

# Systemy Operacyjne

## Konspekt projektu trzeciego

### Temat:

Komunikacja międzyprocesowa z wykorzystaniem mechanizmu monitorów.

### Zadania do zrealizowania:

- Zaimplementować mechanizm priorytetowych kolejek wiadomości dla komunikacji międzyprocesowej.
- Wykorzystując zaimplementowany mechanizm kolejkowy przygotować części składowe systemu przetwarzającego wiadomości w kolejkach.
- Przeprowadzić prezentację systemu, odpowiedzieć na pytania.

### Propozycja rozwiązania punktu 1:

- Struktury kolejek i komunikatów tworzone w konstruktorach obiektów przechowywanych w pamięci współdzielonej.
- Implementacja list dynamicznych komunikatów różniących się od standardowej implementacji tym, że wskazanie na kolejny element nie jest absolutnym wskaźnikiem na komunikat, ale określa pozycję komunikatu względem początku tablicy komunikatów.
- Taka metoda wynika z faktu wykorzystania przez procesy pamięci współdzielonej, która nie ma gwarancji dowiązania pod ten sam zakres przestrzeni adresowej poszczególnych procesów.
- Wszystkie komunikaty z danej kolejki zawarte są w tablicy komunikatów o określonej pojemności.
- Stworzenie tablicy struktur typu Message zawierającej:
  - trzy-elementową tablicę znaków z zakresu ('A', 'B', 'C'), lub NULL
  - priorytet komunikatu
  - wskazanie na następny komunikat zawierające:
    - adres następnego komunikatu względem początku tablicy komunikatów lub
    - w przypadku braku następnego komunikatu wartość spoza zakresu tablicy komunikatów, np. wartość wskazującą na element o jeden dalej niż ostatni komunikat w tablicy.
- Stworzenie klasy Queue dziedziczącej po klasie Monitor do obsługi kolejek priorytetowych z funkcjonalnościami:
  - Nieblokującymi:
    - stan zajętości kolejki.
      - stan zajętości określany na podstawie ilości aktywnych komunikatów w tablicy komunikatów.
  - Blokującymi:

- Pobranie komunikatu z kolejki:
  - pobranie pierwszego komunikatu z listy.
- Wpisanie komunikatu do kolejki:
  - komunikaty zwykłe wpisywane na koniec kolejki,
  - komunikaty priorytetowe wpisywane na początek kolejki zaraz za istniejącymi komunikatami priorytetowymi.
- Blokowanie procesów realizowane w funkcjach pobierania i wysyłania komunikatu przy wykorzystaniu semaforów. Procesy nie zarządzają bezpośrednio semaforami.

#### **Propozycja rozwiązania zadania 2:**

- Utworzenie trzech kolejek komunikatów o identyfikatorach „A”, „B” i „C” o odpowiednich pojemnościach w przestrzeni pamięci współdzielonej,
- Każdy z producentów i prosumentów zaimplementowany jako osobna funkcja wykonywana w osobnym wątku wykorzystujący bibliotekę kolejek priorytetowych,
- Producent jako argument wywołania przyjmuje identyfikator kolejki do której rozpoczyna nadawanie.
- Konsument jako argument wywołania przyjmuje identyfikator kolejki z której rozpoczyna odbierać.
- Producent specjalny różni się od producenta zwykłego nadawanym priorytetem oraz losowością wybierania kolejek do których trafią losowo wygenerowane komunikaty.
- Producent ochronny sprawdza stan kolejek i wstawia swój komunikat do najbardziej zapełnionej. Jego komunikat pozwala na udrożnienie danej kolejki przez prosumenta.

#### **Propozycja rozwiązania zadania 3:**

- Testy systemu realizowane przy pomocy wypisywania na bieżąco informacji o stanie kolejek oraz zawartych w nich komunikatach