FCFS żądania realizowane są w kolejności przyjścia

SSTF Obsługuje działanie najbliższe obecnej pozycji głowicy

Scan idzie od początku do końca i potem zawraca

Kiedy w C Scan dojdziemy do końca przesuwamy czytnik na początek, nie liczymy z tego powodu dodatkowego czasu.

Żądania real time EDF, FD-SCAN

Wybieramy jeden algorytm z powyższych i dodajemy do niego EDF, FD-SCAN

Implementujemy każdy z 4 algorytmów i na wybrany nakładamy strategię real time

Jeśli nie ma żądań real time to używamy po prostu zwykłego algorytmu

Żądania real time mają deadline

Jeśli jesteśmy za daleko od zdażenia real time to go nie obsłużymy, można zrobić counter dla niewykonanych

EDF jest naiwny i nawet kiedy żądanie real time jest za daleko przesuwa głowicę do niego, kiedy żądanie real time się nie doczeka to EDF szuka sobie nowego celu.

Nawet kiedy dłuższy deadline np. 10 jest bliżej niż 5 to EDF idzie do tego drugiego i dopiero kiedy ono umrze próbuje zawrócić do pierwszego, może nie zdążyć ☹

Można znowu zastosować rozkład prawdopodobieństwa

FD SCAN sprawdza czy da się dotrzeć do danego deadlinu i idzie tylko wtedy kiedy jest możliwe jego osiągnięcie

Szukamy najszybszego deadlinu ale tylko takiego który możemy zrealizować

Możemy zdecydować czy idziemy do najszybciej kończącego się deadlinu czy do tego który jest obecnie najbliżej głowicy.

FD SCAN ściąg wszystko po drodze

EDF oraz FD SCAN zaczynają działać w momencie kiedy pojawią się żądania typu real time, tak długo jak one występują

Kiedy nie ma tego typu żądań działamy zgodnie z algorytmem do którego je podpięliśmy

Oba algorytmy real time implementujemy do jednego z głównych algorytmów

Stosunek real time do zwykłych : najlepiej nie więcej niż 1:3.

Jakie statystyki

Liczba nieobsłużonych real time

Liczba przesunięć głowicy

Ewentualnie waiting time dla SSTF

Symulacje kończymy po n żądaniach