YAUHENIYA PADBIAROZSKAYA

308295

Lab 7

Łącza nazwane w API:

1. Komunikacja oparta na kolejkach FIFO:

modyfikacja 1: Serwer po uruchomieniu nie zatrzymywał się na otwieraniu łącza czytania danych od klienta, tylko przechodził do oczekiwania na dane od klienta.

Pierwszy proces otwierający FIFO zawisa na operacji otwarcia, która kończy się, gdy FIFO zostanie otwarte przez inny proces w komplementarnym trybie (O RDONLY/O WRONLY). Można wymusić nieblokowanie, niezawisanie funkcji open() opcją O_NONBLOCK, wtedy proces kontynuuje pracę bez zatrzymywania się.

```
fdsrv = open(fifosrvname, O_RDONLY | O_NONBLOCK);
```

Podczas odczytu z pliku, usuwamy tę opcję, bo proces musi zatrzymać się i czekać na dane:

```
fcntl(fdsrv, F_SETFL, fcntl(fdsrv, F_GETFL, 0) & ~O_NONBLOCK);
…
bread = read(fdsrv, &msg, sizeof(msg));
```

modyfikacja 2: Klient po wysłaniu komunikatu nie zatrzymywał się na otwieraniu łącza czytania danych od serwera, tylko przechodził do oczekiwania na dane od serwera.

Podobnie jak w serwerze, O_NONBLOCK pozwala nie czekać na otwarcie:

```
printf("OK\nOpening client fifo queue \'%s\' for reading...", fifocntname);
fdcnt = open(fifocntname, O_RDONLY | O_NONBLOCK);
```

Ale żeby odczytać dane z serwera proces musi na nie zaczekać:

```
fcntl(fdcnt, F_SETFL, fcntl(fdcnt, F_GETFL, 0) & ~O_NONBLOCK);
bread = read(fdcnt, &msg, sizeof(msg));
```

2. Program 3. zawiera błąd:

```
$ ./file
Server started...
Opening server queue '/psxsrv_psxsrvqueue_Berezka' for reading...OK
Waiting for data...
FAIL!
Error: Message too long
```

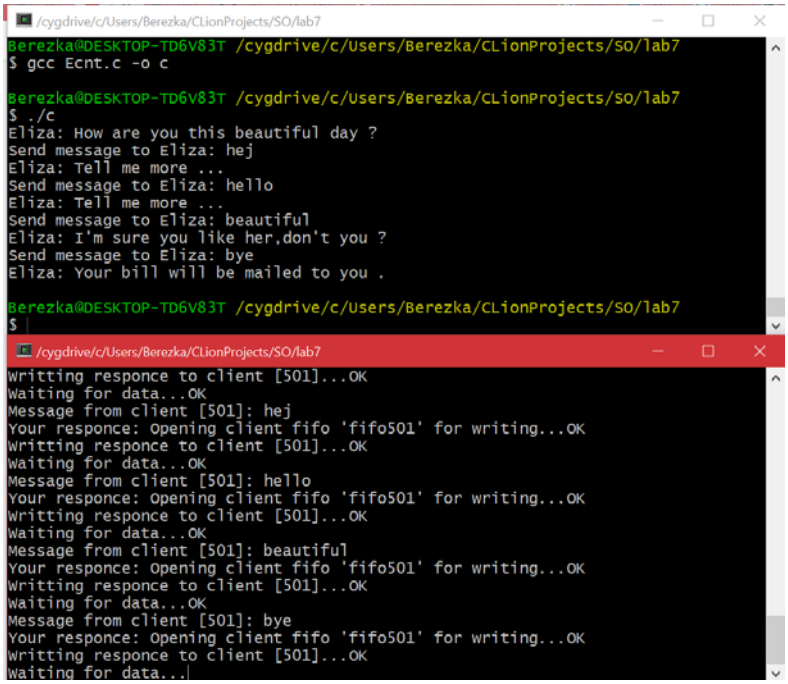
Wartość *errno* jest równa EMSGSIZE, co oznacza, że rozmiar bufora komunikatu jest mniejszy, niż wartość atrybutu *mq_msgsize*, zawartego w strukturze *struct mq_attr*. Możemy sprawdzić te wartość za pomocą funkcji *mq_getattr()* albo wpisując w terminalu polecenie:
`cat /proc/sys/fs/mqueue/msgsize_default`
Po sprawdzeniu otrzymaliśmy maksymalny rozmiar komunikatu 8192 bajtów, a więc wpisujemy tę watość do stałej zdefiniowanej w psx.h:

```
#define MESSAGE_BUF_SIZE 8192
```

```
$./file
Server started...
Opening server queue '/psxsrv_psxsrvqueue_Berezka' for reading...OK
Waiting for data...
```

3. Chatbot

Korzystając z kolejek FIFO, trzeba było przerobić program tak, aby działał w trybie klient-serwer i aby klient mógł rozmawiać z chatbotem.


/cygdrive/c/Users/Berezka/CLionProjects/SO/lab7
Berezka@DESKTOP-TD6V83T /cygdrive/c/Users/Berezka/CLionProjects/SO/lab7
$ gcc Ecnt.c -o c
Berezka@DESKTOP-TD6V83T /cygdrive/c/Users/Berezka/CLionProjects/SO/lab7
$ ./c
Eliza: How are you this beautiful day ?
Send message to Eliza: hej
Eliza: Tell me more ...
Send message to Eliza: hello
Eliza: Tell me more ...
Send message to Eliza: beautiful
Eliza: I'm sure you like her,don't you ?
Send message to Eliza: bye
Eliza: Your bill will be mailed to you .
Berezka@DESKTOP-TD6V83T /cygdrive/c/Users/Berezka/CLionProjects/SO/lab7
$
/cygdrive/c/Users/Berezka/CLionProjects/SO/lab7
Writting responce to client [501]...OK
Waiting for data...OK
Message from client [501]: hej
Your responce: Opening client fifo 'fifo501' for writing...OK
Writting responce to client [501]...OK
Waiting for data...OK
Message from client [501]: hello
Your responce: Opening client fifo 'fifo501' for writing...OK
Writting responce to client [501]...OK
Waiting for data...OK
Message from client [501]: beautiful
Your responce: Opening client fifo 'fifo501' for writing...OK
Writting responce to client [501]...OK
Waiting for data...OK
Message from client [501]: bye
Your responce: Opening client fifo 'fifo501' for writing...OK
Writting responce to client [501]...OK
Waiting for data...