa a memória extendida é justamente reservado para o BIOS da placa mãe. Ele é originalmente gravado de forma compactada num chip de memória flash instalado na placa mãe e fica descompactado neste espaço reservado (chamado de shadow RAM) durante o uso.”*

**BIOS, POST e SETUP**



Figura 1 - Chip de Memória ROM (***R****ead-****O****nly* ***M****emory*).

Quando a Bios é inicializada, a mesma chama o programa POST (Power on Self Test) que realiza os mais diversos testes em todo hardware, como por exemplo: contagem de memória, verificação de dispositivos ligados à placa - mãe e outros. Caso o POST encontre algum erro será sinalizado o mesmo com um aviso sonoro ou visual (luzes de led na placa - mãe).



Figura 2 - POST - Verficação de Hardware

Após o processo de POST, a Bios inicializa as outras Bios dos demais dispositivos, por exemplo placa de vídeo, HD e outros. Após isso é procurado o dispositivo de BOOT de acordo com a sequência definida pelo CMOS SETUP (CMOS é a abreviação de "Complementary Metal Oxide Semiconductor").

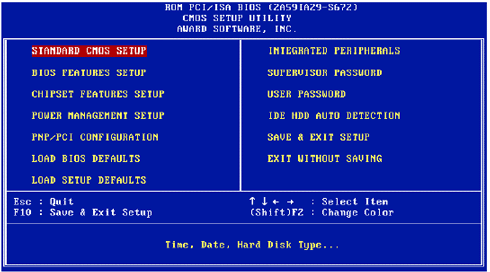


Figura 3 - CMOS Setup

**Inicialização do S.O**

Após os testes, a BIOS lê o primeiro setor do disco, a MASTER BOOT RECORD (MBR), também conhecida como trilha zero.

O MBR por sua vez contém o Lilo ou o Grub, ambos possuem a função de carregar o kernel do Linux (é um arquivo compactado e somente leitura), o /boot/vmlinuz. Este arquivo é carregado em uma área na memória RAM reservado. Sendo este arquivo alterado apenas em uma recopilação ou alteração na versão do sistema

*“O LILO é um acrônimo para a expressão inglesa LInux LOader que, em português, significa carregador de*[*Linux*](https://www.vivaolinux.com.br/linux/)*. É um bootloader, gestor de arranque ou gerenciador de Sistemas Operacionais. Permite configurar o arranque (Boot) de múltiplos sistemas operativos na mesma máquina (não simultaneamente).  
  
Para isso, instala-se nos primeiros 512 bytes de qualquer dispositivo de armazenamento (MBR), imediatamente antes da tabela de partições. Como tal, é independente do(s) sistema(s) operativo(s) instalado(s) e seus sistemas de arquivos, mas é, obrigatoriamente, escrito em código-máquina, é fortemente dependente da plataforma.”*  
O LILO permite escolher um de dezesseis kernels possíveis, cada um contendo opções específicas. 

*“O GRUB*

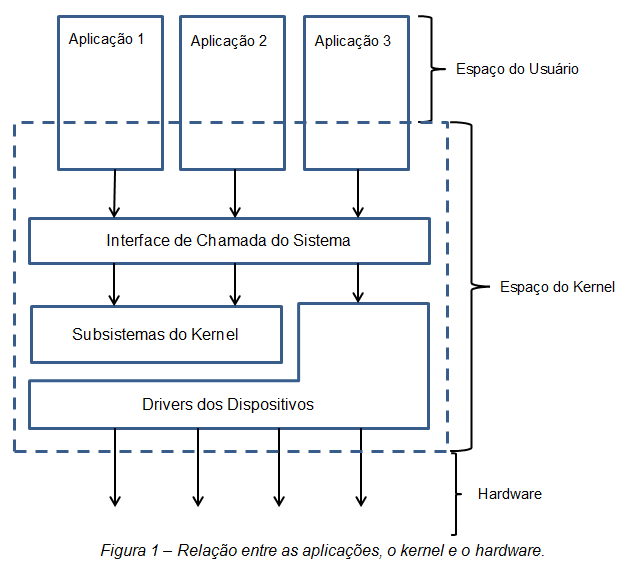
*Em computação, GNU GRUB (ou apenas GRUB) é um multi-carregador de um sistema operacional (multi boot boot-loader) criado pelo projeto GNU. É utilizado para que o usuário possa escolher ao iniciar a máquina, um sistema operacional (SO) dentre dois ou mais sistemas instalados. Em termos técnicos ele é um programa que pode carregar qualquer arquivo executável com um cabeçalho multi boot nos seus primeiros 8 kB. Este cabeçalho consiste numa sequência de bits com: 32 bits de um "número mágico", 32 bits de flags mais 32 bits de um outro número mágico seguidos pela imagem do arquivo executável.   
  
O GNU GRUB foi desenvolvido a partir de um pacote chamado GRand Unified Bootloader, de onde deriva o acrônimo GRUB. O sistema operacional GNU utiliza o GRUB, assim como muitas distribuições*[*GNU/Linux*](https://www.vivaolinux.com.br/linux/)*.”*  
  
Enquanto os "carregadores" tradicionais mantém uma tabela de blocos no disco rígido, o GRUB pode rastrear o sistema de arquivos. Na sua versão do ano 2005 ele suportava os seguintes sistemas de arquivos:

* ext2/ext3/ext4;
* JFS da IBM;
* Iso9660;
* o sistema de arquivos do Minix;
* NTFS;
* ReiserFS;
* SGI's XFS;
* UFS/UFS2;
* VFAT, assim como os sistemas FAT16 e FAT32 utilizado pelo Microsoft Windows;
* o sistema nativo VSTa.

Atualmente em desenvolvimento, o GRUB 2 substituiu o GRUB que passou a chamar-se GRUB Legacy (ou, em português, GRUB legado). O desenvolvimento do GRUB 2 pretende fundir as fontes com o projeto PUPA para criar a próxima geração do GNU GRUB.   
  
O GRUB suporta a adição de 14 cores em substituição ao fundo de tela negro. Algumas distribuições que incluem o GRUB utilizam fundos de tela customizados. Os usuários também podem incluir as suas próprias customizações.   
  
O processo de carregamento do sistema operacional, usando o GRUB:   
  
1. A BIOS busca um dispositivo que faça o carregamento do SO (normalmente um disco rígido) e move o controle para o MBR. O MBR é situado nos 512 primeiros bytes do disco;   
  
2. O MBR contém o estágio 1 do GRUB. Dado o pequeno tamanho deste estágio, ele apenas carrega o próximo estágio do GRUB (que pode residir em qualquer locação do disco). O estágio 1 pode carregar o estágio 1.5 ou o estágio 2 diretamente;   
  
3. O estágio 1.5 é localizado nos 30 primeiros Kb do disco imediatamente após o MBR. O estágio 1.5 carrega o estágio 2;   
  
4. O estágio 2 recebe o controle, e mostra ao usuário o menu com as opções de sistemas operacionais instalados no sistema;   
  
5. O GRUB carrega na memória o núcleo (kernel) do SO escolhido (ou o padrão) e passa o controle a este núcleo. (Para sistemas operacionais que o GRUB não suporta totalmente, o controle é passado para outro carregador que continua o processo até carregar o núcleo em memória).

**Kernel do Linux**

O Linux Kernel é o software que fornece os serviços básicos para todas as outras partes do sistema, gerencia o hardware e distribui os recursos do sistema



**Kernel Monolítico** :: Monolítico vem da palavra monólito, formado de uma só pedra, que forma um só bloco.   
  
O kernel monolítico é aquele que contém todos os módulos e subsistemas em um único executável binário. O kernel monolítico tem como característica, permitir que as funções essenciais sejam executadas através do *kernel space* (espaço do kernel).   
  
Isto é possível através do uso de módulos. O que significa que um módulo, apesar de não estar no mesmo código do kernel, é executado no espaço de memória do kernel fazendo com que o kernel continue sendo único e centralizado.   
  
Geralmente, um kernel monolítico tem melhor desempenho do que outras arquiteturas de kernel, porém, a manutenção da sua programação é mais demorada, pois o kernel deve ser recompilado e substituído por completo, caso queira implementá-lo.   
  
Exemplos: *Linux, Unix, BSD, FreeBSD, FreeDOS*, MS-DOS.

**Sistema X - System X (Parte Gráfica)**

**Conclusão**

**Referências:**

<http://www.lancaster.ac.uk/undergrad/stanev/References/Professional%20Linux%20Kernel%20Architecture.pdf>

<http://www.tecmundo.com.br/aumentar-desempenho/11266-como-funciona-o-boot-de-um-computador.htm>

<http://www.tidia-ae.usp.br/access/content/group/9a9cffb4-f772-4089-a9ce-bbe21d5a0908/aula01.pdf>

<http://homepages.dcc.ufmg.br/~cesarfmc/classes/manut2/TeoricaSetup.pdf>

<http://www.hardware.com.br/tutoriais/entendendo-como-linux-funciona/processo-boot-arquivos-inicializacao.html>

<https://www.vivaolinux.com.br/artigo/Boot-Linux-o-que-acontece-quando-ligamos-o-computador?pagina=3>

<https://www.vivaolinux.com.br/artigo/O-Kernel-Linux>

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

<https://www.vivaolinux.com.br/artigo/O-programa-INIT?pagina=1>

<https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/3_Processes.html>

Imagens:

Figura 1 : <http://cdn.computerhope.com/bios.jpg>

Figura 2 : <http://cdn2.pcadvisor.co.uk/cmsdata/features/3491045/Bios-which-key.jpg>

Figura 3 : <https://img.vivaolinux.com.br/imagens/artigos/comunidade/Figura-1.png>