

Instituto Tecnológico de Costa Rica

Escuela de Computación

Curso: Sistemas Operativos

## Tarea 7

El módulo *fd*, cuenta con una serie de funciones para manejo de archivos similares a las proporcionadas por el lenguaje C. Las funciones típicas para el manejo de archivos son: *fopen*, *fwrite*, *fread* y *fclose*.

La ventaja de utilizar estas funciones en *tasklib* es que mientras una tarea espera por la lectura o escritura de datos, otra puede ejecutarse. El siguiente ejemplo muestra un programa que crea varias tareas donde cada una realiza la copia de uno de los archivos que se pasan como parámetros, por ejemplo:

```
backup ejemplo1.c ejemplo2.c ejemplo3.c
```

Al final, se obtiene una copia de cada archivo pero con extensión “.old”. Note que *libtask* utiliza funciones con nombre *fdwrite* y *fdread* en lugar de las equivalentes *fwrite* y *fread*.

```
#include <unistd.h>
#include <fcntl.h>
#include <task.h>
#include <fd.h>

#define PERMS 0666
#define BUFSIZE 1024
#define STACK 32768

void copytask(void *arg) {
    FILE *f1, *f2;
    char buf[BUFSIZE];
    char backfile[32];
    int n;

    if ((f1 = fopen(arg, "r")) == -1) {
        printf("cp: can't open %s\n", arg); taskexit(-1);
    }
    fdnblock(f1);
    strcpy(backfile, arg);
    strcat(backfile, ".old");
    if ((f2 = fopen(backfile, "w")) == -1) {
```

```

    printf("cp: can't create %s, mode %03o\n", backfile, PERMS);taskexit(-1);
}
fdnblock(f2);
while ((n = fdread(f1, buf, BUFSIZ)) > 0)
    if (fdwrite(f2, buf, n) != n)
        printf("cp: write error on file %s\n", backfile);
fclose(f1);
fclose(f2);
taskexit(0);
}
void taskmain(int argc, char **argv) {
    int i;
    if (argc == 0) {
        printf("Usage: backup file1 file2 ..\n"); taskexit(-1);
    }
    for (i=0; i < argc; i++) {
        taskcreate(copytask, (void*)argv[i], STACK);
    }
}

```

## Ejercicios

1. Escriba un programa similar al anterior pero que no utilice la librería *libtask* y que ejecute una copia después de otra. Agregue a ambos programas una función que permita determinar cuánto tiempo consumen en su ejecución (puede usar las funciones *time* y *difftime* de C). Compare el tiempo de ejecución de ambas versiones.
2. Podría suceder que el tiempo de ejecución de las dos versiones anteriores no sea significativo, pues este programa lo único que hace es leer o escribir. Por eso, escriba dos nuevas versiones del programa anterior (una con la librería y otra sin utilizarla) que generen una copia de los archivos que se pasa como parámetros pero con todo el texto en mayúscula. (utilice la función *toupper* disponible en *ctype.h*). Compare nuevamente el tiempo de ejecución de ambas versiones.

*Nota:* para realizar las pruebas pueden utilizar los archivos de las investigaciones que están disponibles en *Google Drive* bajo el directorio *Investigaciones*.