Operating Systems

Grado en Informática. Course 2016-2017

Assigment 1: File systems

Use the C language to code these two programs

list
- lists the files and/or directories supplied to it as command line arguments
- its complete sysntax is

  `list [-r] name1 name2 name3 ....`

  where

  * **-r** stands for recursive: if present, all directories must be listed recursively (brackets indicate it is an optional argument)
  * **name1 name2 name3 ...** are the names of the files (or directories) to list. If no name is given it will list the current directory (recursively or not depending on the presence of *-r*)
- it will produce a line for each item listed, in the same format as `ls -li` (resolving, if necessary, symbolic links)

```
662223 lrwxrwxrwx 1 root staff 16 Mar  3  2016 /usr/local/java -> jdk1.8.0_73/jre/
```

- names can contain wildcard characters (`*`,`?`...).**IMPORTANT: DO NOT DO THIS UNTIL EVERYTHING ELSE IS DONE**

deltree
- deletes a directory recursively (directory and all of its contents)
- its syntax is:

  `deltree dirname`
- if no *dirname* is given it will do nothing
- if a file or directory could not be removed, an appropiate message should be sent to the user

- These programs should compile cleanly (produce no warnings even when compiling with `gcc -Wall`)

- **NO RUNTIME ERROR WILL BE ALLOWED (segmentation, bus error . . .)**. Programs with runtime errors will yield no score.
- These programs can have no memory leaks
- When one program cannot perform its task (for whatever reason, for example, lack of privileges) it should inform the user
- All input and output is done through the standard input and output

Information on the system calls and library funtions needed to code these two programas is available through `man`: (*opendir, readdir, stat, lstat, unlink, rmdir, realpath . . .).*

The following funtions convert the mode of one file (in a *mode_t* integer) to `"rwxrwxrwx"` form. Note that the three functions have different ways of assigning memory

```
char TipoFichero (mode_t m)
{
  switch (m&S_IFMT) { /*and bit a bit con los bits de formato,0170000 */
    case S_IFSOCK:  return ’s’;  /*socket */
    case S_IFLNK:   return ’l’;    /*symbolic link*/
    case S_IFREG:   return ’-’;   /* fichero normal*/
    case S_IFBLK:   return ’b’;   /*block device*/
    case S_IFDIR:   return ’d’;   /*directorio */
    case S_IFCHR:   return ’c’;   /*char  device*/
    case S_IFIFO:   return ’p’;   /*pipe*/
    default: return ’?’;   /*desconocido, no deberia aparecer*/
  }
}

char * ConvierteModo (mode_t m, char *permisos)
{
  strcpy (permisos,"---------- ");

  permisos[0]=TipoFichero(m);
  if (m&S_IRUSR) permisos[1]=’r’;  /*propietario*/
  if (m&S_IWUSR) permisos[2]=’w’;
  if (m&S_IXUSR) permisos[3]=’x’;
  if (m&S_IRGRP) permisos[4]=’r’;   /*grupo*/
  if (m&S_IWGRP) permisos[5]=’w’;
  if (m&S_IXGRP) permisos[6]=’x’;
  if (m&S_IROTH) permisos[7]=’r’;   /*resto*/
  if (m&S_IWOTH) permisos[8]=’w’;
  if (m&S_IXOTH) permisos[9]=’x’;
  if (m&S_ISUID) permisos[3]=’s’;  /*setuid, setgid y stickybit*/
  if (m&S_ISGID) permisos[6]=’s’;
  if (m&S_ISVTX) permisos[9]=’t’;
  return permisos;
}

char * ConvierteModo2 (mode_t m)
{
  static char permisos[12];
  strcpy (permisos,"---------- ");

  permisos[0]=TipoFichero(m);
  if (m&S_IRUSR) permisos[1]=’r’;  /*propietario*/
  if (m&S_IWUSR) permisos[2]=’w’;
  if (m&S_IXUSR) permisos[3]=’x’;
  if (m&S_IRGRP) permisos[4]=’r’;   /*grupo*/
  if (m&S_IWGRP) permisos[5]=’w’;
  if (m&S_IXGRP) permisos[6]=’x’;
  if (m&S_IROTH) permisos[7]=’r’;   /*resto*/
  if (m&S_IWOTH) permisos[8]=’w’;
  if (m&S_IXOTH) permisos[9]=’x’;
  if (m&S_ISUID) permisos[3]=’s’;  /*setuid, setgid y stickybit*/
  if (m&S_ISGID) permisos[6]=’s’;
  if (m&S_ISVTX) permisos[9]=’t’;
  return (permisos);
}

char * ConvierteModo3 (mode_t m)
{
  char * permisos;
  permisos=(char *) malloc (12);
  strcpy (permisos,"---------- ");

  permisos[0]=TipoFichero(m);
  if (m&S_IRUSR) permisos[1]='r';  /*propietario*/
  if (m&S_IWUSR) permisos[2]='w';
  if (m&S_IXUSR) permisos[3]='x';
  if (m&S_IRGRP) permisos[4]='r';   /*grupo*/
  if (m&S_IWGRP) permisos[5]='w';
  if (m&S_IXGRP) permisos[6]='x';
  if (m&S_IROTH) permisos[7]='r';   /*resto*/
  if (m&S_IWOTH) permisos[8]='w';
  if (m&S_IXOTH) permisos[9]='x';
  if (m&S_ISUID) permisos[3]='s';  /*setuid, setgid y stickybit*/
  if (m&S_ISGID) permisos[6]='s';
  if (m&S_ISVTX) permisos[9]='t';
  return (permisos);
}
```

WORK SUBMISSION

- Work must be done in pairs.
- The source code will be submitted to the subversion repository under a directory named **P1**
- The name of the main programs will be `list.c` and `deltree.c`. Programs must be able to be compiled with `gcc list.c` and `gcc deltree.c` Alternatively a `Makefile` can be supplied so that both programs can be compiled with just `make`
- Only one of the members of the workgroup will submit the source code. The names and logins of all the members of the group should be in the source code of the main programs (at the top of the file)

DEADLINE: October 18th, 2016

ASSESSMENT: October, 21st through 25th