SISTEMAS OPERATIVOS 1. Enxeñería Informática. Curso 2009-2010

Práctica 1: Diskusage.

Tratase de implementar un programa **diskusage** que permita obter a información do espazo empregado dun ficheiro en particular, pero tamén a información das entradas existentes nun directorio, ou simplemente relativa a un directorio.

Como exemplo para ver como funcionaría o programa supoñemos que temos a seguinte estructura de ficheiros e directorios:

```
.
|-- carpeta
|   |-- fichero3.txt
|   |-- fichero4.txt
|   `-- subcarpeta
|       `-- fichero5.txt
|-- carpeta2
|   `-- fichero6.txt
|-- fichero1.txt
`-- fichero2.txt
`-- fichero3.txt -> fichero2.txt   (Enlace Simbólico)
```

Cando ao programa se lle pase como parámetro un ficheiro regular ($diskusage ficheiro), amosará a mesma información que se obtería co comando **$du fich**.

$ diskusage fichero1.txt
4 fichero1.txt

Cando o programa se execute sobre un directorio, por defecto amosará recursivamente a información de ocupación do directorio así como dos subdirectorios existentes no directorio, e así recursivamente.

$diskusage carpeta
8 carpeta/subcarpeta
20 carpeta

Se o programa se executa sen parámetros suporase que se executa sobre o directorio actual “.”

$diskusage
334224 ./carpeta2
8 ./carpeta/subcarpeta
20 ./carpeta
334256 .

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

- **-a** : Fai que se amose a información sobre os arquivos que se atopan nos directorios.
- **-b** : Amosa a información en bytes en lugar de como número de bloques de disco ocupados.
- **-s** : Amosa a información resumida sobre os directorio/os (non inclué a información detallada de

cada un dos subdirectorios).
•**-o** : Fará que a información de ocupación dos directorios se amose de forma ordeada por cada nivel de directorio: Mostraranse en primeiro lugar os directorios que ocupan unha cantidade menor de espazo. Tódolos subdirectorios (e tamén arquivos se está activado -a) amosaranse de forma secuencial xunto coa información dos seus contidos de forma recursiva. Nota: o número de entradas máximo que pode haber un directorio non é coñecido de antemán.
•**-u** : Fará que se amose un resumo da información de ocupación, agrupando por usuarios do sistema operativo. Isto é, dado un directorio que pode conter ficheiros/directorios pertencentes a diferentes usuarios, amósanse os datos para cada usuario. (ver exemplo de execución máis adiante).
•**-h** : Amósase a axuda para a execución do comando así como a información de para que sirve cada un dos parámetros. O resto dos parametros ignorarase. O resultado será "similar" ao obtido co comando $du –help.

Os parametros **-s** e **-a** son incompatibles, polo que se deberá notificar que "os parámetros non son compatibles" cando os usuarios introduzcan a súa combinación.

Por último, destacar que tamén se pode lanzar **diskusage** sobre unha expresión do estilo **carpeta*** ou *. Neste caso o listado terá en conta únicamente aquelas entradas de directorio ou arquivos que emparellen coa expresión introducida (podedes comprobar que o interprete de comandos do sistema operativo xa proveerá o listado de ficheiros que cumplen a expresión no array argv[] que recibe o programa **diskusage**.

Deste xeito a sintaxe do programa diskusage será a seguinte:

$diskusage [-al-s] [-o] [-u] [-h] ficheiro|directorio|expresión (os elementos entre [] son opcionais).

Exemplos de execución:

$diskusage -a

```
4       ./fichero2.txt
334220  ./carpeta2/fichero6.txt
334224  ./carpeta2
0       ./fichero3.txt
4       ./carpeta/fichero4.txt
4       ./carpeta/fichero3.txt
4       ./carpeta/subcarpeta/fichero5.txt
8       ./carpeta/subcarpeta
20      ./carpeta
4       ./fichero1.txt
334256  .
```

$diskusage -s carpeta

```
20  carpeta
```

$diskusage -b carpeta

```
4207  carpeta/subcarpeta
8511  carpeta
```

$diskusage -o

8 ./carpeta/subcarpeta
20 ./carpeta
334224 ./carpeta2
334256 .

$diskusage -o -u

8 ./carpeta/subcarpeta
20 ./carpeta
334224 ./carpeta2
334256 .

Resumo de uso por usuarios.
pepe 24
juan 334232

$diskusage -a -s

Non se poden resumir e mostrar todas as entradas simultáneamente. Probe 'diskusage -h' para máis información.

Comentarios

- A información de cada ficheiro pode obterse con stat ou lstat.
- As funcións opendir, readdir, closedir, rewinddir 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. Cando non sexa posible obter o nome do propietario dun ficheiro/directorio ($diskusage -u), amosarase simplemente o seu "identificador de usuario".
- O lugar apuntado por un link simbólico pódese coñecer coa función realpath. Recordemos que (ln -s permite crear links simbólicos).
- Para a ordeación de datos poderase empregar a función **qsort** dispoñible na librería standard de C.

Modo de entrega

As prácticas entregaranse por email antonio.fari.so@gmail.com ou jcasanova@udc.es antes de proceder a 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 1º Integrante Grupo][Nome 2º Integrante Grupo] // de ser só un 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. É habitual que, durante a defensa, o profesor lle pida a algún dos integrantes a realización de pequenos cambios no

código que se poidan considerar pertinentes (para solucionar algún problema/ engadir algunha pequena variante). Nese caso, o alumno deberá amosar a súa solvencia á hora de abordar o cambio solicitado.

As prácticas entregadas e que posteriormente non se saiban defender correctamente, "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.
  - Funcionará correctamente. O alumno debe 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: Domingo **29 de novembro de 2009 (ata as 23:59:59h)** a partires dese intre calquera práctica recibida será considerada **fóra de prazo** (o que implica que a súa valoración máxima será dividida por 2). Calquera práctica recibida posteriormente ao 13 de decembro será considerada "non-apta".

Detalles a recordar

- O valor máximo das prácticas é de 1.5 puntos. Cómpre que todas as prácticas reciban a calificación de "apta", para poder aprobar a asignatura.
- Non todas as prácticas teñen o mesmo valor. En principio esta primeira práctica ten un valor de 0.75 Sen embargo, os profesores resérvanse a posibilidade de modificar esta valoración (+-0.15).
- Para acadar a máxima nota a práctica debe funcionar correctamente, e a defensa da mesma deberá ser adecuada.
- A detección de prácticas copiadas (salvo raras excepcións) implicará a cualificación da práctica como "non-apta" na convocatoria actual.