

## Projekt

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

- *Proces 1*: czyta dane ze standardowego strumienia wejściowego i przekazuje je do *procesu 2* poprzez mechanizm komunikacyjny **K1**.
- *Proces 2*: pobiera dane przesłane przez *proces 1* i przekazuje do *procesu 3* poprzez mechanizm komunikacyjny **K2**.
- *Proces 3*: pobiera dane wyprodukowane przez *proces 2* i wypisuje je na standardowym strumieniu diagnostycznym. Jednostki danych powinny zostać wyprowadzone po 15 w pojedynczym wierszu i oddzielone spacjami.

## Wytyczne:

1. Program ma umożliwiać uruchomienie:
  - a. w trybie interaktywnym – operator wprowadza dane z klawiatury,
  - b. w trybie odczytu danych z określonego pliku,
  - c. w trybie odczytu danych z pliku `/dev/urandom`.
2. Wszystkie trzy procesy powinny być powoływane automatycznie z jednego procesu inicjującego. Po powołaniu procesów potomnych proces inicjujący wstrzymuje pracę. Proces inicjujący wznowia pracę w momencie kończenia pracy programu (o czym niżej), jego zadaniem jest „posprzątać” po programie przed zakończeniem działania.
3. Jeden ze wspomnianych procesów (wskazany w przydzielonej wersji zadania) konwertuje otrzymane dane do postaci heksadecymalnej.
4. Należy zaimplementować mechanizm asynchronicznego przekazywania informacji pomiędzy operatorem a procesami oraz pomiędzy procesami. Wykorzystać do tego dostępny mechanizm sygnałów.
  - a. Operator może wysłać do dowolnego procesu sygnał zakończenia działania (S1), sygnał wstrzymania działania (S2) i sygnał wznowienia działania (S3). Sygnał S2 powoduje wstrzymanie synchronicznej wymiany danych pomiędzy procesami. Sygnał S3 powoduje wznowienie tej wymiany. Sygnał S1 powoduje zakończenie działania oraz zwolnienie wszelkich wykorzystywanych przez procesy zasobów (zasoby zwalnia proces macierzysty).
  - b. Każdy z sygnałów przekazywany jest przez operatora tylko do jednego, dowolnego procesu. O tym, do którego procesu wysłać sygnał, decyduje operator, a nie programista. Każdy z sygnałów operator może wysłać do innego procesu. Mimo, że operator kieruje sygnał do jednego procesu, to pożądane przez operatora działanie musi zostać zrealizowane przez wszystkie trzy procesy. W związku z tym, proces

odbierający sygnał od operatora musi powiadomić o przyjętym żądaniu pozostałe dwa procesy. Powinien wobec tego przekazać do nich odpowiedni sygnał informując o tym jakiego działania wymaga operator. Procesy odbierające sygnał, powinny zachować się adekwatnie do otrzymanego sygnału. Wszystkie trzy procesy powinny zareagować zgodnie z żądaniem operatora.

- c. Sygnały oznaczone w opisie zadania symbolami S1 ÷ S3 należy wybrać samodzielnie spośród dostępnych w systemie (np. SIGUSR1, SIGUSR2, SIGINT, SIGCONT).
5. Operator może wysłać do dowolnego procesu dodatkowy sygnał, nakazujący zaprzestania konwersji danych na postać heksadecymalną – dane wyjściowe powinny być w takim przypadku dokładnie takie, jak dane wejściowe. Kolejne wysłanie tego samego sygnału powinno ponownie włączyć konwersję.
  6. Wysyłanie sygnałów do procesów powinno odbywać się z wykorzystaniem dodatkowego programu napisanego w języku C. Program ten powinien umożliwiać (przy pomocy menu użytkownika) wybór sygnału oraz procesu do którego ten sygnał ma zostać wysłany.
  7. Mechanizmy komunikacji: K1 i K2, a także informacja o tym, który z procesów ma wykonywać zadanie konwersji na postać heksadecymalną, są podane w pliku lista\_mechanizmow.pdf – każdy student ma przypisane inne mechanizmy komunikacji. Niektóre mechanizmy komunikacyjne wymagają użycia mechanizmów synchronizacji, np. semaforów.

## **Punktacja:**

Absolutne minimum, które pozwala uznać projekt za oddany (10pkt.):

- uruchomienie w trybie interaktywnym,
- bezawaryjne działanie przez 5 minut w trybie danych z pliku /dev/urandom,
- poprawne przetworzenie podanego przeze mnie pliku zawierającego różne znaki m.in.: znaki niedrukowalne, duże/małe litery (w tym polskie), cyfry, znaki specjalne widoczne na klawiaturze,
- konwersja na postać heksadecymalną odbywa się we właściwym z 3 procesów,
- zastosowane są mechanizmy komunikacji zgodne z otrzymaną wersją zadania.

**Jeśli ten punkt nie jest spełniony nie ma mowy o „zaliczeniu” zadania nawet jeśli pozostałe rzeczy działałyby poprawnie.**

Poprawna obsługa sygnałów (16pkt.):

- program wstrzymuje pracę po wysłaniu sygnału do dowolnego procesu,
- program wznowia pracę po wysłaniu sygnału do dowolnego procesu,
- program kończy pracę po wysłaniu sygnału do dowolnego procesu.
- program włącza lub wyłącza konwersję na postać heksadecymalną po wysłaniu sygnału do dowolnego procesu.

Obsługa sygnałów zewnętrznym programem (4pkt.):

- wysyłanie sygnałów do procesów powinno odbywać się z wykorzystaniem dodatkowego programu napisanego w języku C. Program ten powinien umożliwiać (przy pomocy menu użytkownika) wybór sygnału oraz procesu do którego ten sygnał ma zostać wysłany.

Poprawne kończenie pracy (5pkt.):

- po zakończeniu pracy w systemie nie są pozostawiane żadne procesy, tymczasowe pliki, mechanizmy komunikacji, semaforey itd.