Systemy Operacyjne 2017/18

Zadania - Zestaw 6

IPC - kolejki komunikatów

Przydatne funkcje:

System V:

```
<sys/msg.h> <sys/ipc.h> - msgget, msgctl, msgsnd, msgrcv, ftok
```

POSIX:

```
<mqueue.h> - mq_open, mq_send, mq_receive, mq_getattr, mq_setattr, mq_close, mq_unlink, mq_notify
```

Zadanie 1. System V (50%)

Napisz prosty program typu klient-serwer, w którym komunikacja zrealizowana jest za pomocą kolejek komunikatów - jedna, na zlecenia klientów dla serwera, druga, prywatna, na odpowiedzi.
Serwer po uruchomieniu tworzy nową kolejkę komunikatów systemu V. Za pomocą tej kolejki klienci będą wysyłać komunikaty do serwera. Wysyłane zlecenia mają zawierać rodzaj zlecenia jako rodzaj komunikatu oraz informację od którego klienta zostały wysłane (PID procesu klienta), w odpowiedzi rodzajem komunikatu ma być informacja identyfikująca czekającego na nią klienta.
Klient bezpośrednio po uruchomieniu tworzy kolejkę z unikalnym kluczem `IPC` i wysyła jej klucz komunikatem do serwera. Po otrzymaniu takiego komunikatu, serwer otwiera kolejkę klienta, przydziela klientowi identyfikator (np. numer w kolejności zgłoszeń) i odsyła ten identyfikator do klienta (komunikacja w kierunku `serwer->klient` odbywa się za pomocą kolejki klienta). Po otrzymaniu identyfikatora, klient rozpoczyna wysyłanie zleceń do serwera (w pętli), zlecenia są czytane ze standardowego wyjścia w postaci typ_komunikatu albo z pliku tekstowego w którym w każdej linii znajduje się jeden komunikat.

Rodzaje zleceń

- Usługa lustra (MIRROR):
  Klient wysyła ciąg znaków. Serwer odsyła ten sam ciąg z powrotem w odwrotnej kolejności. Klient po odebraniu wysyła go na standardowe wyjście.
- Usługa kalkulatora (CALC):
  Klient wysyła proste działanie arytmetyczne (+ - * /) Serwer oblicza i odsyła wynik. Klient po odebraniu wysyła go na standardowe wyjście.
  Działania można przesyłać w postaci np CALC 2+3, CALC 3*4 albo ADD 1 2, MUL 3 4, SUB 5 4, DIV 4 3 - wariant do wyboru.
- Usługa czasu (TIME):
  Po odebraniu takiego zlecenia serwer wysyła do klienta datę i godzinę w postaci łańcucha znaków. Klient po odebraniu informacji wysyła ją na standardowe wyjście.
- Nakaz zakończenia (END):
  Po odebraniu takiego zlecenia serwer zakończy działanie, jak tylko opróżni się kolejka zleceń (zostaną wykonane wszystkie pozostałe zlecenia). Klient nie czeka na odpowiedź.

Poszczególne rodzaje komunikatów należy identyfikować za pomocą typów komunikatów systemu V. Klucze dla kolejek mają być generowane na podstawie ścieżki **`$HOME`**. Małe liczby do wygenerowania kluczy oraz rodzaje komunikatów mają być zdefiniowane we wspólnym pliku nagłówkowym. Dla uproszczenia można założyć, że długość komunikatu (lub maksymalna długość łańcucha znaków przy usłudze lustra) jest ograniczona pewną stałą (jej definicja powinna znaleźć się w pliku nagłówkowym).
Klient i serwer należy napisać w postaci osobnych programów (nie korzystamy z funkcji `fork`). Serwer musi być w stanie pracować z wieloma klientami naraz. Przed zakończeniem pracy każdy proces powinien usunąć kolejkę którą utworzył (patrz `IPC_RMID` oraz funkcja `atexit`). Dla uproszczenia można przyjąć, że serwer przechowuje informacje o klientach w statycznej tablicy (rozmiar tablicy ogranicza liczbę klientów, którzy mogą się zgłosić do serwera).
Nalezy obsłużyć przerwanie działania serwera lub klienta za pomocą CTRL+C. W przypadku klienta, można zdefiniować dodatkowy typ komunikatu STOP, który wysyła klient, kiedy kończy pracę, aby serwer mógł skasowac jego kolejkę.

Zadanie 2. POSIX (50%)

Zrealizuj zadanie analogiczne do Zadania 1, wykorzystując kolejki komunikatów `POSIX`.
Kolejka klienta powinna mieć losową nazwę zgodną z wymaganiami stawianymi przez `POSIX`. Na typ komunikatu można zarezerwować pierwszy bajt jego treści. Przed zakończeniem pracy klient wysyła do serwera komunikat informujący, że serwer powinien zamknąć po swojej stronie kolejkę klienta. Następnie klient zamyka i usuwa swoją kolejkę. Serwer przed zakończeniem pracy zamyka wszystkie otwarte kolejki i usuwa kolejkę, którą utworzył.