

Szymon Kubiszewski

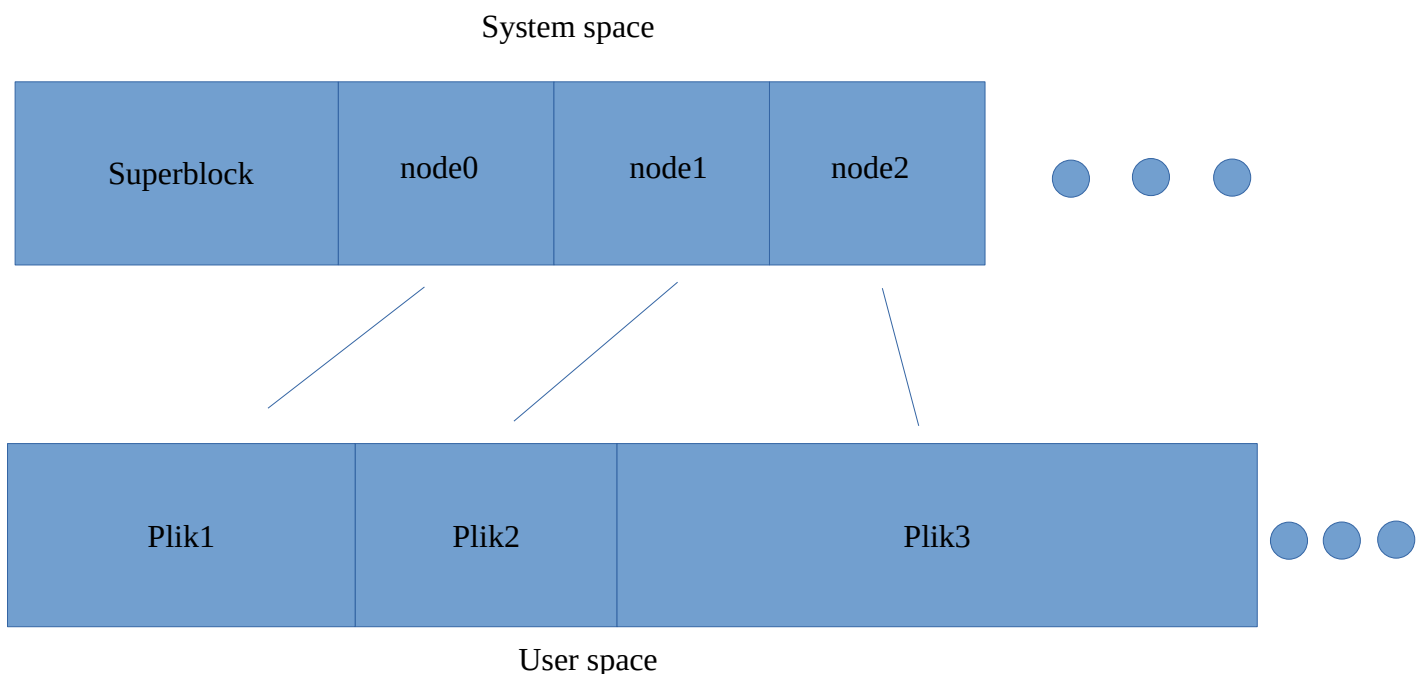
Wirtualny dysk został podzielony na dwa bloki, systemowy oraz użytkownika. W bloku systemowym zajmującym domyślnie ok 1/8 całkowitej pojemności znajdują się dwa typy struktur:

superblok (superblock) – nadrzędny, unikalny podblok zawierający ogólne informacje o pliku, znajduje się na początku pliku dysku

oraz

węzły (node) – zawierają informacje o przypisanych im blokach pamięci, rozmiarach tych bloków oraz nazwach plików

resztę czyli około 7/8 zajmuje blok użytkownika, czyli taki w którym można umieszczać dane. Blok ten nie jest podzielony na podbloki, a odniesienia do konkretnych jego obszarów znajdują się w węzłach.



superblock zawiera pola:

size – rozmiar VD (dysku wirtualnego)

total\_nodes – łączna ilość węzłów == maksymalna ilość plików

full\_nodes – aktualna ilość plików w dysku

free\_space – pozostała ilość wolnego miejsca w obszarze user

next\_free\_node – adres następnego wolnego węzła (usunięte pliki się w to nie wliczają)

next\_free\_addr – adres wolnego bloku pamięci

node zawiera pola:

name – nazwa pliku do którego node zawiera odwołanie

size – rozmiar pliku

address – adres początku zajętego przez plik bloku pamięci

W proponowanym przeze mnie rozwiązaniu nie występuje fragmentacja wewnętrzna, jednakże występuje fragmentacja zewnętrzna, po zwolnieniu zajętego bloku pamięci może on zostać ponownie użyty, podobnie węzeł który na niego wskazywał, jednak gdybyśmy chcieli przykładowo usunąć Plik2 na przedstawionym wyżej schemacie, a w jego miejsce wstawili Plik4 o rozmiarze mniejszym niż Plik2, różnica obszarów, a więc część znajdująca się „pod koniec” pliku2 przestanie być możliwa do użycia.

createVD(size) – tworzy plik wirtualnego dysku w rozmiarze size\*1024 B, nazwa utworzonego pliku to VD

deleteVD() - usuwa plik wirtualnego dysku

copyToVD(name) – kopiuje plik o nazwie name do VD

copyFromVD(name) – kopiuje plik o nazwie name z VD

deleteFile(name) – usuwa plik o nazwie name z VD

map() - wyświetla aktualne informacje o dysku – rozmiary bloków user/system oraz zajęte bloki pamięci i ich rozmiary na dysku

view() - wyświetla pliki znajdujące się na VD

Opis programów testowych:

normalAdd – pokazuje poprawne działanie umieszczania plików w VD

smallVD – pokazuje co się dzieje gdy chcemy umieścić parę plików z których nie wszystkie się zmieszczą

fragm – pokazuje przykład fragmentacji – po dodaniu 3 plików i usunięciu 2 z nich, na jego miejsce dodajemy plik mniejszy od początkowego – po pliku nowo umieszczonym powstaje blok niedostępnej pamięci

fileTooBig - pokazuje próbę dodania pliku większego od całego VD

bigVD – pokazuje poprawne dodanie wszystkich przygotowanych plików testowych do odpowiednio dużego VD

(make) checkCopying – tw./noorzemy VD do którego dodajemy plik1.txt, następnie usuwamy ten plik z poziomu systemu operacyjnego i kopiujemy go z VD w miejsce z którego początkowo go kopiowaliśmy

addSame – próbujemy dwa razy dodać plik1.txt