

# Zadania - Zestaw 5

Potoki nazwane i nienazwane

Zadanie 1 (50%)

Napisz interpreter poleceń przechowywanych w pliku. Ścieżka do pliku to pierwszy argument wywołania programu.

Polecenia w pliku przechowywane są w kolejnych liniach w postaci:

```
prog1 arg1 ... argn1 | prog2 arg1 ... argn2 | ... | progN arg1  
... argnN
```

- Interpreter powinien uruchomić wszystkie N poleceń w osobnych procesach, zapewniając przy użyciu potoków nienazwanych oraz funkcji `dup2`, by wyjście standardowe procesu  $k$  było przekierowane do wejścia standardowego procesu  $(k+1)$
- Można założyć ograniczenie górne na ilość obsługiwanych argumentów oraz ilość połączonych komend w pojedynczym poleceniu (co najmniej 3).
- Po uruchomieniu ciągu programów składających się na pojedyncze polecenie (linijkę) interpreter powinien oczekiwać na zakończenie wszystkich tych programów.

Program należy zaimplementować korzystając z funkcji `pipe/fork/exec`.

Zadanie 2 (10%)

Napisać program przyjmujący jeden argument - ścieżkę do pliku tekstowego zawierających kilka linii tekstu (dowolne znaki). Wykorzystując `popen` oraz komendę `sort` posortować linie podanego pliku tekstowego (posortowane linie mają być wypisane na ekran).

Zadanie 3 (40%)

W problemie producenta i konsumenta występują dwa rodzaje procesów, które dzielą wspólny bufor dla produkowanych i konsumowanych jednostek. Zadaniem producenta jest wytworzenie surowca, umieszczenie go w buforze i rozpoczęcie pracy od nowa. Konsument pobiera surowiec z bufora i wykorzystuje go.

Przy pomocy potoków nazwanych zaimplementować problem Producenta i Konsumenta. Napisać dwa niezależne programy - Producent oraz Konsument, które będą komunikować się poprzez potok nazwany (kolejkę FIFO). Do potoku pisać będzie wiele procesów wykonujących program Producenta, a czytał będzie z niej jeden proces Konsumenta. Dla zademonstrowania, że nie doszło do utraty ani zwielokrotnienia towaru

surowiec będzie pobierany z pliku przez Producenta i umieszczany w innym pliku przez Konsumenta.

Producent:

- przyjmuje trzy argumenty: ścieżka do potoku nazwanego, ścieżka do pliku tekstowego z dowolną zawartością,  $N$  - liczba znaków odczytywanych jednorazowo z pliku
- otwiera potok nazwany
- wielokrotnie (aż do odczytania całego pliku):
  - odczekuje losową ilość czasu (np. 1-2 sekund)
  - zapisuje do potoku nazwanego linię zawierającą swój PID oraz odczytany fragment pliku w następującej formie: #PID#( $N$  odczytanych znaków)

Konsument:

- przyjmuje trzy argumenty: ścieżka do potoku nazwanego, ścieżka do pliku tekstowego (do którego będzie zapisywany odczytany tekst),  $N$  - liczba znaków odczytywanych jednorazowo z pliku
- otwiera potok nazwany
- wielokrotnie:
  - czyta kolejnych  $N$  znaków potoku nazwanego
  - umieszcza odczytane znaki w pliku tekstowym (różnym od plików, z których korzystają producenci)

Pliki tekstowe powinny być krótkie (na 5-10 odczytów) i umożliwiać sprawdzenie poprawności działania (brak utraty, zwielokrotnienia surowca). W szczególności każdy Producent powinien otrzymać wygenerowany w dowolny sposób plik tekstowy z dowolną zawartością, ale w istotny sposób różniącą się od zawartości plików innych Producentów. Na przykład jeden producent może otrzymać plik zawierający tylko konkretną literę, inny tylko liczby itd.

Sprawdzić, że potoki nazwane działają dla niezależnych procesów - utworzyć potok z linii komend, a następnie uruchomić Producenta i Konsumenta w różnych terminalach. Dodatkowo należy napisać program, który tworzy potok nazwany, a następnie uruchamia program Konsumenta i pięciu Producentów (z różnymi argumentami).