

Na linha 1 ocorre uma chamada ao sistema operacional (execve) a fim de iniciar o programa. Na segunda linha, ocorre o carregamento do cache para guardar o programa a ser executado. Há permissão apenas para leitura. Depois, a cache é fechada, após executar algumas operações de mapeamento.

O próximo open é referente ao arquivo que estava localizado na pasta /tmp/file1 com permissões de leitura e escrita. A escrita é realizada na linha posterior e retorna 42, que é o tamanho da alocação de memória que é feita para o buffer do programa.

Depois, o programa “dorme” por 30 segundos e retorna sem erros.

O segundo trecho do código difere apenas no arquivo e na localização dos endereços de memória, mas o fluxo é idêntico. Para mais detalhes, consultar os traces nos arquivos anexados.

Lab 1 - QT 3

Que estrutura de dados no kernel linux 4.8 armazena os file descriptors de um processo? Explique cada um dos campos dessa estrutura

A partir da linha 24, vemos uma estrutura chamada fdtable que é o struct que armazena os file descriptors de um processo. Ela atua em conjunto com o files_struct, que inicia na linha 46.

A fdtable atua apenas como um struct de flags para verificar a permissão de cada operação. Além disso, existem funções logo abaixo que definem o status dessas flags.

As funções *bool close_on_exec* e *bool fd_is_open* definem o status das flags *close_on_exec e *open_fds;

Referência: <http://lxr.free-electrons.com/source/include/linux/fdtable.h#L24>