

Sisteme de operare

Laborator 2

File I/O

1. Creati un nou subdirector *lab2/* in structura de directoare a laboratorului creata anterior (*SO/laborator*) si subdirectoarele aferente *doc*, *src* si *bin*. Nu uitati sa actualizati variabila de mediu *PATH* pentru a include directorul *SO/laborator/lab2/bin*.

2. Scrieti un program C **mycp.c** care primeste doi parametri in linia de comanda: primul parametru este numele unui fisier care trebuie copiat intr-un alt fisier al carui nume este furnizat de al doilea parametru, ca mai jos:

```
$ mycp fisier-src fisier-dst
```

Comparati programul **mycp.c** cu programul **mycp.c** pe care l-ati scris in laboratorul trecut si care copia in bucla date de la file descriptorul 0 si le scria pe file descriptorul 1. Ce observati?

3. Modificati programul **mycp.c** din laboratorul 1, care citeste in bucla date de la *stdin* (file descriptor 0) si le scrie la *stdout* (file descriptor 1) a.i. va primi doua argumente care reprezinta numele fisierului sursa care trebuie copiat si numele fisierului destinatie. Fara sa modificati deloc bucla de copiere (adica pastrati file descriptorii 0 si 1 folositi), deschideti corespunzator cele doua fisiere primite ca argumente si folosind *dup/dup2* faceti programul sa functioneze ca un program de copiere de fisiere ca la punctul 2.

```
$ mycp fisier-sursa fisier-destinatie
```

Ce se intampla daca fisierul sursa este chiar terminalul pe care lucram, obtinut cu comanda *tty*? De exemplu:

```
$ mycp `tty` fisier-destinatie
```

4. Modificati programul **mycp.c** de la punctul 2 a.i. dupa ce deschide fisierul sursa, programul seteaza file descriptorul obtinut pentru acces non-blocant cu ajutorul apelului sistem *fcntl*. Care e diferenta dintre executia celor doua apeluri ale comenzii din secventa de comenzi de mai jos? Cum va explicati rezultatul?

```
$ mycp fisier-sursa fisier-destinatie  
$ rm fisier-destinatie  
$ mycp `tty` fisier-destinatie
```

5. Scrieti un program C **hole.c** care creaza un fisier cu “goluri/gauri” in el. De pilda, dupa ce ati creat fisierul, scrieti un sir de caractere la inceputul fisierului, mutati offsetul curent din fisier la valoarea 100 si scrieti un alt sir de caractere acolo. Care este dimensiunea fisierului nou creat in octeti afisata de comenzi cum ar fi *ls -l <fisier_cu_goluri>* sau *du -sb <fisier_cu_goluri>*?

Dar comanda *od -c <fisier_cu_goluri>* ce va arata?

6. Scrieti un program C **true-du.c** care primeste un fisier ca parametru si afiseaza dimensiunea lui in octeti fara goluri (adica dimensiunea sa reala, fara goluri, avand in vedere ca acestea nu se stocheaza pe disc).