

## Temat: Operacje na pliku.

W oparciu o książki:

1. Maurice J. Bach - „Budowa systemu operacyjnego UNIX”, WNT, W-wa, 1995,
2. W. G. Olifier, N. A. Olifier - „Sietewyje opieracionnyje sistemy”, Izd. Dom „Pitier”, 2001.

### Funkcje systemowe systemu plików.

Przekazanie deskryptora pliku	Użycie namei	Przydział i-węzłów	Atrybuty pliku	WE/Wy na pliku	Struktura systemu plików	Obsługa drzewa
open creat dup pipe close	open, stat, creat, link, chdir, unlink, chroot, mknod, chown, mount, chmod, umount	creat mknod link unlink	chown chmod stat	read write lseek	mount umount	chdir chown

### Niskopoziomowe algorytmy systemu plików.

namei	ialloc ifree	alloc free bmap
iget iput		

### Algorytmy przydziału bufora.

getblk brelse bread breada bwrite
-----------------------------------

SO tworzy na początku swego działania pewną ilość i-węzłów w PAO, buforów w PAO, w partycji jest tworzona tabela i-węzłów oraz bloki. W trakcie działania SO do swoich potrzeb tworzy m. in. listy:

- wolnych wirtualnych i-węzłów oraz i-węzłów
- wolnych buforów
- wolnych bloków.

Przy operacjach na plikach potrzebne zasoby brane są z tych list, zwalniane dopisywane do tych list.

### 1. Odczyt zawartości pliku.

Niskopoziomowe algorytmy systemu plików: algorytm namei – wykorzystując algorytmy:

iget (wyszukuje – czy plik nie był już otwarty przez inny proces:

– jeżeli nie, to przydziela wolny tzw. wirtualny i-węzeł – te i-węzły istnieją w PAO, wczytuje do niego dane z i-węzła z tablicy i-węzłów, zakłada blokadę (dostępu) - że plik jest wykorzystywany przez proces,

- jeżeli tak – wyszukuje przydzielony dla niego wirtualny i-węzeł, sprawdza – czy nie ma blokady, zwiększa licznik odwołań do pliku o 1),

iput (zmniejsza licznik odwołań o 1, jeżeli licznik staje się=0 - to – jeżeli zaszły zmiany w wirtualnym i-węzle lub w treści pliku – zapisuje zmiany do i-węzła w tablicy i-węzłów, zwalnia wirtualny i-węzeł,

bmap (ustawia parametry jądra podczas dostępu do pliku)

– namei przekształca ścieżkę dostępu w i-węzeł szukanego zasobu.

Np. polecenie *cat /etc/passwd* – najpierw iget udostępnia i-węzeł korzenia (jest on przechowywany w PAO) – jądro sprawdza uprawnienia procesu – czy może on to robić – jeżeli tak przegląda tabelę opisującą zawartość katalogu / - jeżeli znajdzie etc zwalnia i-węzeł dla / (iput) i przydziela i-węzeł

dla etc (iget), itd. Po znalezieniu i-węzła pliku *passwd* przydzielane są bufor pod dane pliku, w oparciu o dane w i-węźle odszukiwane są bloki pliku na dysku, następnie te bloki są wczytywane z dysku do buforów, następnie zawartości buforów wyświetlane są na ekranie w postaci przetworzonej przez program *cat*.

## 2. Utworzenie pliku.

Algorytm *alloc* przydziela nowy i-węzeł w tabeli i-węzłów – wyszukuje wolne i-węzły na dysku, tworzy listę wolnych w PAO, przydziela jeden z nich dla pliku, przydziela wirtualny i-węzeł w PAO dla tego pliku, wprowadza inf. początkową o pliku do wirtualnego i-węzła, zapisuje tą inf. na dysku (w tabeli i-węzłów), przydzielane są bufor pod zawartość pliku, program tworzący plik zapełnia je treścią, jeżeli będzie polecenie „zapisz do pliku” to algorytm *alloc* przydzieli bloki na dysku, do nich będzie zapisana zawartość buforów. Po zamknięciu pliku algorytm *ifree* zwalnia i-węzeł w PAO

W systemach plików typu *s5* lub *ufs* - (SO UNIX) – struktura systemu plików wygląda następująco:

boot record
superblok
tabela i-węzłów
obszar danych

Superblok (blok identyfikacyjny systemu plików) zawiera szereg danych, m. in. inf. o systemie plików, jego rozmiar, ilość i-węzłów, listę wolnych i-węzłów, listę wolnych bloków.

Algorytm *alloc* przydziela bloki na dysku pod dane pliku, przydziela też bufor pod ten blok – jądro będzie pracowało z buforem – aż do momentu zapisu danych pliku na dysk.

Algorytm *free* – zwalnia bloki.

## 3. Usuwanie pliku.

Usuwanie pliku – w wersji podstawowej – np. *rm ./dane* – polega na usunięciu zapisu o tym pliku w pliku opisującym zawartość katalogu – w którym ten plik jest:

Nazwa symboliczna	I-węzeł
<del>dane</del>	<del>1022</del>
dane12	1222

Efekt - i-węzeł jest zwolniony – jest do wykorzystania pod nowy plik, bloki zawierające treść pliku są zwalniane (bez „zerowania”) – do ponownego wykorzystania pod dane innego pliku – czyli ew. możliwość odczytania do pewnego momentu.

Dane w i-węźle na dysku istnieją – do momentu ponownego wykorzystania tego i-węzła pod inny plik. Dane w bloku istnieją do momentu nadpisania – zapisania nowej treści do bloku – bloki zapisywane są w całości – jeżeli nowa treść zajmie tylko część bloku – i tak stara do końca bloku zostanie usunięta.

Są polecenia nie tylko usuwające zapis o pliku z zawartości katalogu, ale również nadpisujące dane usuwanego pliku – np. *shred*.

Ze strony <https://linuxiarze.pl/trwale-usuwanie-plikow-i-katalogow-w-linuksie/>

„Aby zapoznać się z opisem programu i funkcjami, w oknie konsoli wpisujemy polecenie:

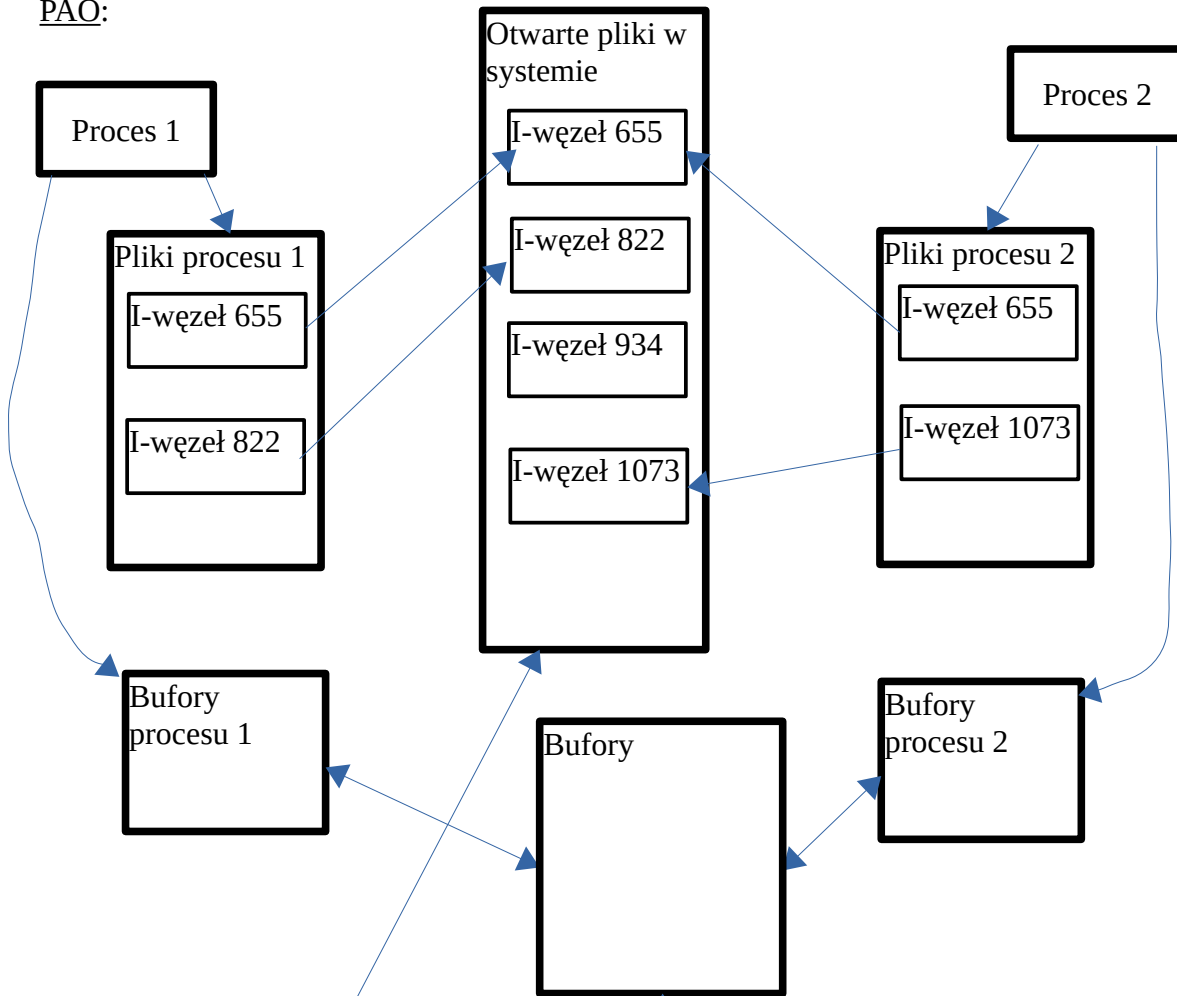
shred --help

Najbardziej będą nas interesowały 4 opcje:

- f, -force – wymuszenie nadpisania pliku.
- u, -remove – skasowanie pliku po zamazaniu.
- z, -zero – dodatkowe zamazanie zerami, aby ukryć zamazywanie.
- n, -iterations=N – zamazanie N razy zamiast domyślnych 3 razy.”

### Struktury obsługujące otwarte pliki w PAO (uproszczony schemat poglądowy).

PAO:



Dysk:

