

### Algorytmy przydziału czasu procesora

- Planowanie metodą FCFS - "pierwszy zgłoszony - pierwszy obsłużony"
- Planowanie metodą "najpierw najkrótsze zadanie"
- Planowanie priorytetowe
- Planowanie rotacyjne.

[http://pl.wikipedia.org/wiki/Dzielenie\\_czasu](http://pl.wikipedia.org/wiki/Dzielenie_czasu)

[http://pl.wikipedia.org/wiki/Algorytm\\_szeregowania](http://pl.wikipedia.org/wiki/Algorytm_szeregowania)

<http://www.prz.rzeszow.pl/we/katedry/zsc/projekty/2000/0/index.html>

<http://wazniak.mimuw.edu.pl/index.php?title=SO-1st-2.3-w3.tresc-1.0-toc>

### Algorytmy zastępowania stron pamięci wirtualnej

- Algorytm FIFO
- Algorytm optymalny.
- Algorytm LRU.
- Algorytmy przybliżające metodę LRU.
- Algorytm dodatkowych bitów odwołań.
- Algorytm drugiej szansy.
- Ulepszony algorytm drugiej szansy.
- Algorytmy zliczające.
- Algorytm buforowania stron.

[http://pl.wikipedia.org/wiki/Stronicowanie\\_pami%C4%99ci](http://pl.wikipedia.org/wiki/Stronicowanie_pami%C4%99ci)

<http://www.prz.rzeszow.pl/we/katedry/zsc/projekty/2000/1/index.html>

<http://wazniak.mimuw.edu.pl/index.php?title=SO-1st-2.3-w5.tresc-1.0-toc>

<http://wazniak.mimuw.edu.pl/index.php?title=SO-1st-2.3-w6.tresc-1.0-toc>

### Algorytmy przydziału miejsca na dysku

- Przydział ciągły
- Przydział listowy
- Przydział indeksowy

<http://www.prz.rzeszow.pl/we/katedry/zsc/projekty/2000/2/index.html>

<http://wazniak.mimuw.edu.pl/index.php?title=SO-1st-2.3-w9.tresc-1.0-toc>

[http://www.dsod.p.lodz.pl/student/wyklady/pol/SO\\_04.pdf](http://www.dsod.p.lodz.pl/student/wyklady/pol/SO_04.pdf) od str. 43

### Algorytmy planowania dostępu do dysku

- FCFS
- SSTF
- SCAN
- C-SCAN
- LOOK

<http://www.prz.rzeszow.pl/we/katedry/zsc/projekty/2000/3/dysk.html>

<http://student.pb.bielsko.pl/~dkulpinski/systemy.doc>

[http://www.dsod.p.lodz.pl/student/wyklady/pol/SO\\_04.pdf](http://www.dsod.p.lodz.pl/student/wyklady/pol/SO_04.pdf) od str. 35

### Algorytmy defragmentacji dysku twardego

Algorytmów i sposobów defragmentacji jest wiele i przeważnie firmy tworzące programy defragmentujące nie chwala się, w jaki sposób realizują to zadanie.

Dlatego w przypadku tych algorytmów pozostawiam dowolność, co do implementacji, nawet na zasadzie własnego pomysłu.

Sama idea ma być taka:

- Ustalamy dowolny rozmiar dysku,
- Ustalamy rozmiar pamięci RAM potrzebnej do defragmentacji – im mniej tym lepiej,
- Zakładamy, że pliki na dysku są rozmieszczone z wykorzystaniem przydziału listowego,
- Wykorzystując daną metodę defragmentacji (należy zastosować minimum 3 – można zastosować 3 własne metody), zmieniamy rozmieszczenie bloków danych na dysku, tak, by każdy plik zajmował miejsce na dysku w sposób ciągły.
- Porównujemy szybkość tych metod.

Oczywiście całość operacji jest symulowana na tablicy dwu-wymiarowej.

[http://www.zsnr9.neostrada.pl/soisk/klasa\\_1/2\\_systemy\\_plikow/soisk\\_34\\_zjawisko\\_fragmentacji\\_dysku\\_defragmentacja.doc](http://www.zsnr9.neostrada.pl/soisk/klasa_1/2_systemy_plikow/soisk_34_zjawisko_fragmentacji_dysku_defragmentacja.doc)

### Algorytmy detekcji, unikania i usuwania zakleszczeń

<http://wazniak.mimuw.edu.pl/index.php?title=SO-1st-2.3-w13.tresc-1.0-toc>

<http://wazniak.mimuw.edu.pl/index.php?title=SO-1st-2.3-w14.tresc-1.0-toc>

### Algorytmy działania semaforów

<http://wazniak.mimuw.edu.pl/index.php?title=SO-1st-2.3-w12.tresc-1.0-toc>

Na stronach Politechniki Rzeszowskiej <http://www.prz.rzeszow.pl/we/katedry/zsc/projekty/2000/index.html> widoczne są przykładowe skrypty lub aplety Javy do każdego z algorytmów. Naszym celem jest podobna implementacja, jednak z większymi możliwościami.