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 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.. .

Uwagi ogólne: Odpowiedzi na większość pytań znajdziecie Państwo 3 pozycjach literaturowych: • A. Silbershatz, J.L. Peterson, P.B. Galvin, Podstawy systemów operacyjnych • A.S. Tannenbaum, Rozproszone systemy operacyjne • Efcia, Systemy operacyjne - skrypt do wykładu Polecam zapoznanie się z treścią, 2 pierwsze pozycje wyjaśniają działanie algorytmów na przykładach i są szczególnie przydatne w kontekście laboratorium, pozycja ostatnia może się przydać do egzaminu (chociaż nie zawiera odpowiedzi na wszystkie pytania) Wszystko powinno być dostępne online. Wszystkie przykłady poradzenia sobie z trudnościami implementacyjnymi są jedynie propozycjami, nie musicie się Państwo do nich stosować, możliwe jest też znalezienie drogi lepszej bądź prostszej, w tym dokumencie chodzi tylko o przekazanie pewnej „intuicji", jak można sobie poradzić z zadaniem. Skróty: FCFS – algorytm „First Come First Served" SJF – algorytm „Shortest Job First" RR – algorytm „Round Robin" Zadanie 1 – algorytmy planowania dostępu do procesora Założenia ogólne: będzie potrzebny Państwu sposób symulowania czasu. Patrząc niskopoziomowo, w ten czy inny sposób na pewno zostanie wykorzystana jakaś pętla do przetwarzania ciągu zgłoszeń do CPU. Na pewno też w jakiś sposób trzeba zamodelować samo żądanie, jego czas pojawienia się oraz czas potrzebny do wykonania. Jednym z możliwych podejść jest przyjęcie bezwymiarowej jednostki czasu, która będzie odpowiadała jednej iteracji w pętli. Wtedy w każdej iteracji można zmniejszać czas potrzebny do

wykonania o 1, o ile żądanie aktualnie jest wykonywane przez CPU. Parametry takie jak czas pojawienia się żądania czy czas trwania muszą być w jakiś sposób generowane, na potrzeby symulacji spodziewałabym się jakiegoś generatora liczb pseudolosowych (w wersji ambitnej z rozkładem prawdopodobieństwa w kontekście czasu trwania żądań), w celu pokazania przypadków skrajnych (słabości konkretnych algorytmów) przygotowania odpowiednich sekwencji własnoręcznie i wczytywanie do programu. Samo losowanie momentu pojawienia się zdania również można zrealizować na kilka sposobów. Możliwe jest przygotowanie całej sekwencji zgłoszeń na początku pracy programu, a potem już tylko obsługa przez zaimplementowane algorytmy, ale można również sekwencję uzupełniać dynamicznie w trakcie symulacji. W drugim przypadków również przyda się generator liczb pseudolosowych, aby w losowym momencie dodawać nowe żądania do kolejek. W kwestii porównywania wyników należy zastanowić się nad statystykami potrzebnymi do prezentacji różnic pomiędzy algorytmami. Na pewno warto rozważyć policzenie średniego czasu oczekiwania na dostęp do procesora. Na potrzeby wykazania pewnej słabości SJF sugerowałabym podawanie najdłuższego czasu oczekiwania jednego procesu. Podobnie w przypadku RR należałoby przyjąć pewną jednostkę (być może wartość z przedziału (0,1], być może więcej niż 1) czasu dla przełączania pomiędzy obsługą procesów (to oczywiście również należy uwzględnić w innych algorytmach), co pozwoli zauważyć pewne słabości RR przy zbyt małym kwancie czasu. Wyliczanie średniego czasu od rozpoczęcia wykonywania pewnego zadania do jego zakończenia również pozwoli wykazać różnice w zachowaniu FCFS w stosunku do algorytmów wywłaszczających takich jak SJF czy RR. Warto rozważyć czy lepiej podawać wartości uśrednione pewnych parametrów czy np. sumę (np. liczbę przełączeń pomiędzy żądaniami). Odnośnie algorytmu SJF należy zastanowić się, jak oznaczyć procesy zagłodzone (takie, które utknęły w kolejce na zbyt długi czas, w teorii mógłby on być nieskończony). W samej implementacji mogą Państwu być pomocne pewne algorytmy i struktury danych, do których bezpośrednio nawiązują algorytmy. FCFS to oczywiście kolejka FIFO, więc wydaje się być naturalną strukturą do wykorzystaniu przy realizacji tego algorytmu (choć oczywiście można posługiwać się zwykłą tablicą czy jakąś kolekcją). Do SJF w jakiś sposób będziecie Państwo prawdopodobnie używali posortowanych kolekcji, więc być może możliwe jest zastosowanie wydajnych metod wyszukiwania elementów w zbiorach uporządkowanych (np. wyszukiwania binarnego). W kwestii omówienia działania algorytmów odsyłam do wykładów, literatury oraz zachęcam do kontaktu, możemy wyznaczyć termin na prezentację w formie telekonferencji.