

SISTEMAS OPERATIVOS 1.

Curso 2010-2011

Práctica 1: Copy

Desenvolver o programa **copy** que permita realizar a copia de arquivos e directorios dentro do sistema de arquivos. A copia xerada será completamente independente do orixinal. O programa *copy* a desenvolver será unha simplificación do comando *cp* dispoñible no sistema operativo GNU/Linux.

Será posible copiar un arquivo a outro, ou copiar arbitrariamente varios ficheiros a un directorio de destino.

Sintaxe do comando

```
$ copy [OPCIONES] <ruta orixe> <ruta destino>  
$ copy [OPCIONES] <ruta orixe1> <ruta orixe2> ... <ruta orixe> <ruta directorio destino>
```

Se se dan dous nomes de arquivo aos argumentos orixe e destino, *copy* copiará o ficheiro orixe no destino. Se a ruta de destino é un directorio, *copy* copiará cada un dos ficheiros orixe no directorio destino empregando os nomes dos ficheiros orixe na ruta de destino.

Por defecto *copy* non copiará directorios. Sen embargo, a opción “-R” fará que *copy* copie recursivamente descendendo na estrutura dos directorios orixe e copiando os arquivos aos seus correspondentes directorios de destino. Neste caso espérase que o *copy* reciba só dous parámetros:

```
$ copy [OPCIONES] <ruta orixe> <ruta destino>
```

Cando se copia un enlace simbólico, *copy* seguirá o enlace só cando non se está facendo unha copia recursiva (opción “-R” non activada). Cando se realice a copia recursiva copiarase o enlace e non o ficheiro enlazado.

```
$ copy <rutaLink> <rutaDestino> --> Mira a onde apunta <rutaLink> e crea unha copia en <rutaDestino>
```

Por defecto, *copy* non permitirá copiar un arquivo sobre si mesmo.

O comando *copy* deberá ter en conta e informar ao usuario cando non pode realizar as operacións e abortalas cando non sexa posible, coma por exemplo por non ter permisos de lectura sobre os arquivos de orixe, por non ter permisos de escritura nos directorios de destino. Porque xa existe o arquivo de destino e non está activada a opción de forzar sobreescritura.

Os arquivos das copias realizadas por *copy* deben manter os mesmos permisos que os arquivos orixinais.

Unha vez finalizada a execución de *copy* informarase por pantalla do número de bytes que foron copiados coa execución do comando. Exemplo: “copy copiou 2600 bytes.”

Sintaxe das opcións

```
$copy [-R] [-t numBytes] [-f] [-v] <ruta orixe> ... <ruta destino>
```

Parámetros opcionais:

-R : copia directorios recursivamente. Esta versión recursiva non segue enlaces simbólicos nas rutas de orixe. Unicamente copia os arquivos regulares, directorios e enlaces simbólicos que se atopan por debaixo da estrutura de directorios das rutas de orixe.

-t numBytes : limita o tamaño máximo por arquivo que se pode copiar. Indica a *copy* que os arquivos que ocupan máis de *numBytes bytes* non serán copiados.

-f: permite que *copy* sobrescriba arquivos xa existentes coa ruta de destino. Se a opción -f non está indicada

copy deberá dar un erro sempre que se intente sobrescribir un arquivo que exista previamente.

-v: activa a visualización detallada do proceso de copia imprimindo liña a liña a ruta dos ficheiros antes de copialos e a ruta do destino unha vez copiados. Exemplo: “ficheiro.out -> copia.out”

Comentarios

- A información de cada ficheiro obtense con `stat` (man 2 `stat`) ou `lstat` (man `lstat`)
- As funcións `opendir`, `readdir` e `closedir` poden ser utilizadas para obter o contido dun directorio dado.
- A función de `C mkdir` permite crear un novo directorio (man 2 `mkdir`)
- O lugar apuntado por un enlace simbólico pódese coñecer coa función `realpath`. Recordemos que (`ln -s` permite crear links simbólicos dende a terminal). A función `symlink` permite a creación de enlaces simbólicos.
- As funcións `open`, `creat`, `close`, `read` e `write` permiten a apertura, creación, peche, escritura e lectura de ficheiros. A copia dun ficheiro poderá realizarse pasando todo o contido do arquivo orixe a memoria e posteriormente escribindo o contido desa memoria a disco.

Modo de entrega

As prácticas entregaranse por email a antonio.fari.so@gmail.com (grupos EI e grupo ETIX venres) ou jcasanova@udc.es (grupos ETIX mércores) 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]::[ETIX ou EI]::[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 da rede de docencia concatenados: `nome1.apellido.apellido-nome2.apellido.apellido.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, e 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 debe ter en conta os valores que devolven todas as funcións que utilice (OLLO aos valores devoltos polas funcións empregadas!!).
 - Non debe conter *memory-leaks* (o profesor usará **valgrind** para revisar este aspecto).
- A data límite de entrega será o: Domingo 7 de novembro de 2010.

Detalles a recordar

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