

Sistemas Operacionais – 2024/1

Primeiro Trabalho - Relatório

buildimage: Leitor de Arquivos ELF

Data de Entrega: 14 de abril de 2024

Alunos: Camila Cardoso Cáceres, Luiz Gustavo S. S. Junqueira

Relatório de Implementação

O objetivo do programa desenvolvido é de, a partir de dois arquivos ELF – bootblock e kernel – gerar um arquivo de imagem de um sistema operacional, que pode ser identificado e iniciado pela BIOS.

Para isso, no código, primeiramente criamos o arquivo de imagem, onde serão escritos os programas do bootblock e do kernel, em seguida os abrimos e verificamos se realmente são arquivos do tipo ELF pelo ELF Header e lemos o Program Header, que contém as informações do programa.

Com isso, na função *write_bootblock* primeiramente o programa do bootblock é lido. A partir dos campos *p_offset* e *p_filesz* do Program Header é possível identificar a posição e tamanho do programa do bootblock dentro do arquivo ELF, o qual é escrito no primeiro setor da imagem (512 bytes).

Após escrever o bootblock, colocamos a assinatura no final do primeiro setor para que o bootloader identifique que aquele é um arquivo de imagem de sistema. Então, o mesmo processo de escrita do bootblock se repete para o kernel, na função *write_kernel*, levando em consideração que o kernel possui 1 ou mais setores.

Depois de escrever os programas, é feita a escrita do número de setores do kernel no segundo byte do arquivo de imagem, na função *record_kernel_sectors*, para informar o bootloader.

Além dessas funcionalidades, temos a função *extended_opt*, que escreve na saída as informações sobre os arquivos bootblock e kernel. Também temos a função *count_kernel_sectors* que calcula o número de setores que o programa do kernel ocupa, garantindo que sempre ocupe um número inteiro de setores e seja arredondado para cima, ou seja, 513 bytes seriam 2 setores.

Do código fornecido pelo professor, removemos o parâmetro ELF Header de algumas funções visto que não era utilizado para o funcionamento do programa.