

Maryna Lukachyk

Laboratorium 7

Zaawansowana komunikacja międzyprocesowa - łącza nazwane, kolejki komunikatów

Łącza nazwane w API

Potoki nazwane (FIFO)

- istnieją trwale w systemie plików (mknod potok p)
- wymagają otwarcia O_RDONLY lub O_WRONLY
- zavisają na prośbie otwarcia nieotwartego potoku, możliwe jest tylko jednoczesne

otwarcie do odczytu i zapisu, przez dwa różne procesy

Istnieje pięć funkcji systemowych do obsługi kolejek FIFO:

- mknod - tworzy nowe łącza
- open - otwiera łącze
- write - zapisuje dane do łącza
- read - czyta dane z łącza
- close - zamyka deskryptor związany z łączem
- unlink - zamyka łącze

```
struct pipe_inode_info {
struct wait_queue * wait; /*kolejka procesow czekajacych na pisanie, czytanie lub procesow bedacych w trakcie otwierania lacza i oczekujacych na proces po drugiej stronie lacza*/
char * base; /*wskaznik do bufora z danymi*/
unsigned int start; /*poczatek danych w buforze*/
unsigned int len; /*ilosc danych*/
unsigned int lock; /*zmienna sluzaca do synchronizacji dostepu do struktur przechowujacych dane. Rozna od zera oznacza, ze istnieja procesy, ktore pisza lub czytaja z lacza*/
unsigned int rd_openers; /*ilosc procesow bedacych w trakcie otwierania lacza do pisania*/
unsigned int wr_openers; /*ilosc procesow bedacych w trakcie otwierania lacza do czytania*/
unsigned int readers; /*ilosc procesow czytajacych z lacza*/
unsigned int writers; /*ilosc procesow piszacych do lacza*/
};
```

Funkcja open()

-otwiera łącze.

DEFINICJA: int open(char* nazwa_lacza, int flagi)

WYNIK:

0 w przypadku sukcesu

-1, gdy błąd errno :

ENXIO - otwieranie łącza do pisania, gdy nie ma procesów czytających;
brakuje pamięci w jądrze;
sygnał;

ENOMEM -

EINTR - w czasie wykonania funkcji proces otrzymał

Pierwszym argumentem funkcji jest nazwa łącza. Drugi argument określa sposób dostępu do łącza.

Argument flagi może przyjmować następujące wartości:

-O_WRONLY - otwieranie łącza tylko do pisania

-O_RDONLY - otwieranie łącza tylko do czytania

-O_RDWR - otwieranie łącza do pisania i czytania

-O_NDELAY - określa czy operacje, których wykonanie nie może być natychmiast wykonane mają być blokowane. Flaga ustawiona - brak blokowania.

Funkcja read()

	z O_NDELAY	bez O_NDELAY
danych do odczytania jest mniej niż danych w łaczu	odczytana liczba bajtów = count	odczytana liczba bajtów = count
puste łącze	sa pisarze - EAGAIN, nie ma ich - 0	sa pisarze - czeka na dane, nie ma ich - 0
danych do odczytania jest więcej niż danych w łaczu	odczytana liczba bajtów = ilości danych w łaczu	odczytana liczba bajtów = ilość danych w łaczu

Należy zauważyć, że funkcja read nie może przeczytać więcej danych niż rozmiar łącza. Dlatego też, jeżeli chcemy odczytać więcej bajtów niż łącze może pomieścić, funkcje read trzeba wywołać kilkakrotnie. To jednak może powodować, że nie otrzymamy spójnych danych.

close()

- zamyka deskryptor łącza.

DEFINICJA: int close(int fd)

WYNIK: 0 gdy sukces; -1 gdy błąd

unlink()

-usuwa łącze nazwane

DEFINICJA: int unlink(char* nazwa_lacza)

WYNIK: 0 gdy sukces ; -1 gdy błąd

Jeżeli w czasie wywołania są procesy, które z łącza korzystają to zostaje usunięta tylko nazwa łącza z dysku, tak że już nowe procesy nie mogą z niego korzystać. Łącze zostanie usunięte, gdy wszystkie procesy korzystające z niego zamkną deskryptory związane z tym łączem.

System V

W oparciu o licencjonowany kod SysV powstało wiele komercyjnych wersji Uniksa między innymi AIX, A/UX, HP-UX oraz SCO Xenix.

Ostatnią wersją **System V Release 4** była **SVR4.2MP** wydana pod koniec 1993 roku przez Unix System Laboratories (dawne Bell Labs) należące już wtedy do Novella, który rozprowadzał ją jako UnixWare 2. Dziś prawa do kodu źródłowego System V należą do firmy Novell.

Urządzenia komunikacyjne:

- kolejki komunikatów: koniec–koniec
- pamięć współdzielona: wielu–wielu
- semafore: liczby całkowite

Nie są deskryptorami plików i nie wykonuje się na nich operacji I/O standardowymi funkcjami read/write - każde urządzenie ma swój specyficzny zbiór operacji I/O.

Identyfikatory urządzeń System V IPC (typu int) są globalne w systemie, inaczej niż deskryptory plików(i nie są kolejno generowanymi małymi liczbami). Oznacza to, że jeden proces może użyć identyfikatora utworzonego przez inny proces.

Wynika stąd możliwość “wkleszczania się”, celowego lub nie, w komunikację prowadzoną przez inne procesy.

Kolejki komunikatów POSIX

Własności:

- Kolejka może być otwarta w jednym z trybów: **O_RDONLY**, **O_WRONLY**, **O_RDWR**
- kolejki przekazują komunikaty jako całe jednostki.
- zawsze odbierany jest najstarszy komunikat o najwyższym priorytecie.
- kolejki POSIX posiadają zdolność blokowania procesu w oczekiwaniu na komunikat gdy kolejka jest pusta, lub natychmiastowego powrotu z kodem sygnalizującym brak komunikatu
- powiadamianie asynchroniczne (uruchomienie określonej funkcji jako nowego wątku)

Wniosek do zadania 3:

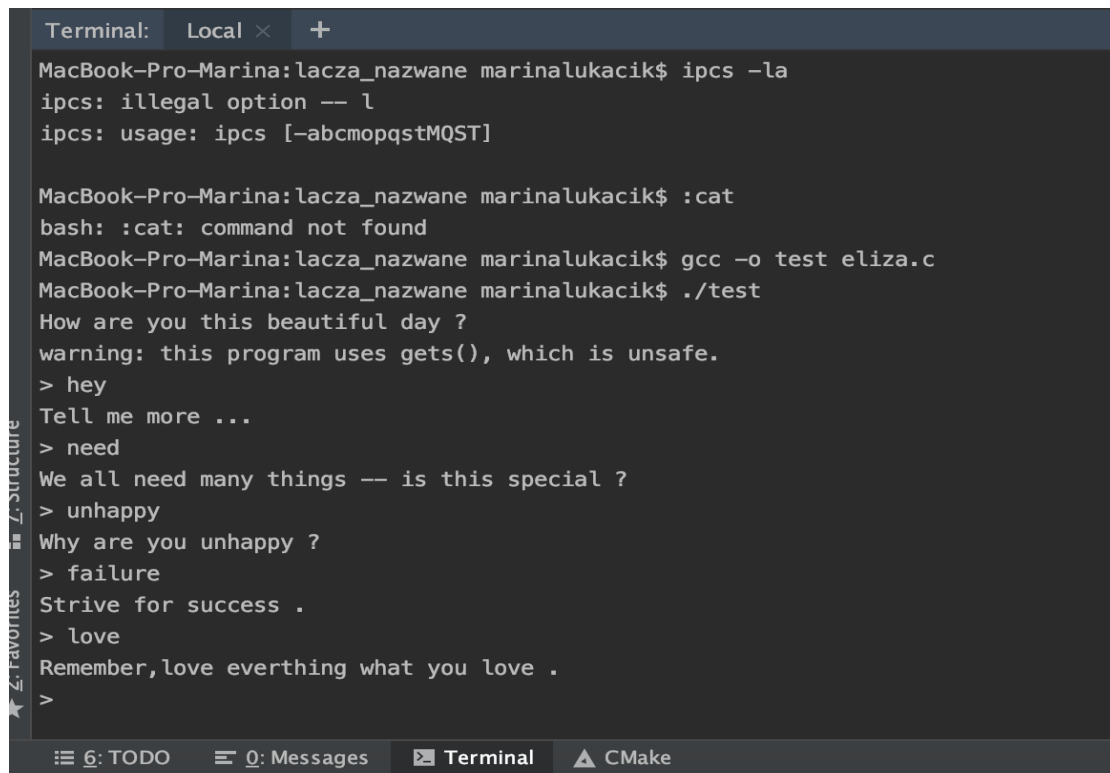
Po sprawdzeniu otrzymywanego errno okazuje się że dostajemy błąd EMSGSIZE, mówi o tym że rozmiar bufora wiadomości jest mniejszy niż zawarty w mq_msgsize; Tworząc kolejkę podaliśmy NULL(został użyty domyślny rozmiar komunikatu).

Odczytujemy go za pomocą funkcji `cat /proc/sys/fs/mqueue/msg_default`.

Nasz bufor musi mieć rozmiar min tego komunikatu żeby zaczął działać poprawnie.

```
----- Limity komunikatów -----  
maksymalna liczba kolejek w systemie = 32000  
maksymalny rozmiar komunikatu (bajtów) = 8192  
domyślny maksymalny rozmiar kolejki (bajtów) = 16384
```

1 Chatbot



```
Terminal: Local × +  
MacBook-Pro-Marina:lacza_nazwane marinalukacik$ ipcs -la  
ipcs: illegal option -- l  
ipcs: usage: ipcs [-abcmopqstMQST]  
  
MacBook-Pro-Marina:lacza_nazwane marinalukacik$ :cat  
bash: :cat: command not found  
MacBook-Pro-Marina:lacza_nazwane marinalukacik$ gcc -o test eliza.c  
MacBook-Pro-Marina:lacza_nazwane marinalukacik$ ./test  
How are you this beautiful day ?  
warning: this program uses gets(), which is unsafe.  
> hey  
Tell me more ...  
> need  
We all need many things -- is this special ?  
> unhappy  
Why are you unhappy ?  
> failure  
Strive for success .  
> love  
Remember,love everthing what you love .  
>
```