

Systemy operacyjne

Ćwiczenia 8

Zadanie 1

Rozważmy proste operacje na plikach: `read` i `write`, przy założeniu, że plik został już otwarty. Opisz ścieżkę obsługi żądania po stronie systemu operacyjnego. Wymień poszczególne podsystemy i opisz funkcje które pełnią. W szczególności w odpowiedzi powinny pojawić się: system plików, kontrola dostępu, blokowanie rekordów, pamięć podręczna dysku, sterownik urządzenia blokowego.

Zadanie 2

Opisz najczęściej wykorzystywane struktury organizacyjne pliku (tj. sterta; plik sekwencyjny, indeksowany, indeksowany sekwencyjnie, haszowany). Które z wymienionych struktur:

- są najefektywniejsze do składowania rekordów o stałej lub zmiennej długości rekordów?
- 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 3

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`.

Zadanie 4

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

- położeniu bloków danego pliku na dysku?
- wolnych blokach?

Jaki jest narzut pamięciowy utrzymywania tych struktur? Jakie są konsekwencje częściowego ich uszkodzenia? Czy jest możliwe ich naprawienie - jeśli tak, to w jaki sposób?

Zadanie 5

Katalogi można zaimplementować jako specjalne pliki, które mogą być obsługiwane przez pewien ustalony interfejs programisty, albo jako regularne pliki zawierające dane. Jakie zalety i wady widzisz w obu rozwiązaniach?

Zadanie 6

Wyjaśnij krótko fragmentację w kontekście systemów plików. Dlaczego jest szkodliwa? Wymień techniki zapobiegania fragmentacji. Zaproponuj algorytm defragmentacji działający na systemie plików z przydziałem listowym zakładając, że można blokować dostęp do całych plików.

Zadanie 7

Opisz uniksowy system kontroli dostępu do plików. Jakie jest znaczenie bitów `rwX` 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*) i jak się je stosuje?

Zadanie 8

Wyjaśnij różnicę między doradczym (ang. *advisory*) i obowiązkowym (ang. *mandatory*) blokowaniu plików. Zakładając, że można blokować dowolne fragmenty pliku (tj. plik ma strukturę sterty o zmiennej długości rekordów) zaproponuj strukturę danych, która będzie efektywnie przechowywać informacje o blokadach i umożliwiać szybkie wyszukiwanie czy podany obszar jest zablokowany na odczyt lub zapis.

Zadanie 9

W obrębie standardowego wykładu ze struktur danych omawiane są B-drzewa. Opisz je po krótku. Czemu są tak przydatne w przypadku projektowania dyskowych struktur danych? Podaj główne różnice między B-drzewami i [B+-drzewami](#)?

Zadanie 10

Jak księgowanie (ang. *journalling*) pomaga zachować spójność struktur systemu plików? Czy metoda ta gwarantuje zachowanie spójności danych w plikach?