

Wykonany przeze mnie program symuluje algorytm FCFS, SJF oraz planowania priorytetowego. W pliku process umieszczone są dane procesów, w następującym porządku:

-nazwa_procesu czas_trwania_procesu czas_rozpoczecia priorytet

```
p1: 100% |██████████| 7/7 [00:07<00:00, 1.00s/it]
p6: 100% |██████████| 2/2 [00:02<00:00, 1.00s/it]
p4: 100% |██████████| 3/3 [00:03<00:00, 1.00s/it]
p5: 100% |██████████| 3/3 [00:03<00:00, 1.00s/it]
p3: 100% |██████████| 5/5 [00:05<00:00, 1.00s/it]
p2: 100% |██████████| 6/6 [00:06<00:00, 1.00s/it]
```

FCFS (ang. *first-come, first-served*) oznacza "pierwszy zgłoszony - pierwszy obsłużony".

Algorytm ten opiera się na prostej zasadzie, którą właściwie wyjaśnia jego nazwa. Najlepiej do implementacji algorytmu nadaje się kolejka FIFO, w której blok kontrolny procesu wchodzącego do kolejki dołączany jest na jej koniec, zaś wolny procesor przydzielany jest procesowi z początku kolejki.

SJF (ang. *shortest-job-first*), czyli "najkrótsze zadanie najpierw", opiera swe działanie na powiązaniu procesu z długością jego najbliższej fazy. Procesor zostaje przydzielony procesowi, który ma najkrótszą następną fazę procesora. W przypadku, gdy dwa procesy mają następną fazę procesora równe, stosuje się algorytm FCFS.

Planowania priorytetowego jest ogólnym przypadkiem algorytm SJF. Zamiast długości następnej fazy z algorytmu SJF, procesowi przypisuje się priorytet. Procesy o wyższym priorytecie mają pierwszeństwo w dostępie do procesora, przed procesami o priorytecie niższym. Procesom o równych priorytetach przydziela się procesor zgodnie z algorytmem FCFS.

Dane testowe:

Nazwa	czas_trwania_procesu	czas_rozpoczecia	priorytet
P1	7	0	2
P2	6	2	3
P3	5	4	1
P4	3	2	5
P5	3	4	3
P6	2	3	2

FCFS

Total wait time= 62

Average process wait time: 10.34

SJF

Total wait time= 48

Average process wait time: 8.0

PRIORITY

Total wait time= 61

Average process wait time: 10.16

Najszybszym z algorytmów okazał się algorytm SJF.