

Sistemas Operacionais – 2024/1
Primeiro Trabalho
buildimage: Leitor de Arquivos ELF
Data de Entrega: 7 de abril de 2024

Descrição do Trabalho

Um sistema operacional (SO) para iniciar precisa de um arquivo de imagem que pode ser carregado diretamente na memória do computador pelo programa de boot (bootloader), que por sua vez deve ser carregado pelo BIOS (*Basic Input Output System*). Quando você compila um programa, usando gcc em Linux por exemplo, o compilador gera um arquivo em um formato com informações úteis para o sistema operacional, conhecido como ELF (*Executable and Linkable Format*). Esse arquivo contém o código e os dados do programa e informações que dizem ao SO como carregar o programa em memória e quais recursos (*e.g.*, bibliotecas dinâmicas) são necessários. Porém, no início do processo de boot, você não possui um SO para carregar arquivos executáveis. Assim, você precisará extrair um programa de boot e um kernel (fornecidos como parte do trabalho) de seus arquivos ELF correspondentes e gerar um arquivo de imagem como se eles fossem carregados em memória.

Para criar o arquivo **buildimage**, você terá de se familiarizar com os seguintes atributos do formato de arquivo ELF:

- Cabeçalhos ELF e seus campos `e_phnum` e `e_phoff`.
- Cabeçalhos de programa e seus campos `p_offset` e `p_filesz`. Note que as propriedades `p_filesz` e `p_memsz` de um segmento em um executável podem diferir.

Você deve ler o documento sobre o formato de arquivo ELF disponibilizado no AVA, pois ele contém explicações detalhadas dos atributos que você precisará neste projeto.

Você também deve implementar a opção `-extended`. Essa opção imprime informações para depuração da imagem. Em particular, quando essa opção é fornecida, **buildimage** deve mostrar o número de setores usados pela imagem, os números dos setores no disco que conterão os diferentes executáveis carregados por **buildimage**, os segmentos especificados nos cabeçalhos ELF para cada arquivo executável carregado e o tamanho do kernel em setores para você ter uma ideia de como o carregador precisará funcionar para carregar todo o SO em memória.

Implementação

Para implementar **buildimage**, você terá de implementar as seguintes funções em C além de complementar `main()`:

- `read_exec_file()`: lê um arquivo executável no formato ELF.
- `write_bootblock()`: escreve o bloco de boot no arquivo imagem.
- `write_kernel()`: escreve o kernel para o arquivo imagem.
- `count_kernel_sectors()`: conta o número de setores do kernel.
- `record_kernel_sectors()`: informa ao carregador de boot quantos setores o kernel tem.
- `extended_opt()`: imprime as informações quando a opção `-extended` é fornecida.

Teste

Um executável com uma implementação equivalente de **buildimage** é fornecido no código inicial do projeto para fins de comparação. Você deve testar se a imagem que você gerou está correta executando-a no emulador **bochs**.

Entrega do Trabalho

O trabalho pode ser feito em grupo de no máximo dois alunos e deve ser entregue até o dia 7 de abril de 2024. Além do código fonte devidamente comentado, o grupo deve entregar um breve relatório descrevendo o trabalho. Neste relatório, o grupo deve incluir uma breve introdução, decisões de implementação, funcionalidades não implementadas, problemas enfrentados na implementação, etc. O relatório deve ser entregue em um arquivo PDF. Tanto o relatório quanto os arquivos fontes da implementação devem ser colocados em um arquivo ZIP e enviados via AVA. Arquivos em outros formatos, como RAR, não serão considerados e o trabalho não será avaliado. Inclua em seu arquivo ZIP apenas os arquivos fontes e Makefile, ou seja, não inclua arquivos compilados (.o).

Avaliação

Além da correção do programa, o professor e/ou assistente de ensino poderão fazer entrevistas com os grupos. Durante a entrevista, o grupo deverá explicar o funcionamento do programa e responder perguntas relativas ao projeto.