

INSTYTUT TELEINFORMATYKI I AUTOMATYKI

Wydział Cybernetyki WAT

Przedmiot: SYSTEMY OPERACYJNE

SPRAWOZDANIE Z ĆWICZENIA LABORATORYJNEGO Nr 8

Temat ćwiczenia: ŁĄCZA NAZWANE I NIENAZWANE

Wykonał:

Piotr Matyjek
Grupa: **I6Y3S1**

Ćwiczenie wykonane dnia
08.12.2017

Prowadzący ćwiczenie
mgr. inż. Stanisław Matusiak

Ocena:

.....

Opis rozwiązania

Przydzielony został mi numer zadania 3. W pierwszym programie „matyjek_pipe1.c” najpierw tworzone są dwa pliki fifo, czyli łącza nazwane. Z pliku „fifo1_2” będzie korzystał program „matyjek_pipe1.c” w trybie zapisu, w trybie czytania będzie korzystał program „matyjek_pipe2.c”. Drugi program będzie korzystał też z drugiego łącza nazwanego „fifo2_3” w trybie zapisu. Trzeci program będzie korzystał z „fifo2_3” w trybie odczytu.

Następnie w „matyjek_pipe1.c” otwierany jest deskryptor pliku „fifo1_2” używając funkcji read.

Następnie program prosi o podanie ile liczb rzeczywistych będzie użytkownik podawał (zmienna „n”). Po odczytaniu „n”, ta jest zapisywana do deskryptora pliku, po czym program wchodzi do pętli while, o ile $n > 0$. Plik fifo jest samo synchronizujący, więc po operacji zapisu, kolejna operacja zapisu nie powiedzie się dopóki to co zapisane w pliku nie zostanie odczytane przez inny program. W pętli alokowana jest pamięć dla tablicy liczb typu „double”, a następnie w pętli „for” program będzie odczytywał liczby rzeczywiste i zapisywał je do pliku fifo, aby były odczytane przez drugi program. Po wyjściu z pętli „for”, program odczytuje nową wartość „n”, którą przesyła do drugiego programu poprzez łącze nazwane i usuwa tablicę liczb rzeczywistych. Następnie program zapętla się o ile $n > 0$. Gdy $n = 0$ program wychodzi z pętli „while”, zamyka deskryptor pliku i usuwa oba pliki łączy nazwanych.

W drugim programie „matyjek_pipe2.c” pierwszą funkcją wykonywaną jest sleep, usypiający proces na jedną sekundę. Funkcja ta została wykorzystana, aby wykorzystując skrypt do uruchomienia tych trzech programów, drugi program nie próbował otwierać pliku łącza nazwanego, który jest tworzony w programie pierwszym. Następnie otwierany jest deskryptor pliku „fifo1_2”, odczytywana jest wartość „n” z tego pliku, a następnie otwierany jest kolejny deskryptor pliku łącza nazwanego „fifo2_3”. Do tego pliku łącza zapisywana jest wartość „n”, aby trzeci program mógł ją odczytać. Następnie w pętli „while” z warunkiem „n” zerowana jest zmienna „suma”, po czym wykonywana jest pętla „for”. W tej pętli z kolejnymi iteracjami, program odczytuje z łącza „fifo1_2” kolejne liczby rzeczywiste i sumuje zapisując wyniki w „suma”. Wartość tej zmiennej jest każdorazowo wyświetlana na ekranie. Po zakończeniu pętli „for” program zapisuje do pliku łącza „fifo2_3” wartość zmiennej „suma”, a następnie z „fifo1_2” odczytuje kolejną wartość „n” i tą znowu zapisuje do łącza „fifo2_3”. Gdy $n > 0$, to wykonywana jest kolejna iteracja w pętli „while”. Jeżeli $n = 0$, to zamykane są deskryptory plików i program się kończy.

Na początku wykonywania trzeciego programu, ten tak samo jak drugi jest usypiany na jedną sekundę. Funkcja sleep wywoływana jest, aby po uruchomieniu programu poprzez skrypt, program ten poczekał na uruchomienie programu pierwszego, który tworzy pliki łączy nazwanych. Pierwszą funkcją po sleep, jest funkcja otwierająca deskryptor pliku w trybie odczytu, a następnie odczytywana jest wartość „n” z tego pliku. Następnie program wchodzi do pętli „while”, o ile $n > 0$ i odczytywana jest wartość sumy liczb rzeczywistych obliczona w programie drugim. Następnie obliczana jest średnia arytmetyczna i na ekran wyświetlana jest wartość tej średniej, po czym z pliku łącza „fifo2_3” odczytywana jest kolejna wartość „n”. Gdy $n > 0$ to wykonywana jest kolejna pętla, a jak $n = 0$, to zamykany jest deskryptor pliku i program się kończy.

Do uruchomienia programów został przygotowany skrypt. Uruchomienie go spowoduje poprawne wykonywanie się wszystkich trzech programów, lecz na początku trudno będzie zrozumieć co pojawia się na ekranie. Wy tłumaczone to zostało w sekcji „Wyniki uruchomienia”. Zalecane jest uruchamianie programów z trzech oddzielnych konsol (na jednej konsoli jeden z trzech programów) w celu zwiększenia czytelności i zrozumienia wyświetlanych wyników.

Kody źródłowe programów

matyjek_pipel.c

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/types.h>
#include <stdlib.h>
#include <sys/stat.h>

int main()
{
    int i, filedesc;
    unsigned int n;
    double *tab;

    mkfifo("fifo1_2", S_IWUSR | S_IRUSR );
    mkfifo("fifo2_3", S_IWUSR | S_IRUSR );

    filedesc = open("fifo1_2", O_WRONLY);

    printf("podaj ile liczb: ");
    scanf("%d", &n);

    write(filedesc, &n, sizeof(int));
    while(n)
    {
        tab = malloc(sizeof(double)*n);

        for(i=0;i<n;i++)
        {
            printf("podaj liczbe rzeczywista: ");
            scanf("%lf", &tab[i]);

            write(filedesc, &tab[i], sizeof(double));
        }

        printf("podaj ile liczb: ");
        scanf("%d", &n);

        write(filedesc, &n, sizeof(int));

        free(tab);
    }

    close(filedesc);
```

Matyjek_pipe2.c

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>

int main()
{
    unsigned int n;
    int i, filedesc, fd2;
    double liczba, suma = 0;

    sleep(1);          //sleep po to aby poprawnie uruchamiał się poprzez skrypt

    filedesc = open("fifo1_2", O_RDONLY);
    read(filedesc, &n, sizeof(int));

    fd2 = open("fifo2_3", O_WRONLY);
    write(fd2, &n, sizeof(int));

    while(n)
    {
        suma=0;
        for(i=0;i<n;i++)
        {
            read(filedesc, &liczba, sizeof(double));
            //usleep(5000);
            suma=suma+liczba;

            printf("\nsuma= %lf\n",suma);
        }
        write(fd2, &suma, sizeof(double));

        read(filedesc, &n, sizeof(int));
        write(fd2, &n, sizeof(int));
    }

    close(filedesc);
    close(fd2);

    return 0;
}
```

matyjek_pipe3.c

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>

int main()
{
    unsigned int n;
    int i, filedesc;
    double liczba, suma = 0;

    sleep(1);          //sleep po to aby poprawnie uruchamiał się poprzez skrypt

    filedesc = open("fifo2_3", O_RDONLY);
    read(filedesc, &n, sizeof(int));

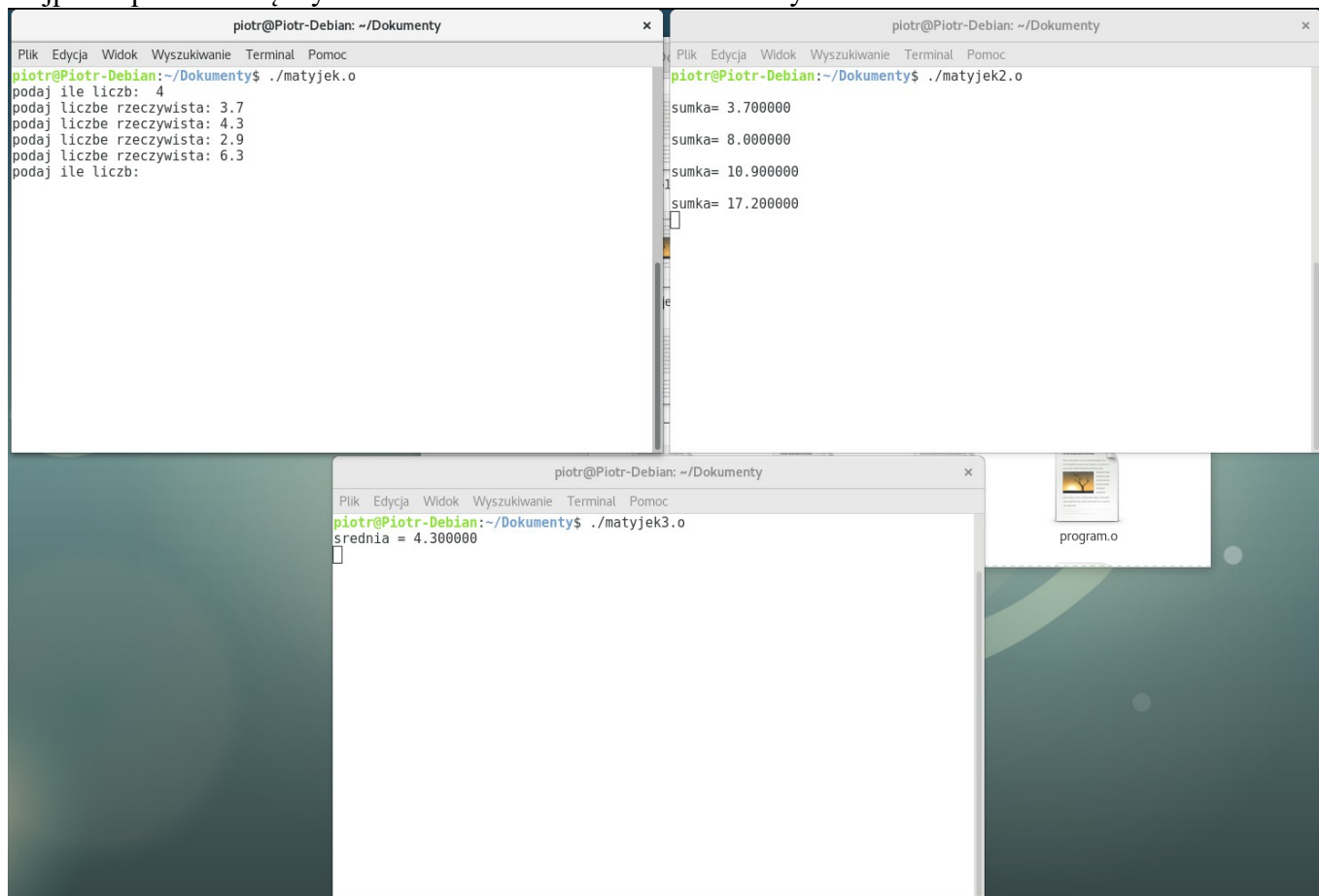
    while(n)
    {
        read(filedesc, &suma, sizeof(double));

        liczba=suma/n;
        printf("srednia = %lf\n",liczba);
        read(filedesc, &n, sizeof(int));
    }
    close(filedesc);

    return 0;
}
```

Wyniki uruchomienia

Najpierw przedstawię wyniki uruchomienia na trzech oddzielnych konsolach:



The image shows three terminal windows from a Linux desktop environment. The top-left window shows the execution of a program that prompts for the number of numbers and then for each number, calculating their sum. The top-right window shows the execution of a program that calculates the sum of four specific numbers. The bottom window shows the execution of a program that calculates the average of four specific numbers. A file icon labeled 'program.o' is visible on the desktop.

```
piotr@Piotr-Debian: ~/Dokumenty
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
piotr@Piotr-Debian:~/Dokumenty$ ./matyjek.o
podaj ile liczb: 4
podaj liczbe rzeczywista: 3.7
podaj liczbe rzeczywista: 4.3
podaj liczbe rzeczywista: 2.9
podaj liczbe rzeczywista: 6.3
podaj ile liczb:

piotr@Piotr-Debian: ~/Dokumenty
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
piotr@Piotr-Debian:~/Dokumenty$ ./matyjek2.o
sumka= 3.700000
sumka= 8.000000
sumka= 10.900000
sumka= 17.200000

piotr@Piotr-Debian: ~/Dokumenty
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
piotr@Piotr-Debian:~/Dokumenty$ ./matyjek3.o
srednia = 4.300000
```

W lewej-górnej konsoli pracuje program „matyjek_pipe1.c”, w prawej-górnej konsoli pracuje program „matyjek_pipe2.c”, a w dolnej „matyjek_pipe3.c”.

W obecnej sytuacji programy czekają na podanie kolejnej wartości „n”.

```
piotr@Piotr-Debian: ~/Dokumenty
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
piotr@Piotr-Debian:~/Dokumenty$ ./matyjek.o
podaj ile liczb: 4
podaj liczbe rzeczywista: 3.7
podaj liczbe rzeczywista: 4.3
podaj liczbe rzeczywista: 2.9
podaj liczbe rzeczywista: 6.3
podaj ile liczb: 5
podaj liczbe rzeczywista: 4.7
podaj liczbe rzeczywista: 3.67
podaj liczbe rzeczywista: 2.89
podaj liczbe rzeczywista: 7.93
podaj liczbe rzeczywista: 5.534
podaj ile liczb: 0
piotr@Piotr-Debian:~/Dokumenty$

piotr@Piotr-Debian: ~/Dokumenty
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
piotr@Piotr-Debian:~/Dokumenty$ ./matyjek2.o
suma= 3.700000
suma= 8.000000
suma= 10.900000
suma= 17.200000
suma= 4.700000
suma= 8.370000
suma= 11.260000
suma= 19.190000
suma= 24.724000
piotr@Piotr-Debian:~/Dokumenty$

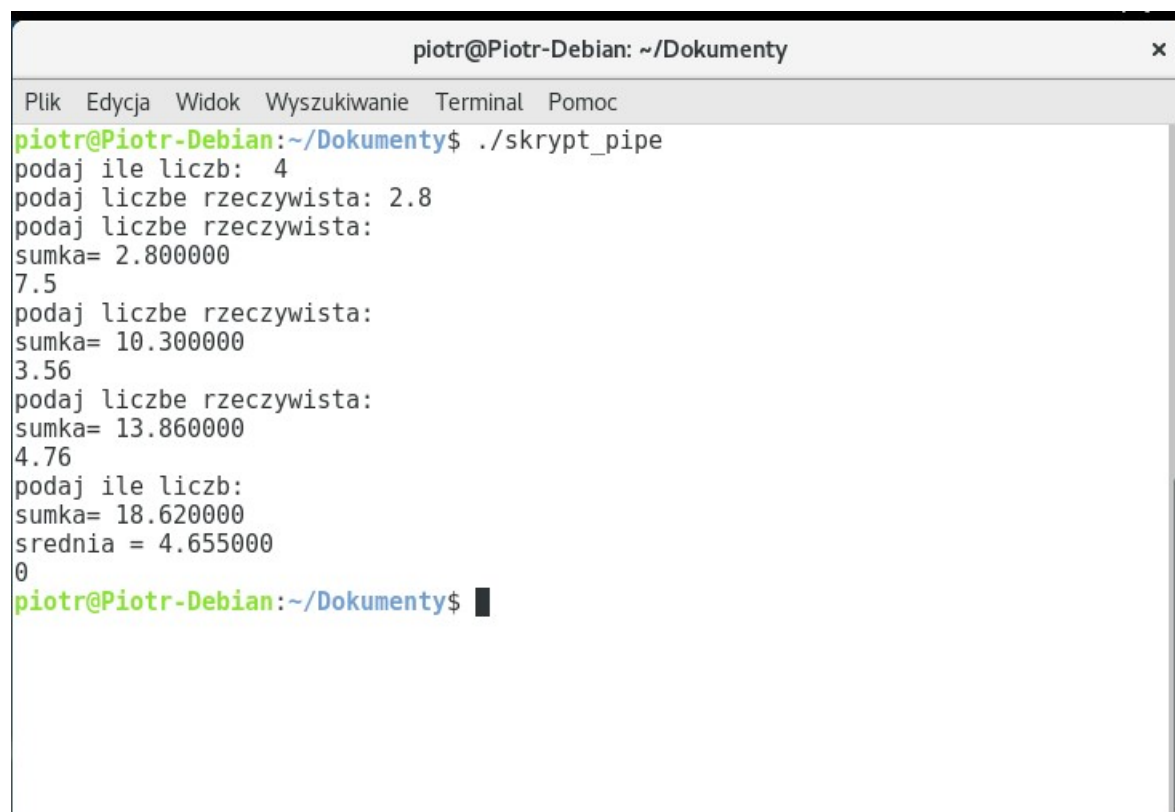
piotr@Piotr-Debian: ~/Dokumenty
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
piotr@Piotr-Debian:~/Dokumenty$ ./matyjek3.o
srednia = 4.300000
srednia = 4.944800
piotr@Piotr-Debian:~/Dokumenty$
```

Podane zostały kolejne liczby rzeczywiste, a następnie zakończono działanie programów poprzez podanie 0 jako wartości „n”.

Wyniki uruchomienia programów poprzez skrypt:

```
piotr@Piotr-Debian: ~/Dokumenty
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
piotr@Piotr-Debian:~/Dokumenty$ ./skrypt_pipe
podaj ile liczb: 4
podaj liczbe rzeczywista: 2.8
podaj liczbe rzeczywista:
suma= 2.800000
7.5
podaj liczbe rzeczywista:
suma= 10.300000
3.56
podaj liczbe rzeczywista:
suma= 13.860000
4.76
podaj ile liczb:
suma= 18.620000
srednia = 4.655000
```

Czerwonymi kółkami oznaczone są miejsca, w których użytkownik wprowadził swoje wartości odpowiednich liczb. Zielonymi kółkami oznaczono linijki tekstu wyświetlone przez program drugi. Fioletowym kółkiem oznaczono linijkę wyświetloną przez program trzeci. Niebieskie kółko oznacza miejsce, w gdzie użytkownik wpisze kolejną wartość „n”. Linijki nieoznaczone są wyświetlane przez program pierwszy. Trudność w zrozumieniu tych wyników polega na tym, że pomiędzy tekstem zachęty, na przykład: „podaj liczbę rzeczywista:”, a linijką w której użytkownik wpisze żadaną wartość wypisywany jest przez program drugi lub trzeci tekst informujący o obecnej sumie liczb lub ich średniej.



```
piotr@Piotr-Debian: ~/Dokumenty
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
piotr@Piotr-Debian:~/Dokumenty$ ./skrypt_pipe
podaj ile liczb: 4
podaj liczbe rzeczywista: 2.8
podaj liczbe rzeczywista:
sumka= 2.800000
7.5
podaj liczbe rzeczywista:
sumka= 10.300000
3.56
podaj liczbe rzeczywista:
sumka= 13.860000
4.76
podaj ile liczb:
sumka= 18.620000
srednia = 4.655000
0
piotr@Piotr-Debian:~/Dokumenty$
```

Zrzut ekranu pokazujący poprawność wykonywania programu.

PODSUMOWANIE

Programy zostały napisane w języku C i uruchomione w środowisku systemu Linux Debian ver. 9. Programy wykonują się poprawnie, jak widać na zamieszczonych powyżej przykładach wywołania programów. Do tego sprawozdania dołączam pliki z kodami programów, których nie zdołałem wysłać na zajęciach, oraz skrypt, który skompiluje programy, uruchomi je, a później usunie pliki wykonywalne. Jedyne zmiany jakie zostały dokonane w stosunku do zaprezentowanego na zajęciach kodu, to dołączenie funkcji sleep do drugiego i trzeciego programu, w celu możliwości poprawnego uruchomienia ich za pomocą skryptu.

W razie rezygnacji z uruchamiania programów za pomocą skryptu, należy skompilować programy i uruchomić je w oddzielnych konsolach w kolejności: „matyjek_pipe1.c”, „matyjek_pipe2.c”, „matyjek_pipe3.c”. Taka kolejność jest wymagana, gdyż w „matyjek_pipe1.c” tworzone są pliki łączące nazwanych.