FACULDADE DE CIÊNCIAS E TECNOLOGIAS DA UNIVERSIDADE NOVA DE LISBOA

Programming Assignment 2

Fundamentos de Sistemas de Operação

David Gago nº41710 P5

João Almeida nº42009 P6

MIEI

Prof. M. Cecília Gomes

Ano Lectivo 2013/2014

Descrição geral:

O projecto consiste em implementar um sistema de ficheiros semelhante ao utilizado em UNIX/Linux. O sistema de ficheiros está contido num disco de capacidade variável dado pelo utilizador.

Existem várias **funções** disponíveis ao utilizador:

fs_format - cria um novo sistema de ficheiros limpando a informação, alocando espaço para a tabela de inodes (10%). Todas as entradas são marcadas como livres.Primeiramente, há um verificação se o magic number que está no Super Block está a 0.
Se estiver o magic number toma o valor da constante do nosso sistema de ficheiros. O numero de blocos de inodes é 10% do numero de blocos existentes e esta informação é colocada no Super Block e o disco é escrito com a informação de data. Os inodes são todos colocados a válido.

fs_debug - Imprime a informação do disco que está a ser usado.
Começa por ler o super bloco do disco comparando o magic number à constante global para sabes se é válido e imprimindo o resto dos dados do bloco.
De seguida percorre todos os blocos de i-nodes e verifica se estes são válidos, caso sejam imprime o tamanho e os blocos associados a este.

fs_mount - Verifica se está algum disco montado.
Se não estiver monta o disco, ou seja, copia o super bloco para RAM, inicializa a tabela de blocos livres/ocupados e preenche-a percorrendo todos os inodes do disco e marcando os blocos no vector de directs como usados.
São usados três ciclos, um para percorrer os blocos de inodes, outro para percorrer os inodes de cada bloco e outro para percorrer o vector de directs caso o i-node seja válido.

fs_create - Cria um novo i-node correspondente a um novo ficheiro(tamanho 0).
Primeiro faz uma verificação e retorna -1 caso o magic number do Super Block seja diferente da constante do magic number do nosso sistema de ficheiros.
Depois faz um ciclo que percorre os blocos de inodes. A cada ciclo, lê o conteudo do bloco de inodes e entra noutro ciclo em que percorre os inodes desse bloco. Se o inode for válido, guarda esse indice e marca o como usado. Diz que o seu tamanho do ficheiro é 0. No final se o indice tiver um valor maior que 0, escrever o inode no bloco correspondente.

fs_delete - Remove o inode do ficheiro. Actualiza o mapa de blocos livres/ocupados, depois liberta o inode. Primeiro faz uma verificação e retorna 0 caso o magic number do Super Block seja diferente da constante do magic number do nosso sistema de ficheiros.
De seguida carrega o inode em memoria. Coloca o inode como válido, actualiza o estado do inode. De seguida, vai se à entrada do mapa de blocos na entrada correspondente ao ficheiro e coloca-se a entrada a livre. Em caso de sucesso retorna 1, caso contrário retorna 0.

fs_getsize - Verifica se o disco está montado. Se estiver, e o índice do i-node dado seja válido, carrega o i-node. Caso este seja válido, devolve o número de bytes associado a este, caso contrário devolve erro.

fs_read - Lê a informação de um inode válido.
Começa por verificar se o disco está montado e se o inode existe, caso falhe alguma das verificações retorna um valor de erro. De seguida carrega o inode para a memória, verifica e se está a tentar ler depois do final do ficheiro. Depois entra num ciclo que lê os dados para o buffer em blocos pertencentes ao vector de directs a partir do offset dado. Retorna o numero de bytes transferidos.

fs_write - Este método tenta escrever length caracteres do buffer dado a partir de um offset no i-node dado.
Começa por verificar se o disco está montado e se o inode existe, caso falhe alguma das verificações retorna um valor de erro.
De seguida carrega o i-node para memória, verifica se há um ficheiro criado nesse i-node e se está a tentar escrever depois do final do ficheiro, se falhar retorna erro também.
Entra num ciclo em que vai escrever os dados em blocos pertencentes ao vector de directs ou tenta alocar novos blocos para o ficheiro, se não conseguir alocar novo bloco sai do ciclo sem ter escrito todos os dados. Depois de escrever todos os dados sai do ciclo e liberta todos os blocos que estavam anteriormente a ser usados pelo i-node e deixaram de o ser.
Salva o inode com os dados modificados e retorna o número de bytes escritos no ficheiro.

Limitações
Não encontrámos limitações no nosso programa para além das impostas pelo enunciado.

Conclusão
Este trabalho foi concluído com sucesso e serviu para consolidar o nosso conhecimento sobre a maneira como os sistemas de ficheiros funcionam e as dificuldades da implementação destes.
Serviu também para melhorarmos as nossas capacidades de programação em C.