Kolejki FIFO

To laboratorium ma na celu zapoznanie się z zasadą tworzenia łącz nazwanych, oraz zasady komunikacji pomiędzy dwoma niespokrewnionymi procesami.

1.Wstęp

•Kolejki FIFO są podobne do łącz PIPE. Też umożliwiają przepływ danych w jedną stronę, także korzysta się w nich z deskryptorów plików jednak mają jedną zaletę nad łączami nienazwanym, możemy je zidentyfikować po nazwie. Dzięki temu, możemy je wykorzystać do komunikacji pomiędzy nie spokrewnionymi procesami.

•Kolejki tworzymy za pomocą funkcji:

int mknod(char *nazwa_pliku,int mode, int dev);

•nazwa_pliku - plik tymczasowy

•mode - logiczna suma uprawnień i S_IFIFO

•dev - numer urządzenia, może być 0

Można także użyć polecenia systemowego:

mknod nazwa_pliku p

•Usunięcie następuje przez:

unlink (char *nazwa_pliku);

Z kolejki można korzystać po otworzeniu jej do czytania lub pisana standardową funkcją open.

2.Na rozgrzewkę 1 pkt.

•Utworzyć przez mknod /tmp/fifo.1 p kolejkę fifo.

•Sprawdzić przez ls -la czy plik powstał i jakiego jest typu

•Przepisać program który zapisuje do kolejki prosty komunikat

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <errno.h>
extern int errno;
#define FIFO1 "/tmp/fifo.1"
main()
{
char buff[]="ala ma kota a kot ma ale";
int fifo;
if ((fifo = open(FIFO1, O_WRONLY )) < 0)
    perror("nie moze otworzyc fifo1 do pisania");
if (strlen(buff) != write(fifo,buff,strlen(buff)))
    perror("blad zapisu do fifo");
exit(0);
}
```

•Napisać analogiczny program który czyta z kolejki komunikatów

•Uruchomić program czytający następnie piszący a potem na odwrót

3.Stosowanie O_NDELAY 1 pkt

•Przy funkcji open zastosować flagę O_NDELAY, najpierw przy jednym programie potem przy drugim

•Uruchomić pierwszy potem drugi

•Uruchomić drugi potem pierwszy

•Porównać wyniki. Co daje O_NDELAY?

4. Automatyzacja 1 pkt

Do programu piszącego dodać funkcje tworzące kolejkę fifo a do odbierającego usuwającą ją.

5. Komunikator 1 pkt

•Stworzyć prosty komunikator:
•dwa osobne programy
•dwa terminale
•komunikacja w obie strony (2 kolejki potrzebne)
•W wersji podstawowej - na przemian to na jednym to na drugim terminalu piszemy tekst, który powinien pojawiać się na terminalu odbierającym.
•W wersji rozszerzonej - stworzyć dwa osobne procesy, jeden proces odpowiedzialny za odbieranie drugi za wysyłanie. Wtedy wysyłanie i odbieranie będzie niezależne (1 pkt)

6. Sprawozdanie

Pamiętajcie o sprawozdaniu