

Systemy operacyjne – laboratorium

1. Wstęp

Celem niniejszego sprawozdania jest porównanie dwóch algorytmów kolejkowania procesora. Projekt zakłada napisanie symulacji tego procesu przy użyciu dowolnie wybranego języka programowania. W niniejszym projekcie wykorzystano język Python 3.

Na przestrzeni lat pojawiło się wiele algorytmów kolejkowania procesów, spośród których można wymienić np.:

- FCFS (First Come First Served)
- SJF (Shortest Job First)
- HRRN (Highest Response Ratio First)
- Round Robin
- Szeregowanie loteryjne
- I wiele innych

Należy wspomnieć, że każdy z powyższych algorytmów może występować w wersji wywłaszczeniowej i w wersji niewywłaszczeniowej. Do testów algorytmów kolejkowania wybrano algorytmy FCFS i SJF w wersjach niewywłaszczeniowych. Opis przeprowadzonych eksperymentów znajduje się części drugiej sprawozdania.

2. Porównanie algorytmów kolejkowania

FCFS – algorytm kolejujący procesy według czasu przybycia

SJF – algorytm, kolejujący procesy według czasu trwania procesów, krótsze procesy obsługuje przed procesami o dłuższym czasie trwania.

W projekcie wykorzystano plik odpowiedzialny za generowanie obiektów w postaci listy zawierającej trzy wartości integer, opisujące w kolejności:

- czas przyścia procesu
- czas trwania procesu
- czas oczekiwania procesu, który w momencie tworzenia obiektu zawsze wynosi 0, a wzrasta dopiero od momentu zakolejkowania procesu.

Moduł ten, o nazwie *process_generator.py* służący do generowania wartości pseudolosowych wykorzystuje moduł *random* znajdujący się w bibliotece standardowej języka Python, oraz moduł *NumPy*, który jest biblioteką zewnętrzną. Główną funkcjonalność modułu zapewnia funkcja *generate_processes*, która jako argument przyjmuje liczbę procesów jaką ma wygenerować. Moduł dostarcza również funkcję pozwalającą na zapis wygenerowanych list do pliku, choć ta funkcjonalność nie była bezpośrednio wykorzystana w projekcie, gdyż wygenerowane listy, były bezpośrednio procesowane przez funkcje kolejkujące, nie wczytywały one danych z pliku.

Wybrane algorytmy były sprawdzane dla różnych ilości procesów (10, 100, 1000) oraz dla różnych czasów wykonania, o średnim czasie wykonania 10, 20 i 30 jednostek czasu z odchyleniem standardowym 5.

Należy spodziewać się, że algorytm SJF zapewni krótszy czas wykonywania wszystkich procesów, poprzez skrócenie ogólnego czasu oczekiwania. Wyniki eksperymentu przedstawiono w tabeli nr.1.

Tabela 1 Zestawienie czasów oczekiwania dla procesów poddanych kolejkowaniu algorytmami FCFS i SJF

	Śr. 10, odchylenie 5			Śr. 20, odchylenie 5			Śr. 30, odchylenie 5		
Liczba procesów	10	100	1000	10	100	1000	10	100	1000
FCFS, śr. czas o.	36,4	469,42	4468,1	95,8	913,42	9447,1	115,1	1376,8	14381
SJF, śr. czas o.	29,7	347,63	3170,8	77,1	771,67	8049,9	110,6	1236,6	12970
różnica	-18%	-26%	-29%	-20%	-15%	-15%	-4%	-10%	-10%
Śr. różnica	-16,3%								

3. Wnioski

W sprawozdaniu pokazano porównanie średniego czasu wykonania określonej liczby procesów, przychodzących w różnym czasie i mających różny czas wykonania. Jak należało się spodziewać algorytm FCFS, który wykonuje procesy zgodnie z kolejnością ich przyjścia okazał się wolniejszy od algorytmu SJF. Algorytm FCFS nie optymalizuje wykonywania procesów poprzez minimalizację czasu ich wykonania. Jak pokazano w tabeli nr 1, mogą zdarzyć się przypadki, w których różnica w czasie oczekiwania procesów, a więc i w czasie wykonania całości operacji jest mała, rzędu paru procent. Jest to kwestia, przychylnego stosunku, czasu przyjścia danych procesów do czasu ich trwania. Aby zniwelować wpływ takich przypadków na analizę, porównując dwa algorytmy powinniśmy brać pod uwagę średnią różnicę w czasie oczekiwania procesów. W ramach przeprowadzonych eksperymentów różnica wyniosła 16,3% na korzyść algorytmu SJF. Jest to wyraźna różnica, która dla innych parametrów wejściowych mogłaby być jeszcze większa. W ramach dalszej pracy nad projektem warto przeprowadzić większą ilość eksperymentów np. dla tysiąca, czy nawet miliona prób, oraz dla różnych wartości zadanych czasu przyjścia oraz czasu trwania procesów.