**Przedmiot:** Systemy operacyjne - laboratorium

**Tematyka:** System plików

Zagadnienie systemu plików jest pierwotnie omawiane i przerabiane w ramach przedmiotu

Wstęp do administracji i bezpieczeństwa komputerowego

Poniżej znajdują się zadania będącę kontynuacją tego tematu.

Do zadań 1.1, 1.2 i 1.3: Utwórz trzy rodzaje plików o rozmiarze ok. 100 MB:

* Plik pusty:
  + nazwa: zero.dat
  + zawartość: puste wartości
  + podpowiedź: polecenie dd, źródło /dev/zero
* Plik z danymi losowymi:
  + nazwa: losowy.dat
  + zawartość: liczby losowe
  + podpowiedź: źródło /dev/urandom lub /dev/random
* Plik z powielonymi danymi losowymi:
  + zawartość: liczby losowe 10MB, powtórzone x10, nazwa: powielony-10MB.dat
  + zawartość: liczby losowe 1kB, powtórzenie x100000, nazwa: powielony-1kB.dat
  + podpowiedź: źródło /dev/urandom lub /dev/random
  + polecenie ‘cat’ w celu powielenia x10 lub x10^5 zawartości pliku

1.1. Badanie stopnia upakowania (kompresji)

Spakuj pliki dowolnym programem (gzip, zip, 7z, rar, etc.) i wyjaśnij nowe rozmiary plików.

Proszę przetestować różne opcje stopnia kompresji. Jakie znaczenie dla kompresji ma powtarzalność ciągów ?

1.2. Dowiązania

Do pliku losowy.dat stwórz dowiązanie miękkie i twarde (ln -s, ln), odpowiedzio: losowy-soft.dat, losowy-hard.dat.

Korzystając z poleceń: ls, du oraz stat wyjaśnij zajętość przestrzeni przez poszczególne dowiązania.

1.3. System plików

Na pliku zero.dat załóż system plików ext4 i zamontuj w systemie plików.

Podpowiedź: wykład “System plików”, slajdy 20-23.

Spraw, aby montowanie/odmontowywanie tego systemu plików możliwe było przez użytkownika bez uprawnień root (podpowiedź: plik /etc/fstab).

1.4. Przeniesione do laboratorium dot. bezpieczeństwa

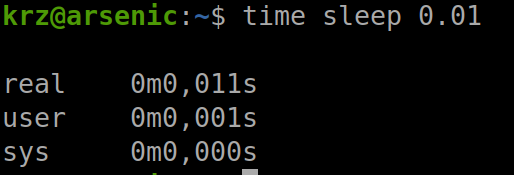
1.5. Napisz program w C/C++ kopiujący zawartość pliku jednego pliku do drugiego, korzystając z bufora o zadanym rozmiarze:

./kopiuj <rozmiar bufora> <plik.src> <plik.trg>

Proszę sprawdzić jaka jest zależność czasu kopiowania od rozmiaru bufora.

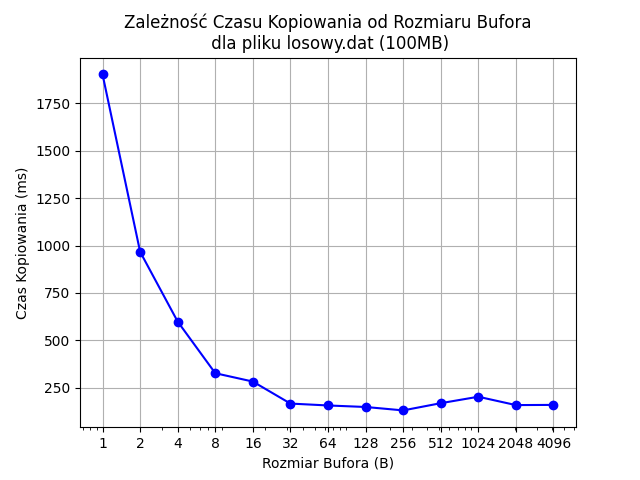
Wskazówka jak mierzyć czas:

* Funkcja odczytu czasu w C/C++
* Lub polecenie: time <nazwa programu> // ćwiczenie: time sleep 10





Autor: Jakub Młocek



Autor: Przemysław Szpyra

Proszę wynik w postaci wykresu oraz tabeli wyników zuploadować na UPEL.

**1.6. Napisz polecenie/skrypt wykrywający wszystkie przypadki zapętlonego linkowania miękkiego. (wskazówka: ln -s plik dowiązanie)**

./sprawdz-miekkie-linki.sh <katalog>

Podaj liczbę zapętleń (wypisz je) oraz długość poszczególnych zapętleń.

Przykład zapętlonego linkowania:

$ ln -s link\_1 link\_2

$ ln -s link\_2 link\_1

1.7. Napisz polecenie/skrypt wykrywający wszystkie przypadki hardlinkowania.

./sprawdz-twarde-linki.sh <katalog>

Podpowiedzi:

* Sprawdzenie, ile dowiązań ma plik (wpis w pliku katalogu): stat <nazwa pliku>
* Znalezienie plików o zadanej liczbie dowiązań: find -links <parametr>
* Pominąć rekordy, w których dana konfiguracja występuje jednokrotnie.

**1.8. Napisz polecenie/skrypt wypisujący statystykę plików w zadanym drzewie dla każdej konfiguracji ustawień dostępu (wyświetlić pliki o liczności >1).**

./statystyka-uprawnien.sh <katalog>

1 drwx------ - pominąć

1 -rw------- - pominąć

1 -rwsr-sr-x - pominąć

1 -rwx------ - pominąć

1 -rwxr-xr– - pominąć

1 -r-xr-xr-x - pominąć

2 -r--r--r--

4 drwxrwsr-x

10 -rwxr-sr-x

12 -rwsr-xr--

16 -rwsr-xr-x

16169 -rwxr-xr-x

24836 drwxr-xr-x

25683 lrwxrwxrwx

222177 -rw-r--r--

Ile będzie takich konfiguracji ? (pozycje od lewej)

* Pozycja 1: typ pliku: b, c, d, p, f (-), l, s = 7 możliwości
* Pozycja 2: r lub (-) = 2 możliwości
* Pozycja 3: w lub (-) = 2 możliwości
* Pozycja 4: x lub S lub (-) = 3 możliwości
* Pozycja 5: r lub (-) = 2 możliwości
* Pozycja 6: w lub (-) = 2 możliwości
* Pozycja 7: x lub S lub (-) = 3 możliwości
* Pozycja 8: r lub (-) = 2 możliwości
* Pozycja 9: w lub (-) = 2 możliwości
* Pozycja 7: x lub T lub t lub (-) = 4 możliwości

Liczba możliwości: 16128

Wypisać tylko te, dla których znaleziono pliki.

1.9.\* Porównanie efektywności systemu pliku, w zależności od wielkości bloku

1. Przygotuj 4 pliki o wielkości 100 MB każdy i nazwach: disc\_512, disc\_1024, disc\_2048, disc\_4096 (wskazówka: użyj urządzenia /dev/zero).
2. Załóż system plików na każdym z utworzonych plików poleceniem mkfs.ext4 ustawiając odpowiednio dla nazwy pliku wielkość bloku, tj.: 512, 1024, 2048, 4096.
3. Jaka jest maksymalna liczba plików, które można utworzyć na każdym z tych systemów?
4. Zamontuj utworzone pliki tak, jak urządzenia blokowe (wskazówka: skorzystaj z loop).
5. Wrzuć do każdego podmontowanego urządzenia:
   1. 1000 plików o rozmiarach (do wszystkich ten sam zestaw), kolejno: 512, 1024, 2048, 4096 bajtów. - który rozmiar bloku systemu plików najefektywniej przechowuje poszczególne rozmiary plików ?
   2. 1000 plików o losowym rozmiarze od 256 B do 8192 B - który rozmiar bloku systemu plików najefektywnien
6. Przetestuj 3 pośród systemów plików: mkfs.fat (mkfs.msdos, mkfs.vfat), mkfs.btrfs, mkfs.ext4, mkfs.hfs, mkfs.ntfs, mkfs.cramfs, mkfs.reiserfs, mkfs.xfs.