

## sfind: versão simplificada de find

### Metas de aprendizagem

Completando com sucesso o trabalho, os alunos demonstram conhecer e saber utilizar a interface programática de UNIX para:

- criar novos processos;
- fazê-los intercomunicar por sinais;
- percorrer um sistema de ficheiros e dele obter informações.

### Descrição geral

A parte de programação do trabalho consiste na escrita de uma versão (muito) simplificada do utilitário de Unix "find"<sup>1</sup> que percorre de forma recursiva um diretório, encontrando ficheiros que satisfaçam a algumas condições e sobre eles executando as ações solicitadas.

O programa aqui desenvolvido, *sfind*, deve ser capaz de reproduzir os resultados apresentados pela invocação de *find* para os mesmos argumentos.

A estrutura do programa é deixada ao critério do projetista, mas exige-se o cumprimento de certos requisitos, apresentados abaixo.

#### Exemplos de invocação com *find*:

```
shell> find ~ -name lixo.txt -print
/home/user/tmp/lixo.txt
/home/user/sbin/myscanf/tmp/lixo.txt
```

```
shell> find ~ -name lixo.txt -exec file '{}' \; NOTA: file é um utilitário de Unix
/home/user/tmp/lixo.txt: ASCII text
/home/user/sbin/myscanf/tmp/lixo.txt: HTML document text
```

```
shell> find ~ -name lixo.txt -delete
shell>
```

```
shell> find . -type l -print
./so/www
./so/material
...
```

```
shell> find / -perm 0400 -print
/bak/multimedia/figs/bear.jpg
/bak/multimedia/figs/cubs.jpg
...
```

### Requisitos funcionais

O programa deve comportar-se como o comando *find*, tipicamente encontrado nos sistemas Unix, mas apenas num subconjunto das suas funcionalidades. Assim, deve ser capaz de reconhecer e implementar as opções:

- `-name string` (string é o nome exato de um ficheiro);

---

<sup>1</sup> `man find`: `find` - search for files in a directory hierarchy...

- `-type c` (`c` é um carácter que pode ser `f` - ficheiro normal -, `d` - diretório -, `l` - ligação simbólica);
- `-perm mode` (`mode` é um valor em octal correspondente às permissões de acesso a um ficheiro, e.g 0644).

Deve também ser capaz de executar sobre os ficheiros encontrados as seguintes ações:

- `-print` (mostra no ecrã o nome dos ficheiros encontrados);
- `-delete` (apaga os ficheiros);
- `-exec command ;` (executa `command` - nota: os ficheiros encontrados são representados por `{ }` em `command`).

Além disto (o que não acontece com `find`!), pretende-se que, ao se carregar em `CTRL+C`, todo o programa interrompa a operação e coloque ao utilizador a pergunta: «Are you sure you want to terminate (Y/N) ?». Se a resposta for `Y` (ou `y`), o programa termina sem mais delongas; se for `N` (ou `n`), o programa prossegue como se nada tivesse acontecido.

## Requisitos arquiteturais

Apesar de, como foi dito, a estrutura do programa ser deixada a cargo de quem o vai escrever, há alguns requisitos arquiteturais que são exigidos. O programa deve:

- criar um processo por cada diretório a analisar;
- comportar-se de forma recursiva: o que o primeiro processo fizer no diretório inicial, será repetido pelos processos descendentes, mas sobre os diretórios que lhes forem atribuídos;
- percorrer cada diretório distinguindo os tipos de ficheiro nele contidos:
  - a um ficheiro normal encontrado, serão aplicados os testes e as ações estipuladas na linha de comando;
  - a um (sub-)diretório, corresponderá a criação de um processo descendente idêntico, que repetirá o aqui descrito a esse (sub-)diretório.

## Produto final

O trabalho total consiste na produção de um ficheiro compacto, que inclui o código-fonte com o programa desenvolvido e um *makefile* preparado para facilitar a geração do executável. O compacto, identificado com um nome do tipo `TxGyy.tar.gz`, onde `x` e `yy` são o número da turma e do grupo, respetivamente, deve ser submetido no Moodle.

## Avaliação

**NOTA:** *este projeto não será classificado; no entanto, indicam-se as condições que seriam usadas na avaliação.*

A avaliação será efetuada através de testes simples de execução, em que os resultados produzidos pelo programa `sfind` desenvolvido serão comparados com os resultados produzidos por `find` em condições idênticas.

O funcionamento do requisito relativo à reação do programa relativamente à sua interrupção inesperada através da ativação de `CTRL+C`, também será testado.

Também será avaliada a estrutura e a qualidade geral do código-fonte.