MAC0422 - EP4

Daniel Martinez - 10297709
Pedro Paulo Bambace - 10297668

MAC0422 - EP4

Integrantes

Arquivos criados/modificados

- /usr/src/include/minix/const.h
- /usr/include/minix/const.h
- /usr/src/include/fcntl.h
- /usr/include/fcntl.h
- /usr/src/lib/posix/_open_tmp.c
- /usr/src/lib/syscall/_open_tmp.s
- /usr/src/lib/syscall/Makefile.in
- /usr/src/lib/posix/Makefile.in
- /usr/include/minix/callnr.h
- /usr/src/include/minix/callnr.h
- /usr/src/servers/fs/open.c
- /usr/src/servers/fs/proto.h
- /usr/src/servers/fs/table.c
- /usr/src/servers/fs/file.h
- /usr/src/servers/fs/path.c
- /usr/src/servers/fs/proto.h
- /root/banana.c
- /root/uva

Detalhes de implementação

Implementação do `open_tmp()`

Primeiramente, definimos nosso valor de I_TEMPORARY como 0030000, pois tivemos dificuldade de fazer o número sugerido pelo enunciado se comportar da maneira correta. Em seguida, criamos a chamada de biblioteca `open_tmp()` que realiza a chamada de sistema OPENTMP no *File System*. Essa chamada de sistema cria um novo arquivo temporário que é deletado com o final da execução do programa. Para isso, criamos em `open.c` uma cópia modificada da função `common_open()` (chamada `common_open_temp()`)que salva os *i-nodes* com esse novo modo (I_TEMPORARY), além de salvar o *i-node* do diretório desse arquivo em um novo campo da tabela `filp`. Modificamos também a função `do_close()` para que quando o *i-node* seja de um arquivo temporário e não houverem mais nenhuma referência a ele, ele seja deletado. Para isso, criamos uma cópia também da função `search_dir()` (chamada `search_dir_inode()`) que busca (e no nosso caso também deleta) os filhos de um diretório baseado apenas em seu *i-node*.

Build e testes

O mesmo arquivo de build dos EPs anteriores foi utilizado. Basta executar ./b para compilar as mudanças do sistema.