|  |  |
| --- | --- |
| **Politechnika Świętokrzyska w Kielcach**  **Wydział Elektrotechniki, Automatyki i Informatyki** | |
| Laboratorium Programowanie Współbieżne | |
| Temat:  **Tworzenie i komunikacja za pomocą łączyPIPE** | Autor: **Rafał Robak**  Grupa: **3ID12A** |
| Numer laboratorium: **3** | Data wykonania : **27.11.2020** |

**Cel**  
**Napisać program typu klient-serwer który:**

* **pobiera od użytkownika nazwę pliku 0,5pkt**
* **tworzy łącze nienazwane *PIPE* 0,5pkt**
* **tworzy nowy proces za pomocą funkcji fork (nie *popen*!) 0,5pkt**
* **przesyła do procesu potomnego nazwę pliku. 0,5pkt**
* **proces potomny odczytuje plik o podanej nazwie a zawartość (tekst) lub informację o możliwych błędach (np. brak pliku), przesyła do procesu rodzicielskiego (komunikaty o błędzie 0,5pkt)**
* **proces rodzicielski wyświetla na standardowym wyjściu co otrzymał od potomka (poprawne przesłanie pliku do 2pkt).**

**Kod programu:**

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <unistd.h>

#include <string.h>

#include <errno.h>

int main()

{

int serverfd[2];

int clientfd[2];

int childpid;

if(pipe(serverfd)<0)

{

perror("Blad serverfd");

exit(1);

}

if(pipe(clientfd)<0)

{

perror("Blad clientfd");

exit(2);

}

if((childpid=fork())==-1)

{

perror("Blad fork");

exit(3);

}

if(childpid==0)

{

close(serverfd[0]);

close(clientfd[1]);//zamykam nie uzywane deskryptory

char name[64];

int tmp=read(clientfd[0],name,sizeof(name));//odebranie nazwy pliku

if(tmp<=0)

{

perror("blad odczytu nazwy pliku");

exit(4);

}

printf("%s" ,name);

name[tmp-1]='\0';//Ustawiam ostatni znak bo uzywalem fgets

FILE \*file = fopen(name,"r");

if(file==NULL)

{

printf("%s",strerror(errno));

int status=errno;

write(serverfd[1],&status,sizeof(status));

}

else{

int status=0;

write(serverfd[1],&status,sizeof(status));

char buf[512];

int n;

while((n=fread(buf,1,511,file))>0)

{

write(serverfd[1],buf,n);

}

}

close(serverfd[1]);

close(clientfd[0]);

}

else

{

close(clientfd[0]);

close(serverfd[1]);

char name[64];

printf("Podaj nazwe pliku: ");

fgets(name,64,stdin);

int tmp=strlen(name);

if(write(clientfd[1],name,tmp)!=tmp)//wyslanie nazwy pliku do procesu potomnego

{

perror("Blad wyslania nazwy pliku");

}

int status;

read(serverfd[0],&status,1);

if(status!=0)

printf("%s\n",strerror(status));

else

{

int n;

char buf[511];

while((n=read(serverfd[0],&buf,sizeof(buf)))>0)

write(1,buf,n);

}

close(clientfd[1]);

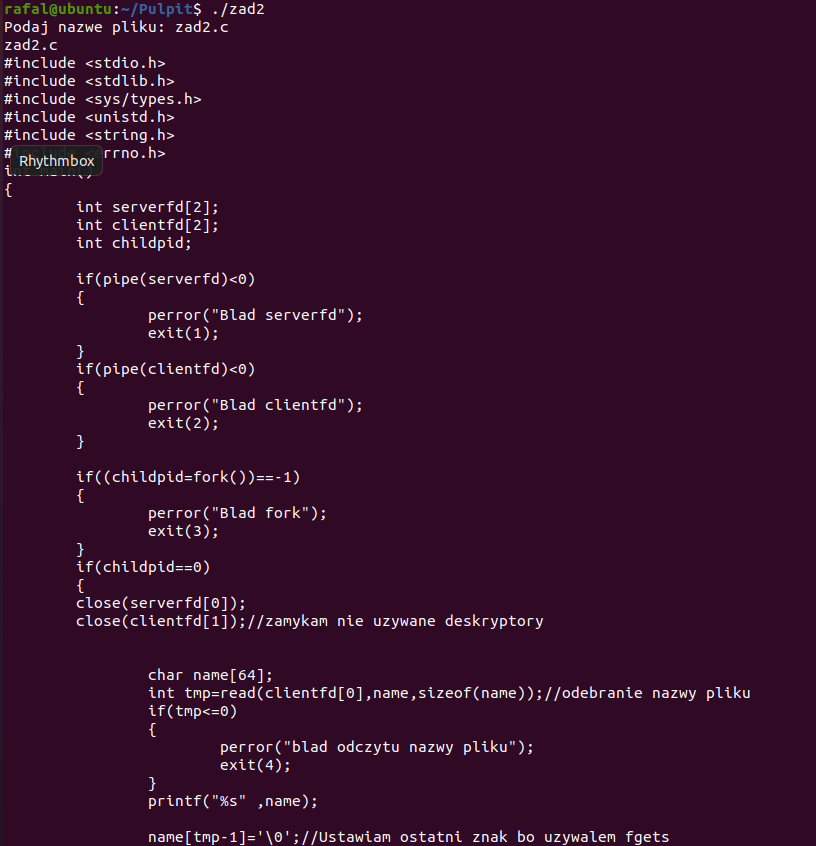
close(serverfd[0]);

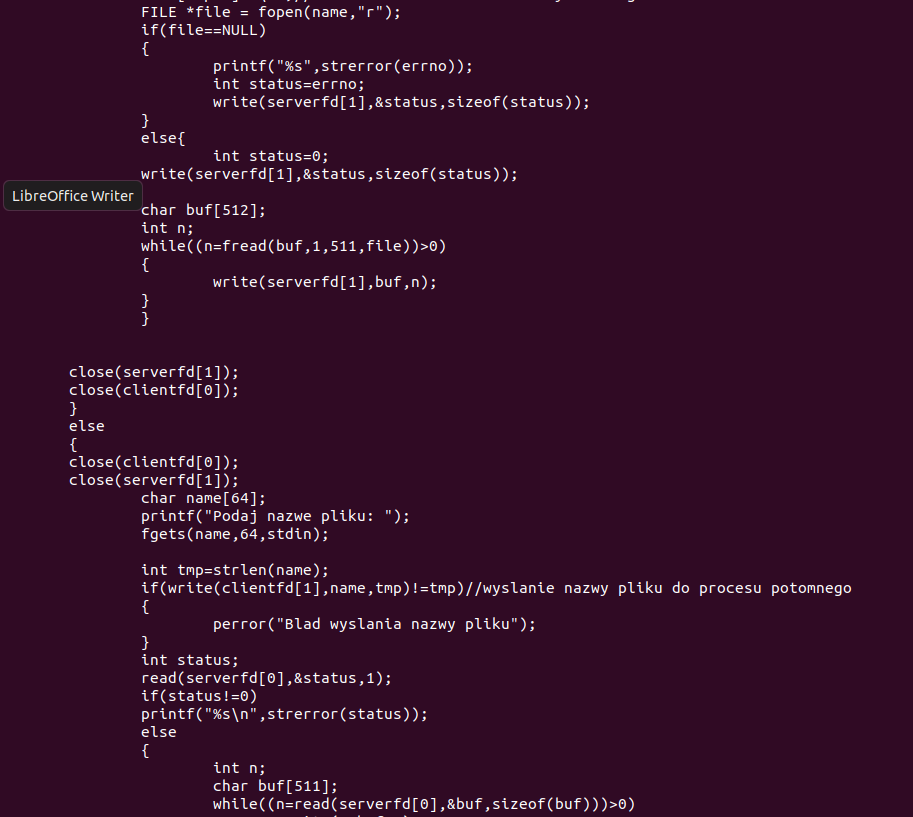
}

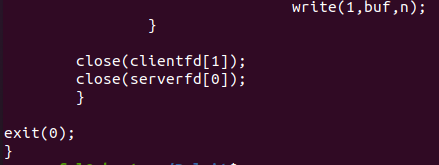
exit(0);

}

Wynik programu:







**Wnioski:**

W trakcie laboratorium udało się wykonać wszystkie podpunkty zadania. Największym problemem okazały się wady funkcji fgets która została zamieniona na funkcję fread. Błędem w programie było przekazanie błędu odczytu pliku do procesu macierzystego. Zostało ono zrealizowane po konwersji do int a powinno być zrealizowane na buforze zawierającym wiadomość string.