

Projekt

Treść

Opracować zestaw programów typu *producent - konsument* realizujących następujący schemat synchronicznej komunikacji międzyprocesowej:

- *Proces 1*: czyta dane ze standardowego strumienia wejściowego i przekazuje je w niezmienionej formie do *procesu 2*.
- *Proces 2*: pobiera dane przesłane przez *proces 1*. Oblicza liczbę odczytanych bajtów, wyświetla ją na standardowym strumieniu diagnostycznym, a następnie przekazuje odebrane dane mechanizmem komunikacji do *procesu 3*.
- *Proces 3*: pobiera dane wyprodukowane przez *proces 2* i umieszcza je w standardowym strumieniu wyjściowym.

Programy powinny przed przekazaniem kolejnej porcji danych oczekiwać na odczytanie wcześniejszych przez kolejny proces.

Wszystkie trzy procesy powinny być powoływane automatycznie z jednego procesu inicjującego. Po powołaniu procesów potomnych proces inicjujący bezzwłocznie powinien zakończyć działanie.

Należy zaimplementować mechanizm asynchronicznej kontroli procesów - po otrzymaniu przez dowolny proces sygnału:

- Si1 procesy powinny zakończyć swoje działanie,
- Si2 procesy powinny wstrzymać swoje działanie,
- Si3 procesy powinny wznowić swoje działanie.

Aby zrealizować powyższe operacje proces powinien przekazać informacje o odebranych sygnałach pozostałym procesom poprzez mechanizm łącz komunikacyjnych (K3) oraz wysłać do nich sygnał Si4 informujący o oczekującym komunikacie (komunikatach).

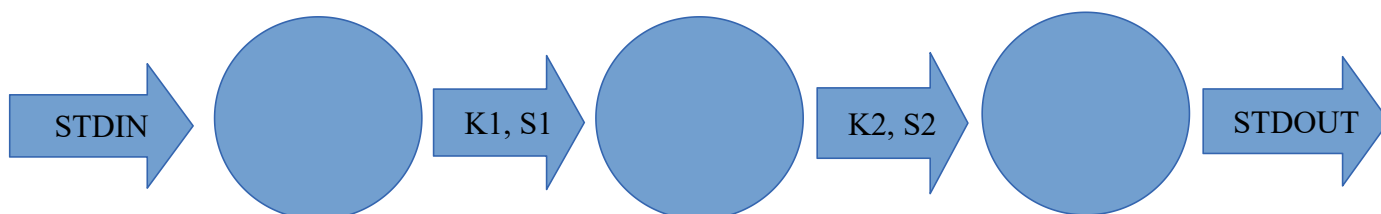
Jako sygnały oznaczone symbolami Si1-Si4 można samodzielnie wybrać sygnały spośród dostępnych w systemie (np. SIGUSR1, SIGUSR2, SIGINT, SIGCONT).

STDIN, STDOUT – standardowe wejście/wyjście

K1, K2, K3 – mechanizmy komunikacyjne

S1, S2 – mechanizmy synchronizacyjne (jeśli są potrzebne)

Mechanizmy komunikacji dostępne są w pliku z listą.



Przykładowe uruchomienia programu:

- w trybie interaktywnym

```
./projekt  
cat | ./projekt  
tee | ./projekt
```
- przekazywanie danych z dowolnego pliku

```
cat plik | ./projekt  
./projekt < plik  
cat plik | ./projekt | tee nowy_plik  
./projekt < plik > nowy_plik  
(weryfikacja pliku)  
diff plik nowy_plik  
colordiff plik nowy_plik  
xxd -p jeden_z_plików
```
- przekazywanie filmu

```
./projekt < film.mp4 > nowy_film.mp4 # i odtworzenie  
cat film.mp4 | ./projekt | vlc -
```

Punktacja

Elementy wymagane (15 pkt):

- bezawaryjna praca z plikiem /dev/urandom przekierowanym na standardowy strumień wyjściowy,
- poprawne przetworzenie zadanego pliku – lektury szkolnej lub pliku binarnego,
- odpowiedź na połowę pytań zadanych przez prowadzącego.

Dwa pierwsze punkty można zastąpić przesyłaniem filmu o długości co najmniej 5 minut.

Poprawna obsługa sygnałów (15 pkt):

- wstrzymywania,
- wznowiania,
- kończenia pracy

po wysłaniu sygnału do dowolnego procesu.

Poprawne kończenie pracy – program nie powinien pozostawiać po sobie pozostałości (5 pkt).

Lp.	K1	K2	K3
1	pamięć współdzielona	potok	kolejka komunikatów
2	pamięć współdzielona	kolejka FIFO	kolejka komunikatów
3	potok	pamięć współdzielona	kolejka komunikatów
4	kolejka FIFO	pamięć współdzielona	kolejka komunikatów
5	potok	pamięć współdzielona	kolejka komunikatów
6	kolejka FIFO	pamięć współdzielona	kolejka komunikatów
7	kolejka komunikatów	pamięć współdzielona	potok
8	kolejka komunikatów	pamięć współdzielona	kolejka FIFO
9	kolejka komunikatów	potok	pamięć współdzielona
10	kolejka FIFO	kolejka komunikatów	pamięć współdzielona
11	kolejka komunikatów	pamięć współdzielona	potok
12	kolejka komunikatów	kolejka FIFO	pamięć współdzielona
13	potok	kolejka komunikatów	pamięć współdzielona
14	kolejka komunikatów	kolejka FIFO	pamięć współdzielona
15	potok	kolejka komunikatów	pamięć współdzielona
16	kolejka FIFO	kolejka komunikatów	pamięć współdzielona
17	kolejka komunikatów	potok	pamięć współdzielona
18	kolejka komunikatów	pamięć współdzielona	kolejka FIFO
19	pamięć współdzielona	kolejka komunikatów	potok
20	pamięć współdzielona	kolejka komunikatów	kolejka FIFO