

## Lista 3

**Termin wykonania: 2025-05-25**

W repozytorium SVN: <https://repo.cs.pwr.edu.pl/info/PW/> zostały przygotowane dodatkowe pliki do wykorzystania przy rozwiązywaniu zadań:

- `ada/tasks/mutex_template.adb` jest szablonem, który można przerobić i uzupełnić implementacjami protokołów wejściowych i wyjściowych algorytmów wzajemnego wykluczania tak aby generował wymagany output.
- `bash-tools/display-travel-2.bash` jest skryptem wyświetlającym output generowany przez `mutex_template`.

Przetestować wyświetlanie output-u programu `mutex_template`

przez skrypt `display-travel-2.bash`. Na pierwszym ekranie wyświetlane są informacje o etykietach:

**LABELS:**

`LOCAL_SECTION ENTRY_PROTOCOL CRITICAL_SECTION EXIT_PROTOCOL EXTRA_LABEL`

Press "ENTER"

Po naciśnięciu `ENTER` następuje wyświetlanie symulacji:

```
STEP = 48  TIME = 0.043310318
.A.B...D...F.....K..... <- LOCAL_SECTION
..... <- ENTRY_PROTOCOL
.....C.....G.H.I.J...L.M.N.O <- CRITICAL_SECTION
.....E..... <- EXIT_PROTOCOL
TO STEP?
```

Proces  $i$  wyświetlany jest w  $i$ -tej kolumnie w wierszu odpowiadającym jego stanowi. (Ponieważ nie są jeszcze zaimplementowane protokoły wejściowy i wyjściowy, więc w wierszu oznaczonym etykietą `CRITICAL_SECTION` może pojawić się więcej niż jeden proces.) Stany zdefiniowane są w `mutex_template.adb`:

-- States of a Process

```
type Process_State is (
  Local_Section,
  Entry_Protocol,
  Critical_Section,
  Exit_Protocol
);
```

Liczba wierszy jest zdeterminowana przez liczbę stanów. Etykieta `EXTRA_LABEL` dopisana jest w wątku `Printer` po zakończeniu działania algorytmu :

```
Put_Line("EXTRA_LABEL;"); -- Place labels with extra info here (e.g.
"MAX_TICKET=...;" for Bakery).
```

Można to wykorzystać do wyświetlania dodatkowych informacji zebranych podczas działania algorytmu.

Sprawozdania przygotowywać jak dla poprzednich list.

**Uwaga:** Ponieważ implementowane mają być symulacje algorytmów *niskopoziomowych*, więc wolno korzystać jedynie z *atomowych* operacji zapisu lub odczytu zmiennych skalarnych (w tym - skalarnych elementów tablic ale nie tablic jako całych struktur.)

#### **Zadanie 1. (za 10 punktów)**

Zaimplementować w Adzie algorytm piekarniany wykorzystując `mutex_template.adb` tak aby można było wyświetlić symulację jego działania skrypcem `display-travel-2.bash`. W miejscu `EXTRA_LABEL` ma być wyświetlana maksymalna użyta wartość biletu. (Najlepiej jeśli każdy proces sam zapamiętuje największy otrzymany bilet a następnie, tuż przed wykonaniem `Printer.Report( Traces )`, aktualizuje tą wartością globalne maksimum przechowywane np. w obiekcie `protected`.)

#### **Zadanie 2. (za 10 punktów)**

Zaimplementować rozwiązanie Zadania 1 w języku Go.

#### **Zadanie 3. (za 5 punktów)**

Zaimplementować w Adzie algorytm Dekkera wykorzystując `mutex_template.adb` tak aby można było wyświetlić symulację jego działania skrypcem `display-travel-2.bash`. Zauważ, że konieczne jest ustawienie:

```
Nr_Of_Processes : constant Integer := 2;
```

ponieważ jest on przeznaczony dla dwóch procesów.

#### **Zadanie 4. (za 5 punktów)**

Zaimplementować rozwiązanie Zadania 3 w języku Go.

#### **Zadanie 5. (za 5 punktów)**

Zaimplementować w Adzie algorytm Petersona wykorzystując `mutex_template.adb` tak aby można było wyświetlić symulację jego działania skrypcem `display-travel-2.bash`. Zauważ, że również konieczne jest ustawienie:

```
Nr_Of_Processes : constant Integer := 2;
```

#### **Zadanie 6. (za 5 punktów)**

Zaimplementować rozwiązanie Zadania 5 w języku Go.