



1. Introdução

A segunda parte do projecto de SO aborda o tema de sistemas de ficheiros.

2. Entrega e Avaliação

A entrega final consiste na submissão da Parte II. Os detalhes sobre a entrega serão anunciados na página cadeira.

A informação sobre a entrega e avaliação aqui descrita está sujeita a alterações, que serão afixadas na página da cadeira.

Entrega da Parte II: 12 Dez às 23:59

3. Requisitos da Parte II

Objectivo

Nesta parte II do projecto de Sistemas Operativos, pretende-se melhorar o desenho e implementação de um sistema de ficheiros que se caracteriza pela presença de evidentes limitações que afectam a sua utilidade:

- a existência de um único directório para o armazenamento de ficheiros;
- a incapacidade de referenciar o mesmo ficheiro utilizando nomes distintos (ou seja, os chamados *hard links*).

Neste âmbito, o objectivo da parte II do projecto é estender um sistema de ficheiros já existente por forma a ultrapassar as duas limitações acima.

Contexto

O ponto de partida desta parte II do projecto de Sistemas Operativos é o sistema de ficheiros *barefs*, cujo código fonte está disponível no *site* do projecto. O *barefs* pode ser instalado em qualquer directório (*mount point*) num sistema Linux. Os utilizadores e aplicações podem gerir os ficheiros mantidos pelo *barefs* usando a API tradicional de ficheiros do Linux.

O *barefs* é um sistema de ficheiros muito simples, baseado na plataforma FUSE. Possui as seguintes características:

- O tamanho máximo dos ficheiros é de 10 blocos;
- Tamanho dos blocos tem dimensão fixa de 512 bytes. O número de blocos que perfazem o disco é definido em tempo de compilação;
- A arquitectura do sistema de ficheiros é baseada numa simplificação dos i-nodes Unix, com 10 entradas directas para blocos de dados. O tamanho de um i-node é de 64 bytes. O número máximo de i-nodes é, por omissão, de 64 (a tabela de i-nodes ocupa, por isso 8 blocos), mas este valor pode ser definido em tempo de compilação;
- Existe apenas um directório (raiz), onde todos os ficheiros residem. Ou seja, não são suportadas subdirectórios. O directório raiz é um ficheiro estruturado sob a forma de uma tabela de entradas. Cada entrada tem 16 bytes e é constituída por: nome do ficheiro e número do i-node respectivo. Os nomes de ficheiros são limitados a 14

caracteres e o número do i-node a 2 bytes. Considera-se que a tabela do directório raiz pode ocupar, no máximo 2 blocos, isto é, suporta até 64 entradas;

- Os blocos do disco estão organizados da seguinte forma:
 - Bloco 0: reservado para o bitmap de blocos livres;
 - Bloco 1: reservado para o bitmap de i-nodes livres;
 - Blocos 2-9: reservados para a tabela de i-nodes;
 - Blocos ≥ 10 : para os dados dos ficheiros e directórios.

As operações actualmente suportadas pelo sistema de ficheiros *barefs* são as seguintes:

- Criação e remoção de ficheiros no directório *root*;
- Leitura e escrita de ficheiros criados;
- Listagem do conteúdo do directório *root*.

Requisitos

Nesta parte II do projecto, pretende-se que o sistema de ficheiros *barefs* passe a suportar as seguintes operações:

- Criação e remoção de subdirectórios

A semântica deve ser equivalente à semântica Unix. Ou seja, um subdirectório pode ser criada dentro de um dado directório apenas caso não exista já um outro ficheiro/directório com o mesmo nome; caso contrário, a função de criação deve retornar erro. Por outro lado, um subdirectório pode ser eliminado apenas caso esteja vazio; caso contrário, a função de remoção deve retornar erro.

- Criação e remoção de *hard links*

Mais uma vez, a semântica deve ser equivalente à semântica Unix. Um *hard link* para um ficheiro ou directório alvo pode ser criado apenas caso o alvo exista e caso não exista já um ficheiro ou directório com o nome solicitado para o novo *hard link*. Caso ambas as condições não se verifiquem, a função retorna erro.

Quando se elimina o último *hard link* para um dado ficheiro, o ficheiro deve ser eliminado em definitivo.

Note-se que, quando um ficheiro é criado, existe um *hard link* para o ficheiro – esse *hard link* corresponde ao nome inicial do ficheiro.

4. Referências

1. Página da cadeira, secções “Aulas Teóricas” e “Aulas Práticas”.
2. Filesystem in Userspace , <http://fuse.sourceforge.net/>
3. Fuse Documentation, http://www.cs.hmc.edu/~geoff/classes/hmc.cs135.201001/homework/fuse/fuse_doc.html#create
4. Writing a FUSE Filesystem: a Tutorial, <http://www.cs.nmsu.edu/~pfeiffer/fuse-tutorial/>

Anexo – Pacote *barefs*

1. Material fornecido

O material dado consiste num pacote base que permite instalar o sistema de ficheiros *barefs* a ser melhorado. O pacote é o *barefs-student-pkg-v2.tgz* que se encontra na secção “Projecto” na página da disciplina. A Tabela 1 descreve directórias e ficheiros que constituem o referido pacote base.

Tabela 1

Directório / Ficheiro	Conteúdo
<i>barefs-student-pkg-v2/</i>	Directório principal do pacote base
<i>barefs-student-pkg-v2/Readme.txt</i>	Instruções para a montagem do <i>barefs</i>
<i>barefs-student-pkg-v2/barefs/</i>	Directório do <i>barefs</i> (contém ficheiros .c, .h e Makefile inerentes à implementação do pacote base)
<i>barefs-student-pkg-v2/barefs/barefs.c</i>	Módulo onde vão ser implementados os métodos adicionais de manipulação de ficheiros do <i>fuse</i> .
<i>barefs-student-pkg-v2/barefs/fs.c</i>	Módulo com a implementação das funções do sistema de ficheiros.
<i>barefs-student-pkg-v2/barefs/block.c</i>	Módulo com a implementação das funções de gestão de blocos do sistema de ficheiros.

2. API do Fuse (das funções a implementar)

A Tabela 2 descreve algumas funções (métodos) do módulo *barefs.c* que compõem a API do fuse, e que são as mais relevantes em termos do desenvolvimento do projecto. Nesta tabela são também indicadas, para cada uma dessas funções, as partes que carecem de implementação.

Tabela 2

Função	Descrição	Por Fazer
<i>barefs_create(..)</i>	Cria ficheiro	Criação de novo ficheiro em subdirectórios
<i>barefs_getattr(..)</i>	Obtém os atributos de ficheiros e directórios	Actualização dos atributos com o número de <i>hard links</i> para um ficheiro
<i>barefs_mkdir(..)</i>	Cria subdirectório	Criação de subdirectórios
<i>barefs_rmdir(..)</i>	Elimina subdirectório vazio	Remoção de subdirectórios
<i>barefs_link(..)</i>	Cria <i>hard link</i> para alvo já existente	Criação de <i>hard link</i> para um alvo já existente
<i>barefs_unlink(..)</i>	Remove <i>hard link</i>	Remoção de <i>hard link</i>

3. Parâmetros utilizados nas API a implementar

Os parâmetros utilizados nas interfaces das funções do FUSE a serem implementadas são os seguintes:

- ***path*** - nome completo do directório ou ficheiro sobre o qual a função age;
- ***mode*** – permite a especificação de permissões de acesso ao directório ou ficheiro;
- ***struct stat*** - permite a passagem de informação sobre os atributos (tamanho, dono, data da última actualização,...) de directórios e ficheiros de/para as funções do FUSE.

- ***struct fuse_file_info*** – especifica os modos de acesso a um ficheiro. No pacote base, é apenas utilizado a informação referente ao campo *fi* da estrutura.
- ***from*** - nome completo do ficheiro/directório relação ao qual se pretende criar um *hard link*;
- ***to*** - nome completo do *hard link* que se pretende criar.

4. Módulo de gestão do sistema de ficheiros (ficheiro *fs.c*)

O módulo *fs.c* contém um conjunto de funções de gestão do sistema de ficheiros base disponibilizado. A concretização dos requisitos do projecto, vai requerer, necessariamente, quer a implementação, quer a alteração, de algumas funções e estruturas de dados que se encontram definidas neste ficheiro (*barefs/fs.c*).