Zadanie

- Magazyn o pojemności k sztuk towaru,
- n konsumentów, każdy z nich jednorazowo odbiera partię towaru w liczbie sztuk ~ U(a, b),
- m producentów, każdy z nich jednorazowo produkuje partię towaru w liczbie sztuk ~ U(c, d).

Producenci przekazują towar konsumentom za pośrednictwem magazynu. Producenci i konsumenci obsługują tylko całe partie towaru.

Producenci produkują tylko całe partie towaru. Jeśli mejsc w magazynie jest mniej niż liczba wyprodukowanych towarów, proces producenta zostaje wstrzymany. Analogicznie dla konsumenta.

Magazyn reprezentowany jako plik. Log producentów/konsumentów zapisywany do oddzielnych plików.

Koncepcja

Do rozwiązania potrzebujemy monitor, który będzie przechowywał liczbę towarów w magazynie oraz dwie zmienne warunkowe `empty` i `full` za pomocą których wątki producentów próbujących wstawić za dużo sztuk towaru lub konsumentów próbujących odebrać za dużo sztuk towaru nie będą niszczone, a zostaną wstrzymane i umieszczone w odpowiednich kolejkach. Wątek w kolejce `empty` zostanie wznowiony, kiedy dostanie sygnał od producenta, że wyprodukowano kilka sztuk towaru, zaś wątek w kolejce `full` zostanie wznowiony, kiedy dostanie sygnał od konsumenta, że odebrano kilka sztuk towaru.

Pseudocode

Monitor:

```
monitor Buffer
    integer size;
    integer items;
    condition full, empty;
    semaphore_t mutex;

    procedure enter(n)
    begin
        mutex.acquire();
        if items + n >= size
            then wait(full)
        items := items + n;
        if items > 0
            then signal(empty)
        mutex.release();
    end;

    procedure remove(n)
    begin
        mutex.acquire();
        if items - n < 0
            then wait(empty)
        items := items - n;
```

```
        if items < size
            then signal(full)
        mutex.release();
    end;
end monitor;
```

Procedury konsumenta, producenta:

```
procedure producer()
begin
    while(1) do
    begin
        n = random from A to B;
        write_to_file("Trying to insert %d items.", n);
        Buffer.enter(n);
        write_to_file("Inserted %d items.", n);
    end;

end;

procedure consumer()
begin
    while(1) do
    begin
        n = random from A to B;
        write_to_file("Trying to consume %d items.", n);
        Buffer.remove(n);
        write_to_file("Consumed %d items.", n);
    end;

end;
```

Przykładowe działanie

Przykładowe działanie programu dla pustego magazynu o pojemności k = 5, zainicjowanych dwóch procesach producenta i jednego procesu konsumenta:

Plik producenta1:

```
(00:00:00) Trying to insert 2 items.
(00:00:00) Inserted 2 items.
(00:00:03) Trying to insert 4 items.
(00:00:03) Inserted 4 items.
```

Plik producenta2:

```
(00:00:01) Trying to insert 2 items.
(00:00:01) Inserted 2 itemse.
(00:00:04) Trying to insert 4 items.
(00:00:06) Inserted 4 items.
```

Plik konsumenta:

```
(00:00:02) Trying to consume 3 items.
(00:00:02) Consumed 3 items.
(00:00:05) Trying to consume 5 items.
(00:00:05) Consumed 5 items.
```

Plik magazynu:

```
4
```