

Sprawozdanie:

7. Zaawansowana komunikacja międzyprocesowa - łącza nazwane, kolejki komunikatów

Dominik Bober 303099

30 kwietnia 2020

1 Łącza nazwane w API

1. W pierwszym programie proszę wprowadzić modyfikację tak aby:

- I Serwer po uruchomieniu nie zatrzymywał się na otwieraniu łącza czytania danych od klienta, tylko przechodził do oczekiwania na dane od klienta.
- II Klient po wysłaniu komunikatu nie zatrzymywał się na otwieraniu łącza czytania danych od serwera, tylko przechodził do oczekiwania na dane od serwera.

Domyślnie kolejka FIFO oczekuje na otwarcie wątku po drugiej stronie, można wymusić by tego nie robiła stosując flagą `O_NONBLOCK`. Jednak by komunikacja w tym programie funkcjonowała poprawnie tworzę nowy dekryptor i ustawiam ręcznie blokadę na pliku korzystając z funkcji `fcntl(2)` i używając flag `F_GETFL` i `F_SETFL`

2. Program 3. zawiera błąd:

- I Na czym on polega?
- II Jak go naprawić?
- III Gdzie można sprawdzić limity ilości i rozmiaru wiadomości?

Problem wynika z tego, że ograniczenia nie są z góry ustalone i zależą od maszyny. Dlatego należy dostosować rozmiar wiadomości tak by byfor nie został przepełniony.

Error: Mssage to long

W moim przypadku domyślny rozmiar wynosi 8192 bajty, jest on zapisany w pliku `msgsize_default`. Żeby mój program działał poprawnie musiałem zmienić wartość `MESSAGE_BUF_SIZE` w pliku `psx.h` oraz w plikach serwera i klienta wartościom `msgsize` nadałem tę samą wartość.

Wszystkie ograniczenia można znaleźć w `/proc/sys/fs/mqueue` są to:

- `msg\default`
- `msg\max`
- `msgsize\default`
- `msgsize\max`
- `queues\max`

2 Zadania

1 Chatbot

Eliza to jeden z pierwszych programów typu chatbot. Oryginalny program, który symuluje psychoanalityka, jest dostępny np. w implementacji edytora GNU Emacs.

Do celów tego laboratorium możemy się posłużyć jedną z dostępnych w Internecie implementacji w języku C, np. `eliza.c` (program pobrany ze strony: http://www.hackchina.com/en/r/3072/ELIZA.C_.html).

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

Zedytowałem i połączyłem pliki `eliza.c` z plikiem `srvfifo.c` oraz zmodyfikowałem plik `cntfifo.c`.

```
dominik@mimik: ~/Projects/c_so_projects/lab7/eliza_server
File Edit View Search Terminal Help
dominik@mimik:~/Projects/c_so_projects/lab7/eliza_server$ ./eliza_cntfifo
Client [22888] started...
Creating client fifo queue 'fifo22888'...OK
Opening server Eliza 'fifo_dominik' for writing...OK
Send message: witam

Eliza: Tell me more ...

Send message: murder !!!!

Eliza: I don't like killing .

Send message: i like

Eliza: Why do you like that?

Send message: it is fun

Eliza: Tell me more ...

Send message: _
```

Rysunek 1: Klient chatbota Eliza

```
dominik@mimik: ~/Projects/c_so_projects/lab7/eliza_server
File Edit View Search Terminal Help
dominik@mimik:~/Projects/c_so_projects/lab7/eliza_server$ ./eliza
Server started...
Creating server fifo queue 'fifo_dominik'...OK
Opening server Eliza 'fifo_dominik' for reading...OK
Waiting for data...OK
Message from client [22888]: witam
Your response: Opening client fifo 'fifo22888' for writing...OK
Writing response to client [22888]...OK
Waiting for data...OK
Message from client [22888]: murder !!!!
Your response: Opening client fifo 'fifo22888' for writing...OK
Writing response to client [22888]...OK
Waiting for data...OK
Message from client [22888]: i like
Your response: Opening client fifo 'fifo22888' for writing...OK
Writing response to client [22888]...OK
Waiting for data...OK
Message from client [22888]: it is fun
Your response: Opening client fifo 'fifo22888' for writing...OK
Writing response to client [22888]...OK
Waiting for data..._
```

Rysunek 2: Server chatbota Eliza

2 Zliczanie głosów

W czasie wyborów głosy z poszczególnych okręgów zliczane są przez Państwową Komisję Wyborczą. Napisz aplikację, która zlicza głosy z poszczególnych okręgów posługując się kolejką komunikatów (System V albo POSIX) do ich przesyłania.

Założenia:

- Okręgi to aplikacje klienckie generujące losowo pojedynczą liczbę głosów i wysyłające ją do serwera.
- Serwer PKW zlicza liczby przesłane do niego.
- Serwer rejestruje identyfikatory okręgów wyborczych, które się z nim komunikowały.
- Po każdorazowym otrzymaniu informacji od okręgu wyborczego, serwer przesyła do każdego zarejestrowanego okręgu aktualne dane o frekwencji wyborczej.
- Serwer PKW przechowuje informację o maksymalnej liczbie głosów jakie mogły zostać oddane (np. `MAX_VOTES`).
- Ilość okręgów wyborczych jest zmienna i definiowana przy pomocy zmiennej (np. `MAX_CLIENTS`).
- Klient losując liczbę głosów losuje liczbę z zakresu: $[0 - \text{MAX_VOTES}/\text{MAX_CLIENTS}]$.

Następnie rozbuduj program o zliczanie głosów na ustalone komitety wyborcze.

Założenia:

- Liczba komitetów wyborczych jest definiowana przez zmienną (np. `MAX_COMMITTIES`).
- Informacje o liczbie głosów dla poszczególnych komitetów wyborczych są przesyłane w zależności od typu użytej kolejki komunikatów:
 - Typ System V: Komitet wyborczy jest identyfikowany przy pomocy pola typu komunikatu.
 - Typ POSIX: Komitet wyborczy jest identyfikowany przy pomocy priorytetu komunikatu.
- Na podstawie otrzymanych informacji serwer wyświetla następujące informacje:
 - Frekwencja na wyborach.
 - Który komitet wygrał wybory - nie ma progu minimalnej liczby głosów.

Zdecydowałem się użyć kolejki System V, dlatego zmodyfikowałem pliki `cntsv.c`, `srvsv.c` oraz `sv.h`.

Ograniczenia wyglądają następująco:

`MAX_VOTES 100`

`MAX_CLIENTS 4`

`MAX_COMMITTIES 4`

```
dominik@mimik: ~/Projects/c_so_projects/lab7/elections_server
File Edit View Search Terminal Help
dominik@mimik:~/Projects/c_so_projects/lab7/elections_server$ ./elections_cntsv
Client started...
Creating server key file name:
    Path: /tmp/srvsv_dominik
Getting server queue key...OK
Getting server queue identifier...OK
Getting client queue identifier...OK
Your message
    Votes: 5
    Committee: 1
Sending to server...OK
Your message
    Votes: 2
    Committee: 2
Sending to server...OK
Your message
    Votes: 5
    Committee: 3
Sending to server...OK
Your message
    Votes: 2
    Committee: 4
Sending to server...OK
Waiting for data...
Current attendance: 14
Waiting for data...
Current attendance: 23
Waiting for data...
Current attendance: 33
Waiting for data...
Current attendance: 50
Waiting for data...
dominik@mimik:~/Projects/c_so_projects/lab7/elections_server$ _
```

Rysunek 3: Jeden z klientów programu liczącego głosy

```
dominik@mimik: ~/Projects/c_so_projects/lab7/elections_server
File Edit View Search Terminal Help
dominik@mimik:~/Projects/c_so_projects/lab7$ cd elections_server/
dominik@mimik:~/Projects/c_so_projects/lab7/elections_server$ ./elections_srvsv Server started...
Creating server key file:
  Path: /tmp/srvsv_dominik
  Opening server key file...OK
Getting queue key...OK
Getting server queue identifier...the queue already exists, removing it...OK
Data from client [32830]
Committee: 1
Votes: 5
Committee: 2
Votes: 2
Committee: 3
Votes: 5
Committee: 4
Votes: 2
Writting responce to clients...
Data from client [32831]
Committee: 1
Votes: 2
Committee: 2
Votes: 4
Committee: 3
Votes: 3
Committee: 4
Votes: 0
Writting responce to clients...
Data from client [65536]
Committee: 1
Votes: 0
Committee: 2
Votes: 5
Committee: 3
Votes: 2
Committee: 4
Votes: 3
Writting responce to clients...
Data from client [65539]
Committee: 1
Votes: 4
Committee: 2
Votes: 5
Committee: 3
Votes: 4
Committee: 4
Votes: 4
Writting responce to clients...
Results:
  Committee 1 : 11
  Committee 2 : 16
  Committee 3 : 14
  Committee 4 : 9
Attendance: 50
Winning committee: 2
dominik@mimik:~/Projects/c_so_projects/lab7/elections_server$ _
```

Rysunek 4: Server programu liczącego głosy