

## Laboratorium 3 - synchronizacja z wykorzystaniem semaforów

### Konstrukcja buforów i semaforów

Na początku stworzyłem strukturę reprezentującą bufor na dany składnik oraz powiązane z nim semafony, jej parametrami są kolejno:

- typ przechowanego składnika (ciasto lub 1 z 3 możliwych nadzień pieroga),
- semafor\_empty, który reprezentuje liczbę pustych miejsc bufora,
- semafor\_full, który reprezentuje liczbę zajętych miejsc bufora,
- mutex, który decyduje o wejściu do sekcji krytycznej,
- aktualna pozycja miejsca w buforze tudzież liczba składników w nim umieszczonych,
- tablica intów reprezentująca bufor (każde miejsce jest wolne - 0 lub zajęte - 1)

Każdy rodzaj produktu jest reprezentowany przez taką strukturę, konsumenci i producenci współdzielą bufor, a prawidłową synchronizację między nimi zapewniają semafony.

Procedura `init_buffer()` odpowiada za inicjalizację każdego z semaforów/buforów. Semafor empty ma na początku wartość rozmiaru bufora, a pozostałe wartości są inicjowane zerami.

### Producent

Procedura `produce_ingredients()` przyjmuje jako argument wskaźnik na void, gdyż takie wymagania narzuca funkcja tworzenia wątku `pthread_create()`. Następnie ten argument jest rzutowany na wskaźnik na strukturę `Buffer`, na którym już można łatwo operować. W nieskończonej pętli na początku opuszczam semafor empty - jeśli liczba pustych składników jest równa 0, czyli bufor jest pełny, wątek jest blokowany i nie może wyprodukować nic więcej. W przeciwnym wypadku liczba wolnych slotów zostanie zmniejszona o jeden, a producent wejdzie do sekcji krytycznej (zamykając przy tym mutex, który gwarantuje, że tylko on będzie w buforze) i umieści produkt w buforze. Następnie odblokowuje mutex, oznajmiając wyjście z sekcji krytycznej i podnosi semafor full, dzięki czemu zwiększa się liczba zajętych miejsc bufora o 1, a konsumenci otrzymują znak do działania.

### Konsument

Procedura konsumenta `consume_for_dumplings()` różni się głównie tym, że operuje na dwóch zestawach semaforów. Ponadto, na początku opuszczany jest semafor full dla nadzienia pieroga - tym sposobem, jeżeli liczba zapełnionych miejsc wynosi 0 (czyli nie ma z czego brać) wątek blokuje się. W przeciwnym razie zmniejsza liczbę dostępnych składników w buforze o 1. Analogicznie jest z drugim semaforem dla ciasta. Istotna jest kolejność, aby nie blokować dostępu do ciasta innym konsumentom, w momencie gdy brakuje nadzienia

pieroga. Później zamykany jest mutex - wątek wchodzi do sekcji krytycznej bufora ciasta i składnika i tam pobiera elementy, tworząc pierogi. Po wyjściu z sekcji mutex zostaje odblokowany, a semaforey empty podniesione.

## Main

Zgodnie z zaleceniami program przyjmuje 7 argumentów - liczbę wątków dla każdego rodzaju producenta i konsumenta. Po ich wczytaniu zliczam łączną wielkość i tworzę tablice wątków producentów i konsumentów. Następnie w 7 pętlach inicjalizuje wątki za pomocą `pthread_create`: do tablicy przypisywane jest ich id, NULL jako parametr oznacza brak atrybutów; podaję także funkcję producenta/konsumenta i jej argument - odpowiedniego rodzaju bufor. Później funkcja `pthread_join()` czeka na zakończenie działania wątków. Na końcu następuje zwolnienie zasobów dla semaforów za pomocą procedury `destroy_buffer()`.

## Testy

W programie zdefiniowane są dwie zmienne globalne (`SLOW_MODE` i `DEBUG_MODE`). W zależności od ich wartości logicznej można aktywować funkcję `sleep(2)` na wątkach, symulując ich powolne działanie bądź wyświetlać zawartość bufora po każdej zmianie za pomocą procedury `print_buffer()`.

W przypadku braku usypiania, widać, że konsumpcja i produkcja różnego rodzaju pierogów/składników występuje bez żadnego przerwania i przepełniania buforów. (Test dla rozmiaru bufora 10 i po 10 wątków każdego rodzaju)

```
Wyprodukowano: ciasto Pozycja bufora: 8
Wyprodukowano: ser Pozycja bufora: 10
Wyprodukowano: kapusta Pozycja bufora: 10
Wyprodukowano: ciasto Pozycja bufora: 9
Wyprodukowano: ciasto Pozycja bufora: 10
Konsumpcja: pierogi ze składnikiem kapusta
Konsumpcja: pierogi ze składnikiem kapusta
Wyprodukowano: kapusta Pozycja bufora: 9
Wyprodukowano: ciasto Pozycja bufora: 9
Wyprodukowano: kapusta Pozycja bufora: 10
Konsumpcja: pierogi ze składnikiem mięso
Konsumpcja: pierogi ze składnikiem mięso
Konsumpcja: pierogi ze składnikiem mięso
Wyprodukowano: mięso Pozycja bufora: 8
Wyprodukowano: mięso Pozycja bufora: 9
Wyprodukowano: mięso Pozycja bufora: 10
Konsumpcja: pierogi ze składnikiem kapusta
Konsumpcja: pierogi ze składnikiem ser
Wyprodukowano: kapusta Pozycja bufora: 10
Konsumpcja: pierogi ze składnikiem kapusta
Wyprodukowano: ser Pozycja bufora: 10
Wyprodukowano: kapusta Pozycja bufora: 10
Wyprodukowano: ciasto Pozycja bufora: 4
Wyprodukowano: ciasto Pozycja bufora: 5
Konsumpcja: pierogi ze składnikiem kapusta
Konsumpcja: pierogi ze składnikiem kapusta
Wyprodukowano: kapusta Pozycja bufora: 10
```

Z włączonym spowolnieniem i printowaniem zawartości bufora testowanie wymaga nieco więcej uwagi. Widać jednak, że wartości w buforach są pobierane i zapisywane prawidłowo.

```
1 1
Konsumpcja: pierogi ze składnikiem mięso
Bufor mięso: 1 1 1 1 1 1 1 1 1 0
Bufor ciasto: 1 0 0 0 0 0 0 0 0 0
Wyprodukowano: mięso Pozycja bufora: 10
Bufor mięso: 1 1 1 1 1 1 1 1 1 1
Wyprodukowano: ciasto Pozycja bufora: 2
Bufor ciasto: 1 1 0 0 0 0 0 0 0 0
Konsumpcja: pierogi ze składnikiem kapusta
Bufor kapusta: 1 1 1 1 1 1 1 1 1 0
Bufor ciasto: 1 0 0 0 0 0 0 0 0 0
Konsumpcja: pierogi ze składnikiem kapusta
Bufor kapusta: 1 1 1 1 1 1 1 1 0 0
Bufor ciasto: 0 0 0 0 0 0 0 0 0 0
Wyprodukowano: kapusta Pozycja bufora: 9
Bufor kapusta: 1 1 1 1 1 1 1 1 1 0
Wyprodukowano: ciasto Pozycja bufora: 1
Bufor ciasto: 1 0 0 0 0 0 0 0 0 0
Wyprodukowano: kapusta Pozycja bufora: 10
Bufor kapusta: 1 1 1 1 1 1 1 1 1 1
Konsumpcja: pierogi ze składnikiem mięso
Bufor mięso: 1 1 1 1 1 1 1 1 1 0
Bufor ciasto: 0 0 0 0 0 0 0 0 0 0
Wyprodukowano: mięso Pozycja bufora: 10
Bufor mięso: 1 1 1 1 1 1 1 1 1 1
^C
```

W związku tym spełnione są podstawowe założenia synchronizacji. Nie występuje jednocześnie zapis do bufora, czytanie z pustego bufora czy też wzajemne “przeszkadzanie sobie” wielu wątków przy zapisywaniu lub czytaniu z bufora.