

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
DEPARTAMENTO DE INFORMÁTICA APLICADA**

Trabalho Prático II - Implementação da Biblioteca *libt2fs*

1 – Nomes:

Alexandre Leuck - 220493, Gustavo Dambros - 240490 e Mariane Giambastini - 230166

2 - Resumo executivo das funções:

Função	Status
identify2	OK
create2	OK
delete2	OK
open2	OK
close2	OK
read2	OK
write2	OK
seek2	OK
mkdir2	OK
rmdir2	OK
opendir2	OK
readdir2	OK
closedir2	OK
chdir2	OK
getcwd2	OK

3 – Testes e resultados esperados:

Todos os testes foram realizados utilizando o disco disco_32_4_128.dsk.

Identify2 – Foi chamada a função para verificar se os nomes eram passados corretamente. Funciona corretamente.

testes/identify_get_.c

create2 – Foram criados 80 arquivos num único diretório pai e verificado se estava sendo alocado novos clusters para a alocação dos arquivos. Também foi testado a criação de arquivos no diretório raiz. A função funcionou perfeitamente.

testes/teste_80_arquivos.c

delete2 – Foi criado um arquivo de 3000 bytes em um diretório, utilizando o delete, verificado-se que o arquivo foi deletado do diretório pai e os endereços na FAT foram zerados. A função funcionou perfeitamente.

testes/delete.c

open2 – Mesmo teste da **write2**.

close2 - Mesmo teste da **write2**.

read2 - Mesmo teste da **write2**.

write2 - Foi criado um arquivo, tanto na raiz quanto em um diretório, e foi escrito nesse arquivo um texto de aproximadamente 1024 bytes, que ocupou dois clusters, para verificar se o arquivo alocava mais de um cluster. Para isso utilizou-se o **open2**, **write2** e **read2**, para que fosse possível printar o arquivo, também foi verificado o funcionamento da função **close2**. Todas as funções funcionaram corretamente.

testes/mkdir_chdir_create_write_open_read_close.c

seek2 – Cria um arquivo e escreve no arquivo. Faz um seek nos dez primeiros bytes , seguido de uma leitura e verificou-se que os dez primeiros bytes do arquivo não foram lidos. Funciona corretamente.

testes/seek.c

mkdir2 – Foram criados 80 diretórios num único diretório pai e verificado se estava sendo alocado novos clusters para a alocação dos novos diretórios. Também foi testado a criação de diretórios no diretório raiz. Funciona corretamente.

testes/teste_cria_80_dir.c

rmdir2 - Mesmo teste da **readdir2**.

opendir2 - Mesmo teste da **readdir2**.

readdir2 – Foi criado um diretório na raiz e dentro dele foram criados alguns arquivos. Utilizou-se a função readdir2 para ver os arquivos existentes no diretório. Foi aberto um diretório já existente utilizando o **opendir2** e também foram vistos os arquivos existentes nesse diretório. Por fim, usou-se a função **closedir2** para fechar os diretórios e rmdir2 para remover o diretório que foi criado na raiz. Todas as funções funcionaram corretamente.

testes/readdir.c

closedir2 – Mesmo teste da **readdir2**.

chdir2 – Foi chamada a função passando o parametro “teste” um diretório que já existia no disco. Funciona corretamente.

testes/identify_get_.c

getcwd2 – Foi chamada a função para verificar se o caminho do diretório era passado corretamente. Funciona corretamente.

testes/identify_get_.c

4 - Principais dificuldades encontradas:

A principal dificuldade foi conseguir finalizar o trabalho no prazo de entrega.