

Systemy operacyjne

Ćwiczenia 9

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Stallings: 12
- Tanenbaum: 4.1 – 4.5
- Silberschatz: 11, 12

Zadanie 1

Plik jest podstawową jednostką organizacji danych w systemach komputerowych. Jakie **atrybuty** (lub **metadane**) i **operacje** mogą być skojarzone z plikiem? Co to znaczy, że dany zasób (np. urządzenie, gniazdko, potok) ma **semantykę pliku**. W jaki sposób, w systemach UNIX, rozwiązano problem dostępu do niskopoziomowych funkcjonalności zasobów ukrytych za interfejsem pliku?

Zadanie 2

Opisz pobieżnie trzy metody przydziału przestrzeni dyskowej dla pliku: **ciągła**, **listowa**, **indeksowana**. Jakie struktury danych są wykorzystywane do::

- zapewnienia dostępu do zawartości pliku na dysku?
- przechowywania informacji o wolnych obszarach?

Jaki jest narzut pamięciowy utrzymywania tych struktur? Jakie są konsekwencje częściowego ich uszkodzenia – czy i jak można naprawić błędy? Na czym polega **defragmentacja**?

Zadanie 3

Czemu systemy uniksowe wprowadziły interfejs **VFS**? Co wyrażają struktury **inode**, **dentry**, **superblock**? W jaki sposób są reprezentowane: plik i katalog (jakimi operacjami na nich dysponujemy?) oraz **hierarchiczna struktura katalogów**? Czym jest **ścieżka absolutna**, a czym **relatywna**? Jak implementowane są **dowiązania twarde** (ang. *hard link*), a jak **dowiązania symboliczne** (ang. *soft link*)? Czym charakteryzują się **punkty montażowe**?

Zadanie 4

Opisz uniksowy **system kontroli dostępu** do plików. Jakie jest znaczenie bitów **rxw** dla plików, a jakie dla katalogów? Znajdź informacje nt. dodatkowych bitów: **set-uid**, **set-gid**, **sticky**. Na czym polega rozszerzenie systemu uprawnień **ACL** (ang. *access control list*)? Podaj przykład, w którym zastosowanie ACL jest niezbędne, a standardowy system uprawnień jest zbyt ograniczony. W jaki sposób kontrola dostępu implementowana jest w strukturze **inode**?

Zadanie 5

Jaka historyczna motywacja stała za wyróżnieniem pojęć plików o **dostępie sekwencyjnym** i **dostępie swobodnym**? W jaki sposób system może korzystać **pamięci podręcznej dla bloków** (ang. *block cache*) do przyspieszenia operacji odczytu i zapisu? Dlaczego, nawet gdy w buforze są jeszcze wolne bloki, należy zrzucić zawartość niektórych struktur **inode** na dysk? Dlaczego w systemie Linux pamięć podręczna dla bloków zintegrowana jest z pamięcią podręczną dla stron (ang. *page cache*)?

Zadanie 6

Jak **księgowanie** pomaga zachować **spójność** struktur systemu plików? Czy metoda ta gwarantuje zachowanie **spójności** danych w plikach? Czym różni się **księgowanie metadanych** od **księgowania danych**? Księgowanie nie chroni przed awariami sprzętowymi – w tym celu korzystamy z **kopii zapasowych**. Jakie są różnice między **kopią pełną**, **przyrostową** i **różnicową**? Czy **migawki** mogą pełnić rolę kopii zapasowych? W jakim kontekście sprawdzają się / są wykorzystywane **wersjonujące systemy plików**?

Zadanie 7

Opisz najczęściej wykorzystywane struktury organizacyjne pliku – **sterta**, **plik sekwencyjny**, **indeksowany**, **indeksowany sekwencyjnie**, **haszowany**. Które z wymienionych struktur:

- są najefektywniejsze do składowania rekordów o stałej lub zmiennej długości?
- charakteryzują się najdłuższym czasem aktualizacji zawartości rekordów, jeśli rozmiar nowego rekordu jest (a) taki sam lub (b) większy?
- umożliwiają szybki (losowy) dostęp do pojedynczego rekordu lub sekwencji rekordów?

Zadanie 8

Wybierz odpowiednią strukturę pliku wymienioną w poprzednim zadaniu do efektywnej realizacji następujących dyskowych struktur danych:

- logi tekstowe, tj. ciąg rekordów o zmiennej długości; brak modyfikacji i blokowania rekordów; szybkie przeglądanie tj. operacje `next` i `prev`; możliwość dopisywania kolejnych linii na końcu pliku,
- pamięć podręczna przeglądarki internetowej, tj. tablica haszująca `string → string`, gdzie kluczem jest URL, a wartością para: czas ostatnich odwiedzin i treść dokumentu; ma być możliwe usuwanie rekordów; można założyć, że klucz jest stałej długości.

Zaproponuj wewnętrzną strukturę rekordów. Możliwe, że będzie potrzebny więcej niż jeden plik do realizacji struktury. Przyjmij, że pliki można zmapować wywołaniem `mmap` w przestrzeni użytkownika oraz zmieniać rozmiar obszaru wywołaniem `mremap`.