



Politechnika
Wrocławska

Sprawozdanie – Systemy operacyjne

Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji
Kierunek Teleinformatyka

Wykonał: Adrian Panek
Numer indeksu: 259690
Semestr zimowy 2021/2022
Przedmiot: Systemy operacyjne – laboratoria
Data zajęć: Czwartek, 18:55 – 20:35
Miejsce zajęć: C-16, L2.4/sala wirtualna
Repozytorium na Githubie: https://github.com/adrian-panek/OS_LAB-algorithms

Spis treści

1. Wstęp teoretyczny.....	3
2. Opis procedury testowania algorytmów.....	4
3. Opracowane wyniki eksperymentów.....	5
4. Wnioski.....	6

1. Wstęp teoretyczny

Aby przeprowadzić symulację zostały wybrane 2 algorytmy spośród algorytmów przydziału czasu procesora, oraz 2 spośród algorytmów zastępowania stron. Pierwszy algorytm z pierwszej kategorii to FCFS, w którym procesor otrzymuje każdy proces w kolejności, w jakiej zgłosiły się do systemu. Można go porównać do świata rzeczywistego, i jest używany w instytucjach publicznych (tj. Sklepy, urzędy, itp.). Drugi z algorytmów to SJF, w którym to procesor skupia się na procesach, które mają najkrótszy czas wykonania. Ten typ algorytmu również może zostać wykorzystany w świecie rzeczywistym. Niejednokrotnie bowiem zdarza się, że osoba, która ma o wiele mniej produktów w koszyku jest przepuszczana do kasy w sklepie.

Pierwszy z algorytmów zastępowania stron, który został użyty w symulacji to FIFO. Strony znajdujące się w pamięci fizycznej komputera tworzą kolejkę. Każda kolejna strona jest wprowadzana do pamięci na koniec kolejki, natomiast "ofiara", do usunięcia znajduje się zawsze na jej początku. Ten typ algorytmu nie bierze pod uwagę właściwości kolejnych obsługiwanych procesów, a strony zastępowane są w sposób losowy. Drugi z algorytmów zaimplementowanych w projekcie to Algorytm drugiej szansy. Podobnie jak w poprzednim przypadku strony tworzą kolejkę, a każda następna strona jest wprowadzana do pamięci na skutek odwołania do niej. Oznacza to, że jeśli wstawimy stronę na koniec naszej kolejki, to wówczas jej bit odwołania jest równy 1. Gdy "polujemy na ofiarę", wówczas wybór pada na początek kolejki, gdzie możliwe są dwa przypadki.

- Jeśli bit odwołania jest równy 0, to wówczas strona staje się ofiarą
- Jeśli bit odwołania jest równy 1, to wówczas taka strona dostaje drugą szansę

2. Procedura testowania

Do implementacji wykorzystano język Python i najpierw stworzone zostały algorytmy FCFS, oraz SJF. Pierwszy z programów został napisany przy użyciu modułu "namedtuple", który znajduje się w module "collections". Następnie z bazy plików z danymi testowymi dla każdego pliku tworzony jest obiekt procesu, który zawiera czas przybycia oraz wykonania procesu. Ostatnim krokiem jest policzenie w pętli odpowiednich parametrów dla każdego procesu (obiektu) a następnie uśrednienie wyników.

Drugi z podanych algorytmów wykorzystuje moduł "dataclasses" wprowadzony w 3.7 wersji Python'a. Tym razem utworzona została klasa dla całego jednego procesu ze wszystkimi potrzebnymi danymi. Następnie, podobnie jak w poprzednim przykładzie 100 plików z danymi zostaje wczytanych, a następnie liczone są potrzebne parametry dla każdego z procesów. Po zakończonej pracy program sam uśrednia wyniki czasów oczekiwania oraz przetwarzania.

W drugiej części przygotowanych algorytmów implementacja wygląda nieco inaczej. W algorytmie FIFO wykorzystany został moduł Queue z biblioteki queue, która ma za zadanie zasymulować rzeczywistą kolejkę, na podstawie której algorytm oblicza ilość brakujących stron. Do tego typu algorytmów również zostały odpowiednio przygotowane dane, które są wczytywane z pliku w pętli, a następnie wywoływana jest funkcja odpowiedzialna za zliczanie odpowiedniego parametru w danym ciągu losowych liczb. Na koniec program uśrednia wynik dzieląc całkowitą liczbę brakujących stron przez ilość wszystkich ciągów.

Ostatni przygotowany algorytm to Algorytm Ostatniej Szansy. Ten z kolei program został podzielony na trzy funkcje. Pierwsza ma za zadanie przypisać stronie drugą szansę zmieniając wartość logiczną w tablicy. Druga funkcja służy po to, aby aktualizować strony w pamięci, natomiast zadaniem trzeciej funkcji jest zliczanie brakujących stron korzystając z dwóch funkcji pomocniczych opisanych wcześniej. Podobnie jak w przypadku poprzednich algorytmów następnie następuje wczytanie danych z pliku w pętli, obliczenie odpowiednich parametrów, a na końcu ich uśrednienie.

3. Wyniki symulacji

Następnym krokiem było uruchomienie programów w celu wyliczenia odpowiednich wartości parametrów danych procesów.

Dla algorytmów przydziału czasu procesora parametry prezentują się następująco:

- SJF:
 - Średni czas oczekiwania: 15861,9123s
 - Średni czas cyklu przetwarzania: 15856.9301s
- FCFS:
 - Średni czas oczekiwania: 24670,835s
 - Średni czas cyklu przetwarzania: 24675,8294s

Uśrednione wyniki parametrów dla algorytmów zamiany stron:

- Dla liczby ramek $R = 3$
 - FIFO:
 - Średnia ilość brakujących stron: 15,6989
 - Algorytm drugiej szansy:
 - Średnia ilość brakujących stron: 15,7007
- Dla liczby ramek $R = 5$
 - FIFO:
 - Średnia ilość brakujących stron: 13.4258
 - Algorytm drugiej szansy:
 - Średnia ilość brakujących stron: 13.4243
- Dla liczby ramek $R = 7$
 - FIFO:
 - Średnia ilość brakujących stron: 11,8025
 - Algorytm drugiej szansy:
 - Średnia ilość brakujących stron: 11,7964

4. Wnioski

Dla obu algorytmów przydziału czasu procesora uśrednione wartości szukanych parametrów budzą pewne wątpliwości. Tak duże wartości mogą wynikać z faktu, iż dane, które są używane do przetworzenia przez program są generowane przez komputer w sposób totalnie losowy, przez co nie ma realnego wpływu na otrzymane wartości, wobec czego nie powinno się ich traktować jako wartości referencyjnych i sugerować nimi przy dalszych rozważaniach. Mimo dużych rozbieżności między pomiędzy stanem faktycznym a otrzymanymi wynikami, można również zauważyć, że dla algorytmu SJF wartości są około 60% mniejsze niż dla algorytmu FIFO. Może być to spowodowane drobnymi różnicami w implementacji obu algorytmów.

Dla algorytmów zastępowania stron otrzymano o wiele bardziej "przyjemne wyniki". Oba algorytmy zostały uruchomione 3 razy, dla różnej wartości ramek. W tym przypadku ich liczba była równa 3, a także 5 oraz 7. Można zauważyć, że wraz ze wzrostem liczby ramek maleje średnia liczba brakujących stron. Liczba ramek zmieniała się o 2, i również (w przybliżeniu) o taką wartość zmieniały się średnie wyniki dokonanych obliczeń.