SOI Zadanie na laboratorium 4.

Grupa pon. 10:15-12:00 CS500

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

Uwagi szczegółowe do punktu 1

- Kolejki należy zaimplementować **w systemie Linux** przy użyciu **monitorów**.
  - Dopuszczalna – ale absolutnie nie wymagana – jest zmiana implementacji z wieloprocesowej na wielowątkową.
  - W wariancie wielowątkowym pamięć wspólna jest zapewniona przez sam proces macierzysty, więc wystarczy zaimplementować monitory w oparciu np. o mutexy.
  - W wariancie wieloprocesowym należy wykorzystać semafory i pamięć wspólną, jak w zadaniu 3, ukrywając je pod postacią monitorów.
- Kolejki powinny zapewniać:
  - ograniczenie pojemności kolejki
  - mechanizm producent-konsument
  - obsługę priorytetową wskazanych obiektów (obiekty o większym priorytecie wyprzedzają w kolejce te o niższym priorytecie)
  - dostarczanie według zasady FIFO (first in – first out) w ramach każdego priorytetu.
- Zalecana jest implementacja biblioteki realizującej funkcje mechanizmu kolejkowego, wykorzystywanej przez pozostałe programy implementowane w ramach zadania.

Uwagi szczegółowe do punktu 2

Obowiązują reguły ustalone w zadaniu 3 z następującymi zmianami:

- Komunikaty mogą mieć trzy priorytety: normalny, wysoki i najwyższy.
- Nowy typ procesu – producent ochronny (O). Proces ten z zadaną częstością wrzuca do aktualnie najbardziej wypełnionej kolejki jeden pusty komunikat o najwyższym priorytecie.
- Proces K, który otrzymuje komunikat o najwyższym priorytecie, następne 5 komunikatów obsługuje **bez oczekiwania**. Półsekundowe opóźnienie znika. Komunikaty o najwyższym priorytecie nie są przez prosumentów wrzucane ponownie do żadnej kolejki.

Uwagi szczegółowe do punktu 3

Stwierdzić eksperymentalnie dla kolejek o pojemności 20:

- Czy obecność procesu O wystarcza do utrzymania systemu w działaniu na sensownym (parominutowym) horyzoncie przy prawdopodobieństwie ***pr*** równym 0.7? Jaka częstość wysyłania komunikatów przez proces O jest do tego potrzebna?
- Jakie jest największe prawdopodobieństwo ***pr*** dla którego obecność procesu O zapewnia stabilne działanie systemu na sensownym (parominutowym) horyzoncie? Czy, a jeśli tak to przy jakiej częstości wysyłania komunikatów przez proces O wzrost skuteczności przestaje być zauważalny (tzn. komunikaty o najwyższym priorytecie dokładają więcej obciążenia niż usuwają)?