

Warunki zaliczenia laboratorium *Systemy operacyjne*

I. Organizacja laboratorium

1. Zagadnienia do zrealizowania na zajęciach laboratoryjnych, zadania do wykonania oraz wszystkie zasoby związane z laboratorium będą umieszczane przez prowadzącego na Board'dzie:
eportal.ii.pwr.edu.pl/
Board-> Indyka-Piasecka.Agnieszka-> Systemy_operacyjne_(lab)
2. Na zajęciach studenci wykonują wszystkie zadania programistyczne indywidualnie i samodzielnie.
3. Studenci podzieleni zostaną przez prowadzącego na dwie podgrupy (gr. I oraz gr. II). Podział na podgrupy zostanie ustalony przez prowadzącego i przesłany studentom do wiadomości drogą mailową.
4. Każda podgrupa ma ustalony termin zaliczenia kolejnych zadań. Studenci z podgrupy mają obowiązek oddania prac w terminie ustalonym w Harmonogramie laboratorium (poniżej pkt III.).
5. Za zadanie oddane ze spóźnieniem o jeden termin, można otrzymać ocenę max. 3,0. Po większym spóźnieniu zadanie nie będzie oceniane.

II. Warunki zaliczenia laboratorium

1. Poszczególne zadania będą oceniane wg. tradycyjnej skali akademickiej tj. 3,0-5,0.
2. Ocena końcowa wystawiona przez prowadzącego będzie zbliżona do średniej arytmetycznej ocen za poszczególne zadania, uzyskanych przez studenta w trakcie semestru.
3. Obecność na zajęciach jest obowiązkowa.
4. Dopuszczalna w ciągu całego semestru jest tylko 1 nieobecność. Nie jest istotne, czy są to nieobecność usprawiedliwiona, czy nieusprawiedliwiona.

III. Harmonogram laboratorium

Lp.		Termin zaliczenia zadania	
		piątek	poniedziałek
1.	Zajęcia wprowadzające	3.03	6.03
2.	Zadanie 1(algorytm) gr. I i gr.II	10.03	13.03
3.	Zadanie 1(program) grupa I	17.03	20.03
4.	Zadanie 1 (program) grupa II	24.03	27.03
5.	Zadanie 2 (algorytm) gr. I i gr.II	31.03	3.04
6.	Zadanie 2 (program) grupa I	7.04	10.04
7.	Zadanie 2 (program) grupa II	21.04	śr 19.04
8.	Zadanie 3 i 4 (algorytm) gr. I i gr.II	28.04	24.04
9.	Zadanie 3 (program) grupa I	5.05	8.05
10.	Zadanie 3 (program) grupa II	12.05	15.05
11.	Zadanie 4 (program) grupa I	19.05	22.05
12.	Zadanie 4 (program) grupa II	26.05	29.05
13.	Zadanie 5 (algorytm) gr. I i gr.II	2.06	5.06
14.	Zadanie 5 (program) grupa I	9.06	12.06
15.	Zadanie 5 (program) grupa II	16.06	19.06

IV. Ustalenia końcowe

Skrzynką mailową do komunikacji elektronicznej pomiędzy studentem a prowadzącym jest politechniczna skrzynka e-mail studenta oraz służbowa skrzynka e-mail prowadzącego. Listy z prywatnych kont studentów nie będą czytane przez prowadzącego.

Jeśli będzie taka potrzeba prowadzący będzie przysyłał studentom informacje za pośrednictwem systemu EdukacjaCL. Student ma obowiązek regularnego sprawdzania zawartości politechnicznej skrzynki e-mail.

V. Literatura

1. A. Silbershatz, J.L. Peterson, P.B. Galvin, *Podstawy systemów operacyjnych*, WNT, 1993
2. A.S. Tannenbaum, *Rozproszone systemy operacyjne*, Wydawnictwo Naukowe PWN, 1997
3. A.M. Lister, R.D. Eager, *Wprowadzenie do systemów operacyjnych*, WNT, 1994
4. W.R. Stevens, *Programowanie zastosowań sieciowych w systemie UNIX*, WNT, 1995
5. Gabassi, *Przetwarzanie rozproszone w systemie UNIX*, Wydawnictwo Lupus
6. M.J. Bach, *Budowa systemu operacyjnego UNIX*, WNT, 1995

VI. Lista zadań

Zadanie 1 (termin: wg. harmonogramu)

Program ma symulować działanie algorytmów planowania dostępu do procesora dla zgłaszających się procesów. Zbadać średni czas oczekiwania procesów dla różnych algorytmów planowania:

1. - FCFS
2. - SJF (z wywłaszczaniem i bez)
3. - 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 (dobrac parametry tak, by mogła powstać kolejka procesów oczekujących na przydział procesora).
- Możliwa reprezentacja procesu: rekord (numer, długość 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 (termin: wg. harmonogramu)

Symulacja algorytmów planowania dostępu do dysku.

1. Sprawdzić algorytmy FCFS, SSTF, SCAN i C-SCAN.
2. Następnie założyć, że w systemie istnieją także aplikacje real-time, które muszą być obsługane za pomocą EDF i/lub FD-SCAN. Jak wpływa to na wyniki?

Wskazówki:

- 'Dysk' to w naszym przypadku liniowo uporządkowany ciąg bloków o nr od 1 do MAX.
 - Kryterium oceny alg. będzie suma przemieszczeń głowicy dysku, jak wiadomo proporcjonalna do czasu realizacji zleceń.
 - Sformułowanie nie wymienionych powyżej warunków symulacji należy do Państwa tzn.: 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.
-

Zadanie 3 (termin: wg. harmonogramu)

Badanie algorytmów zastępowania stron.

- I. Wygenerować losowy ciąg n odwołań do stron
- II. Dla wygenerowanego ciągu podać liczbę braków strony dla różnych algorytmów zastępowania stron:
 1. FIFO (usuwaamy stronę najdłuższą przebywającą w pamięci fizycznej)
 2. OPT (optymalny - usuwaamy stronę, która nie będzie najdłuższą używana)
 3. LRU (usuwaamy stronę, do której najdłuższą nie nastąpiło odwołanie)
 4. aproksymowany LRU (Last Recently Used)
 5. RAND (usuwaamy losowo wybraną stronę)

Wskazówki:

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

Zadanie 4 (termin: wg. harmonogramu)

Postępująca komplikacja zad. 3. Założyć, że: w systemie działa pewna ilość (rzędu ~ 10) procesów, każdy korzysta z własnego zbioru stron (zasada lokalności wciąż obowiązuje), globalny ciąg odwołań jest wynikiem połączenia sekwencji odwołań generowanych przez poszczególne procesy (każdy generuje ich wiele, nie jedną), zastępowanie stron odbywa się zgodnie z LRU, każdemu system przydziela określoną liczbę ramek, na podstawie następujących metod:

1. Przydział proporcjonalny
 2. Przydział równy
 3. Sterowanie częstością błędów strony
 4. Model strefowy
- Jak strategię przydziału ramek wpływają na wyniki (liczbę braków strony - globalnie, dla każdego procesu)?
 - Program powinien wypisywać na ekranie przyjęte założenia symulacji. Mile widziana możliwość ich zmiany przez użytkownika.
-

Zadanie 5 (termin: wg. harmonogramu)

Symulacja rozproszonego algorytmu 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 obliczeniowej procesora, np $\sim 3\%$). Zasyмуляć poniższe strategię przydziału.

Na procesorze x pojawia się zadanie. Następnie:

1. X pyta losowo wybrany procesor y o aktualne obciążenie. Jeśli jest mniejsze od **progu p** , proces jest tam wysyłany. Jeśli nie, losujemy i pytamy następnego, próbując co najwyżej z razy. Jeśli wszystkie wylosowane są obciążone powyżej p , proces wykonuje się na x .
2. Jeśli obciążenie x przekracza wartość progową p , proces zostaje wysłany na losowo wybrany procesor y o obciążeniu mniejszym od p (jeśli wylosowany y ma obciążenie $> p$, losowanie powtarza się do skutku). Jeśli nie przekracza - proces wykonuje się na x .
3. Jak wyżej, z tym że procesory o obciążeniu mniejszym od minimalnego **progu r** pytają losowo wybrane procesory i jeśli obciążenie zapytanego jest większe od p , pytający przejmuje część jego zadań (założyć jaką).

Wskazówki:

- Przeprowadzić symulację strategii 1-3 dla $N = \text{ok. } 50-100$ i długiej serii zadań do wykonania (parametry dobrać samodzielnie, tak by całość zadziałała:). W każdym przypadku podać jako wynik:
 1. Średnie obciążenie procesorów (zdecydować, rozsądnie, jak będzie obliczane).
 2. Średnie odchylenie od wartości z pkt wyżej.
 3. Ilość zapytań o obciążenie oraz migracji (przemieszczeń) procesów.
 - Użytkownik powinien mieć możliwość podania (zmiany) wartości p, r, z, N .
-