

## L7: Bitronka

Jest rok 2025. W Bitronce — współczesnym imperium handlu detalicznego — nie panuje już chaos. Panuje wojna. W ferworze codziennej walki o bułki pełnoziarniste klienci wprowadzają zamęt na sklepowych półkach. Zamieniają miejscami chipsy z jogurtami, wciskają jabłka między gruszki. Guma do żucia? Najlepiej obok masła. Puszka ananasów? A co, jeśli wyląduje w dziale z nabiałem? Najgorsza w tym wszystkim jest pani Hermina, która codziennie przekłada baterie AA do działu mrożonek, bo „tu jest chłodniej”.

Miedzy regałami czai się jednak jeszcze groźniejsze zagrożenie niż półkowy rozgardiasz — palety. Dziesiątki, setki, całe wieże palet piętrzą się aż po sufit — chwiejne i niestabilne. Złośliwie stoją w poprzek przejść i tarasują drogi ewakuacyjne. Dla nowego pracownika jeden nierozważny krok w alejce może skończyć się fatalnie — przewrócona paleta często oznacza koniec zmiany... na zawsze.

Wieczorami, gdy ostatni klient opuszcza sklep z torbą pełną cukierków zważonych jako cebula, do akcji sprzątania wkraczają pracownicy. Próbuje przywrócić porządek. „Te baterie znowu przy mrożonkach?!” — krzyczy ktoś, a echo niesie się aż do działu owoców. Wiedzą, że porządkowanie sklepu to walka z wiatrakami, bo chaos powróci wraz z pierwszym otwarciem drzwi kolejnego ranka. Ale nie poddają się. Każdy z nich losowo wybiera parę produktów i ocenia, czy ich lokalizacja wymaga zmiany. Jeśli tak — produkty wędrują na właściwe (albo przynajmniej bardziej logiczne) miejsce. Oczywiście o ile nikt po drodze nie potknie się o wystającą paletę. Nad sprzątaniem czuwa kierownik zmiany, który z niedowierzaniem obserwuje, jak codziennie jego zespół toczy tę samą, z góry przegraną bitwę z entropią.

Twoim zadaniem jest napisanie symulacji działania zmiany sprzątającej, która nocą próbuje zaprowadzić porządek - chociażby na chwilę. **Symulacja ma zostać zrealizowana z wykorzystaniem pamięci współdzielonej oraz procesów.**

---

### Etapy:

#### 1. [6 pkt]

Program przyjmuje dwa parametry:  $8 \leq N \leq 256$  - liczbę produktów oraz  $1 \leq M \leq 64$  - liczbę pracowników.

Sklep reprezentowany jest jako tablica jednowymiarowa o rozmiarze  $N$ , gdzie każdy indeks tablicy oznacza jedną półkę, a wartość w nim to aktualny produkt (liczba całkowita od 1 do  $N$ ).

- Stwórz plik `SHOP_FILENAME` i zmapuj go do pamięci procesu, następnie zainicjalizuj tam tablicę reprezentującą półki sklepowe.
- Wypełnij ją kolejnymi liczbami całkowitymi i przetasuj, korzystając z dostarczonej metody `shuffle`.
- **Zakazane jest użycie strumieni oraz funkcji `read`.**
- Przed wyjściem z programu wypisz zawartość tablicy produktów, korzystając z dostarczonej metody `print_array`.

#### 2. [8 pkt]

- Stwórz  $M$  procesów pracowników.
- Każdy pracownik po stworzeniu wypisuje:  
[PID] Worker reports for a night shift
- Po zameldowaniu się, każdy pracownik 10 razy losowo wybiera dwa różne indeksy z przedziału  $[1, N]$ .
- Jeżeli produkty na tych indeksach są w złej kolejności (tzn. wartość w pierwszym indeksie jest większa niż w drugim), zamienia je miejscami, aby zwiększyć porządek na półkach, śpiąc 100 ms podczas zamiany.
- Po zakończonej pracy, proces pracownika kończy się.
- W pamięci procesu zmapuj anonimową pamięć współdzieloną, w której będą znajdować się muteksy chroniące dostęp do wartości w tablicy (jeden muteks na półkę).
- **W tej samej pamięci dzielonej umieszczaj też inne współdzielone zmienne.**
- Wypisz zawartość tablicy produktów przed uruchomieniem procesów pracowników.

- Po zakończeniu wszystkich procesów pracowników wypisz ponownie tablicę i dodaj:  
Night shift in Bitronka is over

### 3. [6 pkt]

- Po stworzeniu procesów pracowników stwórz proces kierownika zmiany.
- Przy starcie kierownik wypisuje:  
[PID] Manager reports for a night shift
- Proces kierownika co pół sekundy:
  - wypisuje zawartość tablicy,
  - synchronizuje plik z jej obecnym stanem,
  - sprawdza porządek. Jeśli jest posortowana, wypisuje:  
[PID] The shop shelves are sorted  
i ogłasza koniec pracy poprzez ustawienie zmiennej w pamięci dzielonej, a następnie kończy pracę.
- Pracownicy pracują do ogłoszenia przez kierownika końca pracy.

### 4. [5 pkt]

- W procesie pracowników dodaj 1% szansy na nagłą śmierć tuż przed zamianą produktów (użyj w tym celu funkcji `abort`).
- Przed śmiercią wypisują:  
[PID] Trips over pallet and dies
- Żywi pracownicy i kierownik zliczają liczbę żywych (lub martwych) pracowników we współdzielonej pamięci.
- Po znalezieniu martwego pracownika wypisują:  
[PID] Found a dead body in aisle [SHELF\_INDEX]  
Uwaga: Zauważ, że ciało może zostać znalezione 2 razy.
- Kierownik po wypisaniu stanu tablicy wypisuje komunikat o ilości żyjących pracowników:  
[PID] Workers alive: [ALIVE\_COUNT]
- Gdy nie ma już żyjących procesów pracowników, kierownik wypisuje:  
[PID] All workers died, I hate my job  
i kończy pracę.