

SYSTEMY OPERACYJNE

Rozwiązanie problemu synchronizacji procesów przy pomocy semaforów na przykładzie problemu producentów-konsumentów

Autor – Szymon Koper

Data – 18.12.2013 r.

Opis

Program rozwiązuje zmodyfikowany problem producenta-konsumenta.
Modyfikacja polega na tym, że producentów i konsumentów może być wielu.

Liczbę uruchamianych procesów przekazuje się do programu posługując się parametrami.
Na podstawie parametru ustawiany jest także limit wyprodukowanych towarów.
Producenci produkują towary do momentu osiągnięcia wyznaczonego limitu produktów.
Konsumentom konsumują towary do momentu, w którym wszystkie produkty zostały
skonsumowane oraz wszystkie procesy producentów zostały zakończone.

Szczegółowe informacje dotyczące uruchamiania programu z parametrami opisane są w dalszej części dokumentu.

W trakcie swojego działania program informuje na bieżąco o występujących zdarzeniach, takich jak:

- rozpoczęcie lub zakończenie działania głównego procesu
- utworzenie lub zakończenie procesu konsumenta lub producenta
- produkcja lub konsumpcja towaru

Przykłady uruchamiania

Program uruchomiony bez argumentów wywoła się z domyślnymi wartościami:

```
./starter
```

Program można także uruchamiać z parametrami zmieniającymi domyślne wartości:

```
./starter -p 4 -c 5 --limit 30
```

Lista parametrów

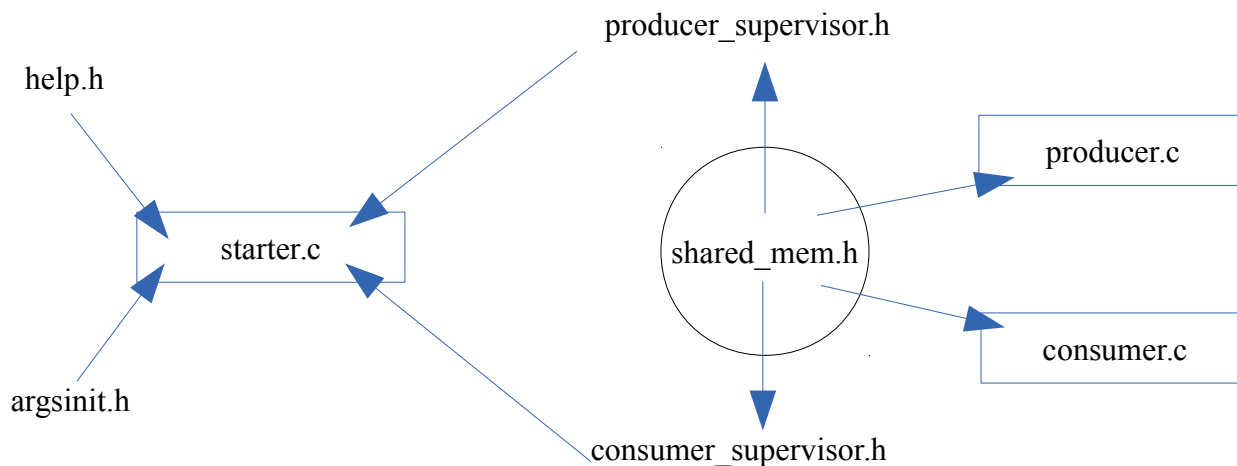
| | |
|-----------------------------------|---|
| -p --producers-number <wartość> | ustawia liczbę procesów – producentów |
| -c --consumers-number <wartość> | ustawia liczbę procesów – konsumentów |
| -m --producers-wait <wartość> | ustawia czas oczekiwania producenta |
| -e --consumers-wait <wartość> | ustawia czas oczekiwania konsumenta |
| -l --limit <wartość> | ustawia liczbę produktów do wytworzenia |
| -h --help | wyświetla pomoc |

W przypadku podania niepoprawnych parametrów program zadziała opierając się na wartościach podanych w rozpoznanych parametrach oraz wypisze listę parametrów niepoprawnych.

Domyślne wartości parametrów

| | |
|--------------------|----|
| --producers-number | 2 |
| --consumers-number | 4 |
| --producers-wait | 1 |
| --consumers-wait | 2 |
| --limit | 10 |

Struktura



Najważniejsze w projekcie są 3 pliki wykonywalne:

- starter
- producer
- consumer

Starter przy pomocy `shared_mem.h` tworzy dowiązanie do segmentu pamięci dzielonej. Pamięć tą inicjuje w przy pomocy wartości domyślnych lub przekazanych mu w parametrach. Zajmuję się tym fragment kodu znajdujący się w pliku `argsinit.h`. Wykorzystałem w tym celu funkcję `Getopt` z biblioteki języka C.

Następnie, korzystając z funkcji nadzorujących działanie procesów potomnych w plikach `producer_supervisor.h` oraz `consumer_supervisor.h`, uruchamia się odpowiednio wiele procesów producentów oraz konsumentów.

Wszystkie uruchomione procesy korzystają ze wspólnego uchwytu segmentu pamięci dzielonej. Dostęp do sekcji krytycznej reguluje semafor.

Gdy zostanie wyczerpany limit produktów do utworzenia kończą swoje działanie początkowo procesy producentów, a następnie po skonsumowaniu wszystkich dostępnych produktów kończą działanie konsumenci.

Dopiero na samym końcu swoje działanie kończy program nadzorujący – starter.

Plik `help.h` zawiera funkcję wyświetlającą informacje w trakcie działania programu oraz pomoc.

Pomysły na rozwinięcie projektu

- kolorowanie wyświetlanych informacji dla lepszej przejrzystości
- możliwość wprowadzenia losowości pomiędzy kolejnymi wywołaniami procesów
- możliwość uruchamiania procesów z opóźnieniem
- możliwość zmiany parametrów oraz uruchomionych procesów w trakcie działania programu