

Systemy operacyjne

Lista zadań nr 11

Na zajęcia 7 stycznia 2021

Przed przystąpieniem do rozwiązywania zadań zapoznaj się z dokumentem: [The Second Extended File System Internal Layout](https://www.nongnu.org/ext2-doc/ext2.html)¹. W zadaniach 1–5 rozważamy pierwszą korektę ext2 (ang. *revision 1*).

UWAGA! W trakcie prezentacji należy być gotowym do zdefiniowania pojęć oznaczonych **wytłuszczoną** czcionką.

Zadanie 1. Wyznacz rozmiar **bloku**, liczbę i-węzłów i bloków przechowywanych w **grupie bloków** (ang. *block group*), liczbę wpisów **tablicy deskryptorów grup bloków** (ang. *block group descriptor table*) na podstawie pół **superbloku** (ang. *superblock*). Wymień składowe należące do grupy bloków oraz podaj ich rozmiar w blokach. Które grupy bloków przechowują kopie zapasową superbloku i tablicy deskryptorów grup bloków?

Zadanie 2. Podstawowymi operacjami na systemie plików są: wyzeruj lub zapal bit w bitmapie i-węzłów albo bloków, zmodyfikuj i-węzeł albo **blok pośredni** (ang. *indirect block*) albo blok danych. Wymień listę podstawowych operacji niezbędnych do realizacji funkcji: dopisującej n bloków na koniec pliku; dodającej plik do katalogu, gdy ten ma zbyt mało miejsca na dodanie wpisu. Zakładamy, że poszczególne kroki funkcji są zawsze wdrażane sekwencyjnie **zapisami synchronicznymi**. Zadbaj o to by funkcje nie naruszyły **spójności systemu plików** w przypadku awarii zasilania. Dopuszczamy powstawanie wycieków pamięci.

Zadanie 3. Przy pomocy wywołania systemowego **rename(2)** można przenieść **atomowo** plik do katalogu znajdującego się w obrębie tego samego systemu plików. Czemu «rename» zakończy się błędem «EXDEV» kiedy próbujemy przenieść plik do innego systemu plików? Powtórz polecenia z zadania 2 dla funkcji przenoszącej plik między dwoma różnymi katalogami. Zakładamy, że w katalogu docelowym jest wystarczająco dużo miejsca na dodanie wpisu.

Zadanie 4. Przy pomocy wywołania systemowego **unlink(2)** można usunąć plik niebędący katalogiem. Powtórz polecenia z zadania 2 dla funkcji usuwającej plik zwykły z katalogu. Kiedy możliwe jest odwrócenie operacji usunięcia pliku tj. odkasowania (ang. *undelete*)? Zauważ, że usunięcie pliku nie odbiera procesom możliwości czytania jego zawartości, o ile go otworzyły przed wywołaniem **unlink(2)**. Kiedy w takim razie plik zostanie faktycznie usunięty z dysku?

Zadanie 5. Wyjaśnij co robi system plików ext2 przy tworzeniu **dowiązania twardego** (ang. *hard link*) i **symbolicznego** (ang. *symbolic link*). Gdzie jest przechowywana zawartość dowiązania symbolicznego? Jak za pomocą dowiązania symbolicznego stworzyć w systemie plików pętlę? Kiedy jądro systemu operacyjnego ją wykryje i zwróci błąd «ELOOP»? Czemu pętli nie da się zrobić z użyciem dowiązania twardego?

Zadanie 6. Czemu **fragmentacja systemu plików** jest szkodliwym zjawiskiem? Zreferuj artykuł [The new ext4 filesystem: current status and future plans](https://www.kernel.org/doc/ols/2007/ols2007v2-pages-21-34.pdf)². Opisz w jaki sposób **odroczone przydziały bloków** (ang. *delayed allocation*) [§3.2] zapobiega powstawaniu fragmentacji. Wytłumacz jak **zakresy** (ang. *extents*) [§2.2] pomagają w ograniczaniu rozmiaru metadanych przechowujących adresy bloków należących do danego pliku. Czy po defragmentacji systemu plików ext4 liczba wolnych bloków może wzrosnąć? Jak mógłby wyglądać najprostszy algorytm defragmentacji [§3.3]?

¹<https://www.nongnu.org/ext2-doc/ext2.html>

²<https://www.kernel.org/doc/ols/2007/ols2007v2-pages-21-34.pdf>

Zadanie 7. Przy użyciu programu `debugfs(8)` dla wybranej instancji systemu plików ext4 (np. **partycja** przechowująca główny system plików Twojej instalacji systemu Linux) pokaż:

- fragmentację systemu plików (`freefrag`) i informacje o grupach bloków (`stats`),
- zakresy bloków z których składa się wybrany duży plik (`extents`),
- że dowiązanie symboliczne może być przechowywane w i-węźle (`idump`),
- do jakiego pliku należy wybrany blok (`blocks`, `icheck`, `ncheck`),
- reprezentację liniową małego katalogu (`bdump`).

Ostrzeżenie! Narzędzie `debugfs` działa domyślnie w trybie tylko do odczytu, więc możesz go bezpiecznie używać na swoim komputerze. Trybu do odczytu i zapisu używasz na własną odpowiedzialność!

Zadanie 8. Przeczytaj krytykę kluczowej idei systemu UNIX, tj. [A Unix File Is Just a Big Bag of Bytes³](#). Na podstawie [Resource Fork⁴](#) wyjaśnij czym były dodatkowe zasoby pliku w historycznych systemach MacOS.

Jaką postać mają **rozszerzone atrybuty pliku** `xattr(7)`? Pobierz z internetu plik przy pomocy przeglądarki Chrome, a następnie wyświetl jego rozszerzone atrybuty przy pomocy polecenia `getfattr(1)`. Następnie policz sumę md5 wybranego pliku i przypisz ją do atrybutu «user.md5sum» poleceniem `setfattr(1)`, po czym sprawdź czy operacja się powiodła.

Zadanie 9 (bonus). Na podstawie §3 artykułu [A Directory Index for Ext2⁵](#) opisz strukturę danych HTree i operację wyszukiwania wpisu katalogu o zadanej nazwie. Następnie wyświetl reprezentację HTree dużego katalogu, np. `/var/lib/dpkg/info`, używając polecenia `htree` programu `debugfs(8)`.

³<http://www.catb.org/~esr/writings/taoup/html/ch20s03.html#id3015538>

⁴https://en.wikipedia.org/wiki/Resource_fork

⁵<https://www.kernel.org/doc/ols/2002/ols2002-pages-425-438.pdf>