

Implementar un programa *lsinfo* que permita obter información relativa a un ficheiro en particular, pero tamén sobre as entradas existentes nun directorio dado que cumplan un patrón determinado, ou simplemente relativa a un directorio.

Cando ao programa se lle pase como parámetro un ficheiro regular (*\$lsinfo fich*), amosará simplemente a mesma información que se obtería co comando *\$ls fich*. A opción *-l* fará que se amose a mesma info que sería obtida con *ls -lisa fich*.

Cando o programa se execute sobre un directorio admitirá as opcións seguintes:

- l* : Fará que se amose a mesma información que *\$ls -lisa dir*
- r* : Implica un procesamento recursivo do directorio en cuestión
- h* : Fai que se teñan en conta tamén as entradas ocultas (opción por defecto é non procesalas).
- c* : Fai que para cada directorio se amose información resumo. En concreto amosarase o número total de ficheiros regulares que contén, así como o total de bytes utilizados polos ficheiros regulares contidos no directorio en cuestión. Se a opción “recursiva” está activa amosarase información de resumo para cada directorio procesado.
- s* : Fai que se ordeen por orde alfabética as entradas existentes no directorio en cuestión. Esta opción non se terá en conta no caso de estar activa a opción “recursiva”.

Nota: Cando se amose o contido dun directorio deberá ser doado de distinguir cal é o directorio do que se listan os seus elementos en cada momento (opción recursiva).

Por último, tamén se pode lanzar o listado sobre unha expresión do estilo *\*.ext* ou *nome.\**. Neste caso o listado terá en conta únicamente aquelas entradas de directorio que emparellen coa expresión introducida. Asumirase un formato de saída como no caso de *\$ls -lisa dir*, e haberá que ter en conta as mesmas opcións que no caso anterior (*-l -r -h -c -s*).

Deste xeito a sintaxe do programa *lsinfo* será a seguinte:

```
$lsinfo fich/dir/expr [-l][-r][-h][-c][-s]
```

### Exemplos de execución:

#### **\$ lsinfo -l wca.tar.gz**

```
164153 20 -rw-r--r-- 1 pepe pepe 19736 2008-04-17 18:45 showDB.zip
```

#### **\$lsinfo -l -h .**

```
163876 4 drwxr-xr-x 3 pepe pepe 4096 2008-10-09 16:48 .
163841 4 drwxr-xr-x 9 pepe pepe 4096 2008-05-17 00:06 ..
180227 16 -rwxr----- 1 pepe pepe 15921 2008-04-17 18:45 showDB.zip
164154 24 -rwxr-xr-x 1 pepe pepe 20946 2008-10-01 06:20 BUILDFACADE
164141 4 -rwxr--r-- 1 pepe pepe 922 2008-04-17 18:44 Makefile
163901 4 drwxr-x--- 3 pepe pepe 4096 2008-04-17 18:42 src
164153 20 -rw-r--r-- 1 pepe pepe 19736 2008-04-17 18:45 csa.tar.gz
164153 0 lrwxrwxrwx 1 pepe pepe 11 2007-07-24 18:26 tmp -> /tmp/tmppepe
```

\$lsinfo -l -h -c /home/pepe/tmp

```
##directorio /home/pepe/tmp:
163876 4 drwxr-xr-x 3 pepe pepe 4096 2008-10-09 16:48 .
163841 4 drwxr-xr-x 9 pepe pepe 4096 2008-05-17 00:06 ..
180227 16 -rwxr----- 1 pepe pepe 15921 2008-04-17 18:45 showDB.zip
164154 24 -rwxr-xr-x 1 pepe pepe 20946 2008-10-01 06:20 BUILDFACADE
164141 4 -rwxr--r-- 1 pepe pepe 922 2008-04-17 18:44 Makefile
163901 4 drwxr-x--- 3 pepe pepe 4096 2008-04-17 18:42 src
164153 20 -rw-r--r-- 1 pepe pepe 19736 2008-04-17 18:45 csa.tar.gz
164153 0 lrwxrwxrwx 1 pepe pepe 11 2007-07-24 18:26 tmp -> /tmp/tmppepe

@@@ total = 57525 bytes, en 5 ficheiros regulares
```

## Comentarios

- A información de cada ficheiro obtense con *stat* ou *lstat*.
- As funcións *opendir*, *readdir* e *closedir* poden ser utilizadas para obter o contido dun directorio dado.
- Dado un identificador de usuario/grupo, hai funcións que nos permitirán obter o seu nome ou ben o nome do grupo. Entre elas é convinte revisar *getpwent*, *getpwuid*, *getgrent* e *getgruid*.
- Como se pode ver no exemplo, os permisos amosaranse no formato *rwrxrwxrwx*.
- O lugar apuntado por un *link* simbólico pódese coñecer coa función *realpath*. Recordemos que (*ln -s* permite crear links simbólicos).
- As datas poden convertirse a formato cadena con *ctime* (cómpre revisar cal é o último carácter que devolve *ctime* para evitar problemas de formateado!!)

## Modo de entrega

As prácticas entregaranse por email ([antonio.fari.so@gmail.com](mailto:antonio.fari.so@gmail.com) ou [jcasanova@udc.es](mailto:jcasanova@udc.es) antes de proceder á súa defensa. Deberase enviar un ficheiro comprimido *tar.gz* contendo: a) código fonte e b) ficheiro *Makefile* (o profesor compilará tecleando *\$make*). O asunto da mensaxe deberá ser o seguinte:

[SO1P1]:[Nome 1ºIntegrante Grupo][Nome 2ºIntegrante Grupo] // de ser só 1 alumno → 1 só nome.

O nome do ficheiro comprimido debe ter os códigos de usuario concatenados: **infxxx00-infyyy00.tar.gz**

A práctica será entregada e defendida ante o profesor na aula de prácticas. Todos os membros dun grupo deberán estar presentes para a entrega, de xeito que o profesor poida revisar o seu correcto funcionamento, así como realizar comentarios/cuestións aos integrantes do grupo, así como pedir a realización de cambios no código que se poidan considerar pertinentes.

As prácticas entregadas e que posteriormente non se saiban defender con solvencia “non serán ben vistas” e (dependendo das circunstancias) poderán implicar un “non apto” para todos os membros do grupo.

O programa debe:

- Compilar correctamente: usarase *gcc* coa opción *-Wall*, e non debe conter erros.
- Executar correctamente:
  - Seguirá as especificacións marcadas.
  - Funcionar correctamente. O alumno deber ter en conta os valores que devolven todas as funcións que utilice (OLLO aos valores devoltos polas funcións utilizadas!!).
  - Non debe conter *memory-leaks* (o profesor usará *valgrind* para chequear este aspecto).

A data límite de entrega será o: Venres 28 de novembro 2007

**Detalles a recordar**

- O valor máximo das prácticas é de 1.5 puntos. Cómpre que todas as prácticas reciban a calificación de “apta”.
- Non todas as prácticas teñen o mesmo valor. En principio esta primeira práctica ten un valor de 0.6 puntos. Sen embargo, reservámonos a posibilidade de modificar esta valoración (+/-0.15)