

## Łączy PIPE

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

### 1. Wstęp

- Łączy komunikacyjne nie nazwane (PIPE) możemy użyć jedynie pomiędzy dwoma spokrewnionymi procesami.
- stworzymy za pomocą funkcji:  
`int pipe(int *filedes);`
- `filedes` - tablica dwuelementowa
- `filedes[0]` - jest deskryptorem pliku otwartym do czytania
- `filedes[1]` - jest deskryptorem pliku otwartym do pisania
- łączy komunikacyjne zwykle używamy do komunikowania się dwóch procesów, ale poniższy przykład będzie na jednym:

```
main()
{
    int pipefd[2],n;
    char buff[100];
    if (pipe(pipefd) <0)
        perror("blad pipe\n");
    printf("read fd = %d, write fd = %d\n",pipefd[0],pipefd[1]);
    if (write(pipefd[1],"hello world\n",12) !=12)
        perror("blad zapisu\n");
    if ((n = read(pipefd[0],buff,sizeof(buff))) <=0)
        perror("blad odczytu\n");
    write(1, buff, n); /* deskryptor pliku=1 wyjście standardowe */
    exit(0);}
```

### 2. Cel

Napisać program typu klient-serwer który:

- pobiera od użytkownika nazwę pliku
- tworzy łączy nienazwane PIPE
- tworzy nowy proces za pomocą funkcji `fork` (nie `popen`!)
- przesyła do procesu potomnego nazwę pliku\_
- proces potomny odczytuje plik o podanej nazwie a zawartość (tekst) lub informację o możliwych błędach (np. brak pliku), przesyła do procesu rodzicielskiego
- proces rodzicielski wyświetla na standardowym wyjściu co otrzymał od potomka

### 3. Pomocne funkcje:

- `gets` lub `fgets`
- `perror`, `strerror`
- `read`, `write`
- `fork`, `wait`
- `pipe`, `close`
- `strcat`, `strlen`, `strcpy`

### 4. Uwaga końcowa

Pamiętać należy o przydziale pamięci na przesyłane komunikaty. Ważny jest też fakt że łączy pipe służy jedynie do przekazywania danych w jedną stronę. Jeżeli chcemy mieć komunikację dwukierunkową trzeba stworzyć 2 łączy. 1 zamknąć u potomka do odczytu a u przodka do zapisu a drugie u potomka do zapisu a u przodka do odczytu. Plik może być dowolnej długości odczytywanie jak i przesyłanie zrealizować w pętli (nie jednokrotnie odczyt).

### 5. Sprawozdanie

Oczywiście nie zapomnijcie o sprawozdaniu.