

Programowanie współbieżne

Laboratorium 2

2020/2021

Zadanie 1 (C/C++/Python):

Utworzyć w języku C parę programów producent-konsument korzystającą z pliku jako wspólnego bufora (producent czyta znaki z klawiatury i dopisuje do pliku, konsument pobiera z pliku i wyświetla). Tekst wprowadzać i odczytywać całymi liniami. Przyjąć, że znacznikiem końca przekazywanego tekstu jest kod Esc (27 w ASCII).

UWAGA: operacje na pliku powinniśmy realizować nie przy użyciu funkcji ze standardu ANSI C, ale przy użyciu rodzimych funkcji uniksowych, które zapewniają nieprzerywalne wykonania operacji zapisu i odczytu.

Rekomendowane funkcje to:

`open()`, `close()`, `unlink()`, `read()` i `write()`

(skojarzone z plikami nagłówkowymi `fcntl.h` i `unistd.h` w przypadku języka C)

`open()`, `close()`, `remove()`, `read()` i `write()`

(skojarzone z plikami nagłówkowymi `fstream`, `cstdio` i `unistd.h` w przypadku języka C++).

`open()`, `close()`, `remove()`, `read()` i `write()`

(skojarzone z modułem `os` w przypadku języka Python).

Zadanie 2 (PODLEGA ODBIOROWI - CZAS 2 ZAJĘCIA - C/C++/Python):

Rozbudować powyższą parę programów tak, aby otrzymać parę klient-serwer składającą się na prosty komunikator tekstowy. Plikom będącym buforami klienta i serwera przydzielić odpowiednie prawa dostępu.

Serwer działa w pętli nieskończonej, klient tylko jednorazowo (pytanie - odpowiedź). Program klienta jako parametr wywołania otrzymuje nazwę pliku z danymi wejściowymi dla serwera (tylko nazwa pliku, jeżeli klienta i serwera uruchamiamy w tym samym katalogu, w przeciwnym razie trzeba podać całą ścieżkę do pliku). Serwer jest jeden, klientów może być wielu, działających równocześnie.

Klient przesyła do bufora serwera:

- nazwę swojego pliku do przesłania odpowiedzi (każdy klient ma swój osobny plik);
- linie swojego tekstu wprowadzonego przez użytkownika (dowolnie dużo, można ustalić górny próg);
- znacznik końca tekstu (kod Esc).

Serwer po przeczytaniu i wyświetleniu całego tekstu od klienta odsyła klientowi:

- kolejne linie swojego tekstu (wprowadzone ręcznie przez właściciela serwera);
- znacznik końca tekstu (kod Esc).

Dostęp do bufora serwera powinien być synchronizowany plikiem zamkowym (lockfile) tworzonym przez klienta w katalogu roboczym serwera i usuwanym przez serwer po udzieleniu odpowiedzi klientowi. Jeżeli serwer czasowo nie jest dostępny, klient wyświetla w pętli co kilka sekund informację: Serwer zajęty, proszę czekać. Wykorzystać w programie klienta następującą konstrukcję (lub jej odpowiednik, jeżeli używamy innych bibliotek operujących na plikach):

```
...

while (open("sciezka_dostepu/lockfile",O_CREAT|O_EXCL,0)==-1) {
    printf("Serwer zajety, prosze czekac\n");
    sleep( ... );
}

...
```

Po udzieleniu odpowiedzi serwer usuwa utworzony przez klienta plik lockfile.

UWAGA 1: Żeby przetestować sytuację, że serwer może być zajęty, należy utworzyć dwóch klientów: jeden wysyła do serwera dane, a drugi, jeszcze przed odpowiedzią serwera do pierwszego też próbuje się skomunikować z serwerem.

UWAGA 2: programy poleca się uruchamiać bezpośrednio na sigmie, aby uniknąć opóźnień związanych z działaniem sieciowego systemu plików. W przypadku wystąpienia problemów przenieść bufora do katalogu **/var/sandbox** lub **/tmp** i tam testować programy.

TERMIN: 17.03.2021