

## **Bootloader**

### **Projeto 1 - Bootloader IF677 - Infra-Estrutura de Software Centro de Informática - UFPE**

#### **O que é?**

- É o primeiro programa a ser carregado e executado
- Possui a responsabilidade de organizar e levantar o sistema operacional

#### **Modo de operação**

- É executado em modo real
- Possui limitação de tamanho (512 bytes)
- É baseado em interrupções da BIOS
- Pode ser dividido em múltiplas etapas (possibilitando utilizar mais que 512 bytes)

#### **Master Boot Record (MBR)**

- Possui um tamanho FIXO de 512 bytes
- É carregado pela BIOS na memória
- Contém o bootloader
- Está localizado em um drive nas seguintes posições: (512bytes)
  - Cabeça 0;
  - Trilha 0;
  - Setor 1.

### **O que está acontecendo ao ligar meu computador?**

**Ligar o PC:** O botão de iniciar é apertado e a energia é liberada.

**BIOS:** Realiza o POST(Power On Self Test), uma checagem nos componentes do sistema (E/S,RAM, CPU, etc) através de sinais. Se estiver tudo ok (sem curto circuito, sem falha do disco, etc), o processo da BIOS continua. Caso não, dá uns bips ou mensagem de erro para evitar que algum componente seja danificado.

A BIOS também carrega o MBR na posição 0x7C00 da memória e continua a execução a partir deste ponto.

**Bootloader:** Carrega o sistema operacional na memória

### **O projeto**

O projeto consiste em desenvolver um bootloader de 2 estágios, que, ao final, chame um kernel feito em assembly, tudo desenvolvido no modo real.

#### **Primeiro Estágio:**

A BIOS carrega o primeiro setor do disco (se ele tiver a assinatura de boot 0xAA55 no final do setor), no endereço linear de memória 0x7c00, e pula para esse endereço. Como um setor no disco geralmente possui 512 bytes, muitas vezes divide-se o bootloader em mais de um estágio, de modo que a BIOS carregue o primeiro estágio (512 bytes), e este carregue um estágio subsequente.

#### **Segundo Estágio:**

O segundo estágio de um bootloader normalmente carrega algumas estruturas que serão úteis ao kernel na memória. Em seguida, carrega o kernel na memória, configura um ambiente adequado para o núcleo, e passa o controle para este.

Sua equipe deve desenvolver uma aplicação, modificando o arquivo kernel.asm do projeto (veja a indicação no arquivo sobre o início do código em start:). O projeto é livre, mas a equipe deverá exercitar a criatividade (ex: criar um jogo ). Qualquer trabalho extra que o grupo faça, será levado em consideração na hora da avaliação.

**O que será cobrado:**

Entender e saber explicar todo o processo de boot  
Entender o funcionamento do código assembly  
Um kernel bem implementado ( e criativo).