**Źródło:**

**https://www.ii.pwr.edu.pl/~juszczyszyn/so.htm**

**Zadanie 1**

Program ma symulować działanie algorytmów planowania dostępu do procesora dla zgłaszających się procesów.

Zbadać średni czas oczekiwania procesów dla różnych algorytmów planowania:

- FCFS

- SJF (z wywłaszczaniem i bez)

- rotacyjnego (z możliwością wyboru kwantu czasu)

Należy samodzielnie sformułować założenia symulacji.

Wskazówki:

- algorytmy najlepiej sprawdzać dla tych samych danych testowych (tj. tych samych ciągów testowych zgłaszających się procesów)

- ciągów testowych powinno być więcej (20? 50?); wynikiem będą wartości średnie.

- w każdym ciągu będzie N procesów o losowych długościach fazy procesora (rozkład długości faz dobrać tak, by odpowiadał sytuacji w rzeczywistym systemie, w którym nie jest równomierny), zgłaszających się w losowych momentach (dobrać parametry tak, by mogła powstać kolejka procesów oczekujących na przydział procesora).

- możliwa reprezentacja procesu: rekord (numer, dł.fazy procesora, moment zgłoszenia się, czas oczekiwania /początkowo równy 0/...)

Uzyskane wyniki należy wytłumaczyć i być gotowym na wyciągnięcie z nich wniosków... :)

Mile widziana możliwość sterowania parametrami symulacji.

Przy zaliczeniu należy być przygotowanym na ew. pytania dotyczące materiału omówionego na wykładzie i związanego z tematem zadania...

**Zadanie 2**

Symulacja algorytmów planowania dostępu do dysku.

\* 'Dysk' to w naszym przypadku liniowo uporządkowany ciąg bloków o nr od 1 do MAX.

\* Kryterium oceny algorytmów będzie suma przemieszczeń głowicy dysku, jak wiadomo proporcjonalna do czasu realizacji zleceń.

\* 1.Sprawdzić algorytmy FCFS, SSTF, SCAN i C-SCAN.

\* 2.Następnie założyć, że w systemie istnieją także aplikacje real-time, które musza być obsłużone za pomocą EDF i/lub FD-SCAN. Jak wpływa to na wyniki?

UWAGA!

Sformułowanie nie wymienionych powyżej warunków symulacji należy do Państwa. Mam na myśli:

-wielkość 'dysku' (ilość bloków)

-liczba i sposób generowania zgłoszeń (pełna kolejka od początku? zgłoszenia w trakcie? rozkład zgłoszeń- równomierny, inny?)

-sposób uwzględnienia obsługi zgłoszeń real-time

-pozostałe... >>> mile widziana umiejętność uzasadnienia przyjętego rozwiązania.

**Zadanie 3**

Badanie algorytmów zastępowania stron.

Należy samodzielnie sformułować założenia symulacji:

- rozmiar pamięci wirtualnej (ilość stron).

- rozmiar pamięci fizycznej (ilość ramek).

- długość (powinna być znaczna - min. 1000) i sposób generowania ciągu odwołań do stron (koniecznie uwzględnić zasadę lokalności odwołań).

Działanie programu:

- wygenerować losowy ciąg n odwołań do stron

- dla wygenerowanego ciągu podać liczbę błędów strony dla różnych algorytmów zastępowania stron:

1. FIFO (usuwamy stronę najdłużej przebywającą w pamięci fizycznej)

2. OPT (optymalny - usuwamy stronę, która nie będzie najdłużej używana)

3. LRU (usuwamy stronę, do której najdłużej nie nastąpiło odwołanie)

4. aproksymowany LRU (wiadomo)

5. RAND (usuwamy losowo wybraną stronę)

- symulacje przeprowadzić (na tym samym ciągu testowym) dla różnej liczby ramek (np. kilku (3, 5, 10?) wartości podanych przez użytkownika)

Zakres materiału: wszystko o pamięci wirtualnej (z wykładu).

**Zadanie 4**

Postępująca komplikacja zad. 4. Założyć, że:

- w systemie działa pewna ilość (rzędu ~10) procesów.

- każdy korzysta z własnego zbioru stron (zas. lokalności wciąż obowiązuje).

- globalny ciąg odwołań jest wynikiem połączenia sekwencji odwołań generowanych przez poszczególne procesy (każdy generuje ich wiele, nie jedną)

- każdemu system przydziela określoną liczbę ramek. na podstawie następujących metod:

1. Przydział proporcjonalny

2. Przydział równy

3. Sterowanie częstością błędów strony

4. Model strefowy.

- zastępowanie stron odbywa się zgodnie z LRU.

Jak strategie przydziału ramek wpływają na wyniki (ilość błędów strony - globalnie, dla każdego procesu)?

Program powinien wypisywać na ekranie przyjęte założenia symulacji. Mile widziana możliwość ich zmiany przez użytkownika.

Wnioski?

Zakres materiału: jak w zad 3.

**Zadanie 5**

Symulacja rozproszonego alg. równoważącego obciążenie procesorów.

W systemie pracuje N identycznych procesorów. Na każdym z nich pojawiają się nowe zadania (procesy), z RÓŻNĄ częstotliwością i RÓŻNYMI wymaganiami (każdy proces wymaga określonego, różnego, udziału w mocy obl. procesora - np ~3%). Zasymulować nast. strategie przydziału:

Na procesorze x pojawia sie zadanie. Nastepnie:

1. x pyta losowo wybr. procesor y o aktualne obciążenie. Jeśli jest mniejsze od progu p, proces jest tam wysyłany. Jeśli nie, losujemy i pytamy następny, próbując co najwyżej z razy. Jeśli wszystkie wylosowane są obciążone powyżej p, proces wykonuje się na x.

2.Jesli obciążenie x przekracza wartość progową p, proces zostaje wysłany na losowo wybrany procesor y o obciążeniu mniejszym od p (jeśli wylosowany y ma obc.>p, losowanie powtarza się do skutku). Jeśli nie przekracza - proces wykonuje się na x.

3.Jak w pkt 2, z tym że procesory o obciążeniu mniejszym od minimalnego progu r pytają losowo wybrane procesory i jesli obc. zapytanego jest większe od p, pytający przejmuje część jego zadań (założyć jaką).

Przeprowadzić symulację strategii 1-3 dla N=ok.50-100 i długiej serii zadań do wykonania (parametry dobrać samodzielnie, tak by całość zadziałała:). W każdym przypadku podać jako wynik:

A. Średnie obciążenie procesorów (zdecydować, rozsądnie, jak będzie obliczane).

B. Średnie odchylenie od wartości z pkt A.

C. Ilość zapytań o obciążenie oraz migracji (przemieszczeń) procesów.

Użytkownik powinien mieć możliwość podania (zmiany) wartości p,r,z,N.