

Wstęp do systemu operacyjnego UNIX

Laboratorium 3:

Pliki i katalogi w systemie UNIX

W systemie operacyjnym Unix rozróżnia się następujące rodzaje plików:

1. pliki regularne
2. katalogi
3. pliki specjalne. Do tej kategorii zalicza się np:
 - pliki urządzeń zewnętrznych (katalog `/dev`),
 - pliki komunikacji międzyprocesowej (typu socket lub pipe).

Każdy rodzaj pliku jest traktowany jednakowo, co stanowi bardzo silną stronę tego systemu.

Plik regularny jest podstawową jednostką zarządzaną przez podsystem obsługi pamięci masowej. Jest on traktowany jako ciąg bajtów o odpowiedniej długości. Dzięki istnieniu katalogu pliki można zapisywać w uporządkowanej strukturze typu drzewa. Wyższą jednostką organizacyjną jest system plików. Pozwala on na efektywne administrowanie zbiorem plików i katalogów. Do przechowywania informacji wykorzystuje on pewną logiczną strukturę, zwaną wolumenem logicznym. Systemy plików montuje się następnie w ostateczną strukturę drzewa plików i katalogów w systemie. Wejście do takiej struktury jest możliwe jedynie przez katalog główny zwany korzeniem – *root* (`/`). W tym systemie plików nie występuje pojęcie dysku (napędu). Istnienie dysku jest całkowicie zakryte przed użytkownikiem.

0. Podłącz się do systemu jako użytkownik *root*. Sprawdź, czy w systemie został zdefiniowany użytkownik *test*. Jeśli nie to zdefiniuj go. Podłącz się do systemu jako użytkownik *test*.

1. W katalogu domowym użytkownika *test* załóż jednym wywołaniem komendy `mkdir katalog lab3`, a w nim podkatalogi `text1`, `text2` oraz `text3`. Odpowiednią opcję komendy `mkdir` znajdź w manualu. Komendą `ls` o opcję `-R` sprawdź, czy katalogi zostały utworzone poprawnie.

Komenda `dd` służy do kopiowania informacji z urządzenia na urządzenie (device to device). Faktycznie, pod pojęciem urządzenia należy rozumieć tutaj plik (specjalny). Komenda ta posiada szereg opcji, z których najważniejsze to:

- `if= file` – dane czytaj z pliku o ścieżce dostępu *file*.
- `of= file` – dane zapisz do pliku *file*.
- `bs= rozmiar` – odczytuj i zapisuj *rozmiar* bajtów na raz. Jest to rozmiar bloku kopiowanego.
- `count= blocks` – skopiuj dokładnie *blocks* bloków wejściowych o rozmiarze *rozmiar*.

Jeśli nie określimy jawnie parametrów `if` bądź `of` domyślnie zostaną przyjęte standardowe wejście i standardowe wyjście.

W katalogu `/dev` znajdują się pliki reprezentujące urządzenia w systemie. Na razie interesujące będą dla nas dwa. `/dev/null` jest to systemowy śmietnik o nieskończonej pojemności niestety bez możliwości odzyskania zapisanej tam informacji. Plik `/dev/urandom` jest generatorem bajtów o wartościach losowych.

2. Przejdź do katalogu `./lab3/text1`. Zapoznaj się z manuałem komendy `dd` zwracając szczególną uwagę na opcje `ibs` oraz `obs`. Przy pomocy komendy `dd` utwórz w bieżącym katalogu plik o nazwie `rand.bin`.

W systemach plików obsługiwanych w systemach UNIX z każdym plikiem związana jest struktura zwana *i-węzłem* (*i-node*) opisująca jego zawartość. Struktura ta zawiera informacje dotyczącą m.in. praw dostępu do pliku, właścicieli: indywidualnego i grupowego, dat utworzenia modyfikacji i dostępu, oraz tablicę opisującą sposób rozmieszczenia zawartości pliku w blokach danych systemu plików.

3. Korzystając z komendy `ls` określ numer i-węzła opisującego plik `rand.bin`. Z jaką opcją należy wywołać komendę `ls`?

Użycie komendy `ls -l` spowoduje wypisanie w postaci długiej informacji o plikach i katalogach znajdujących się w bieżącym katalogu:

```
1 drwxrwxr-x   9 bory    bory      4096 Jul 29 22:44 rystmp
2 -rw-rw-r--   1 bory    bory      804378 Apr 12:56 rystmp.tgz
3 -rw-rw-r--   1 bory    bory       125 Nov 16  2002 s
4 drwxr-x---   2 bory    bory      4096 Apr  8 11:23 saad
5 -rw-rw-r--   1 bory    bory     187523 Apr  8 11:43 saad1.ps
6 -rw-----   1 bory    bory     128063 Jan 30  2003 schaltplan.jpg
```

Pierwszy znaczek listingu dotyczy typu pliku. W przykładzie mamy regularny - i katalog d. Spotyka się jeszcze link l, socket s i pipe p. Typ pliku nie podlega zmianie.

Kolejnych 9 znaków dotyczy praw dostępu i należy je czytać w 3 grupach po 3. Kolejne trójki dotyczą właściciela pliku u (*ang. user*), grupowego właściciela pliku g (*ang. group*) oraz wszystkich pozostałych użytkowników w systemie o *ang. other*. Liter y w każdej trójce występują zawsze w tej samej kolejności -, r, w, x – jeśli dane prawo jest odpowiedniemu właścicielowi przydzielone lub - w odpowiednim miejscu, jeśli to prawo jest cofnięte. Prawa dostępu do pliku lub katalogu może zmieniać jego właściciel lub użytkownik root. Służy do tego komenda `chmod`. Jej składnia na pierwszy rzut oka wydaje się być skomplikowana. Jest jednak strukturalna co ułatwia jej zrozumienie. Komenda wymaga podania sposobu zmiany lub nowych praw dostępu oraz listy plików lub katalogów, których zmiana ma dotyczyć. Zmianę praw dostępu specyfikujemy symbolicznie, podając którego właściciela (grupy praw) zmiana ma dotyczyć (u, g, o lub a to wszystkich *ang. all*), następnie co z prawem robimy (+ dodajemy, - usuwamy, = usuwamy wszystkie inne i wskazane dodajemy) oraz którego prawa modyfikacja ma dotyczyć. Ustawienie praw dostępu realizujemy przy pomocy zapisu oktalnego, podając prawa dostępu, które mają zostać ustawione. Oto kilka przykładów:

```
1 [bory@thorin bory]$ ls -l prog
2 -rwxrwxr-x   1 bory    bory      13437 Aug  1 00:34 prog
3 [bory@thorin bory]$ chmod o-x prog
4 [bory@thorin bory]$ ls -l prog
5 -rwxrwxr--   1 bory    bory      13437 Aug  1 00:34 prog
6 [bory@thorin bory]$ chmod go-r prog
7 [bory@thorin bory]$ ls -l prog
8 -rwx-wx---   1 bory    bory      13437 Aug  1 00:34 prog
9 [bory@thorin bory]$ chmod 700 prog
10 [bory@thorin bory]$ ls -l prog
11 -rwx-----   1 bory    bory      13437 Aug  1 00:34 prog
12 [bory@thorin bory]$ chmod g=rx prog
13 [bory@thorin bory]$ ls -l prog
14 -rwxr-x---   1 bory    bory      13437 Aug  1 00:34 prog
```

4. Zapoznaj się ze składnią komendy `chmod` (manual). Zwróć szczególną uwagę na sposób specyfikacji praw dostępu z wykorzystaniem notacji liczbowej.

5. Przejdź do katalogu `/lab3`. Do katalogów `./text2` oraz `./text3` skopiuj z katalogu `./text1` plik `rand.bin`. Przejdź do katalogu `./text3`. Dla pliku `rand.bin` cofnij prawo zapisu dla wszystkich właścicieli. Spróbuj usunąć plik komendą `rm`. Czy się udało? Przejdź do katalogu `/lab3`.

6. Będąc w katalogu `/lab3` cofnij dla właściciela prawo zapisu do katalogu `text2`. Przejdź do katalogu `text2`. Spróbuj teraz usunąć plik `rand.bin`. Przejdź do katalogu `/lab3`. Cofnij prawo wykonywania dla właściciela katalogu `text2`. Czy dasz radę wejść do katalogu `text2`? A wylistować jego zawartość? Przywróć prawo wykonywania i cofnij odczytu dla właściciela katalogu `text2`. Jak teraz wyglądają możliwości listowania zawartości i przechodzenia do katalogu `text2`? Ustaw dla katalogu `text2` prawa 750.

Druga kolumna listingu komendą `ls` z opcją `-l` zawiera liczbę dowiązań do danego pliku lub katalogu. Pozycja ta obejmuje liczbę dowiązań „twardych”.

Zasadniczo w systemie plików istnieją dwa typy dowiązań:

1. Dowiązanie twarde (*ang. hard link*) jest inną nazwą istniejącego pliku. Dowiązanie i oryginalna nazwa są w pełni równoprawne. (Technicznie rzecz biorąc, posiadają ten sam i-węzeł, zaś i-węzeł zawiera całą informację o pliku - faktycznie nie jest błędem stwierdzenie, iż i-węzeł stanowi plik). W ten sposób plik może występować pod wieloma równoważnymi nazwami - wskazującymi na te same dane. Usunięcie jednej z tych nazw przez komendę `rm` nie powoduje jeszcze usunięcia pliku. Jest on usuwany dopiero z chwilą usunięcia ostatniej nazwy. We wszystkich istniejących implementacjach nie można tworzyć twardych dowiązań do katalogów, a dowiązania twarde nie mogą przekraczać granic systemów plików (ograniczenia te nie są jednak narzucone przez standard POSIX),
2. Dowiązania symboliczne (*ang. symbolic link*), z drugiej strony, są specjalnym typem plików (nieobsługiwanym przez niektóre jądra), w którym plik dowiązania faktycznie wskazuje na inny plik (przez nazwę). Dla większości operacji (otwarcie, odczyt, zapis i tak dalej) otrzymujących jako argument dowiązanie symboliczne jądro automatycznie „odwzorowuje” dowiązanie i działa na samym celu dowiązania. Przy niektórych (np. usuwanie) działa na samym dowiązaniu, nie zaś na celu przez nie wskazywanym. Liczba dowiązań symbolicznych nie jest zliczana w i-węźle czyli również nie jest wyświetlana przez komendę `ls`

Do tworzenia linków zarówno symbolicznych jak i „twardych” służy komenda `ln`. Jeśli w wywołaniu komendy podano tylko jeden argument – nazwę pliku, to komenda `ln` utworzy w bieżącym katalogu dowiązanie do niego mające taką samą nazwę (oczywiście nie można w ten sposób utworzyć dowiązania do pliku w bieżącym katalogu). Jeśli podano dwa argumenty typu plik, to pierwszy z nich zostanie utworzony jako dowiązanie do drugiego. Jeśli ostatni z argumentów jest katalogiem, to w tym katalogu zostaną utworzone dowiązania do plików podanych jako poprzednie argumenty. Podanie więcej niż dwu argumentów, z których ostatni nie jest katalogiem jest błędem. Domyślnie `ln` tworzy dowiązania twarde (*hard links*). Tworzenie linków symbolicznych wymaga użycia opcji `-s`.

7. Usuń plik `rand.bin` z katalogu `/lab3/text2`. Będąc w katalogu `/lab3` sprawdź przy pomocy komendy `ls`, czy plik `rand.bin` znajduje się tylko w katalogu `./text1`. Przejdź do katalogu `./text3`. utwórz w nim dowiązanie symboliczne do pliku `../text1/rand.bin`. Przejdź do katalogu `./text2`. Utwórz w nim dowiązanie (twarde) do pliku `../text1/rand.bin`. Przejdź do katalogu `/lab3`. Sprawdź komendą `ls` zawartości wszystkich podkatalogów (wypisz również numery i-węzłów).

Wynik listingu powinien być zgodny z przedstawionym poniżej. Zwróć uwagę, że zmieniła się liczba dowiązań do pliku `rand.bin` z katalogu `text1`. Rozmiar pliku w katalogu `text3` (dowiązanie symboliczne) wynosi dokładnie tyle, ile znajduje się znaków w ścieżce dostępu do pliku wskazywanego.

```
1  .:
2  total 12
3  drwxrwxr-x   2 bory   bory       4096 Aug  5 01:43 text1
4  drwxrwxr-x   2 bory   bory       4096 Aug  5 01:44 text2
5  drwxrwxr-x   2 bory   bory       4096 Aug  5 01:45 text3
6
7  ./text1:
8  total 56
9  -rw-rw-r--   2 bory   bory       50236 Aug  5 01:44 rand.bin
10
11 ./text2:
12 total 56
13 -rw-rw-r--   2 bory   bory       50236 Aug  5 01:44 rand.bin
14
15 ./text3:
16 total 0
17 lrwxrwxrwx   1 bory   bory       17 Aug  5 01:45 rand.bin -> ../text1/rand.bin
```

8. Przejdź do katalogu `/lab3/text1`. Usuń plik `rand.bin`. Przejdź do katalogu `/lab3` i ponownie wylistuj zawartość wszystkich podkatalogów. Co się zmieniło? Usuń dowiązanie symboliczne z katalogu `text3`.

Trzecią i czwartą kolumnę listingu komendy `ls` z opcją `-l` zajmują kolejno nazwa użytkownika będącego właścicielem pliku oraz nazwa grupowego właściciela pliku (najczęściej jest to grupa podstawowa właściciela

pliku). Zmiany jednego jak i drugiego może dokonać użytkownik root przy pomocy komendy `chown`. Grupę może zmienić również użytkownik pod warunkiem, że zmiana dotyczy grup których jest członkiem. Składnia tej komendy jest bardzo prosta. Wymaga ona podania nazwy nowego właściciela indywidualnego i/lub grupowego oraz listy plików, których zmiana ma dotyczyć. Właścicieli specyfikujemy według następującego schematu:

- `nowy_właściciel`

Jeśli podano tylko właściciela (nazwę użytkownika lub jego identyfikator UID), to ten użytkownik staje się właścicielem pliku, a grupowy właściciel pliku pozostaje niezmienny.

- `nowy_właściciel.grupa`

- `nowy_właściciel:grupa`

Jeśli nazwa użytkownika jest zakończona dwukropkiem lub kropką, bezpośrednio po której następuje nazwa grupy (lub numer identyfikacyjny grupy GID), to zmieniana jest także grupa pliku.

- `nowy_właściciel.`

- `nowy_właściciel:`

Jeśli natomiast wpisujemy dwukropek lub kropkę, lecz nie napiszemy nazwy grupy, użytkownik staje się właścicielem plików, a grupa jest ustawiana na grupę główną użytkownika.

- `.grupa`

- `:grupa`

Jeśli podano dwukropek lub kropkę i grupę, lecz pominięto nazwę użytkownika, zmieniana jest tylko grupa plików; w tym wypadku `chown` dokonuje tych samych operacji co komenda `chgrp`.

9. Przejdź do katalogu `/lab3/text2`. Spróbuj zmienić właściciela pliku `rand.bin` na użytkownika root. Czy się udało?

10. Podłącz się do systemu jako użytkownik root. W katalogu `/tmp` utwórz plik o nazwie `ls.txt` zawierający manual dla komendy `ls` (przekierowanie z komendy `man ls`). Zmień właściciela pliku na użytkownika test oraz grupowego właściciela na grupę podstawową tego użytkownika. Odłącz się od systemu i podłącz ponownie jako użytkownik test. Z katalogu `/tmp` przenieś do katalogu osobistego plik `ls.txt`. Czy przeniesienie było możliwe?

W celu zapewnienia większej elastyczności i wygody w administrowaniu oraz podniesienia jego bezpieczeństwa wprowadzono dodatkowe prawa własności plików i katalogów:

- prawo SUID, obowiązuje dla pliku regularnego i oznacza, że proces z niego uruchomiony będzie wykonywał się z efektywnym identyfikatorem właściciela pliku, a nie użytkownika, który go uruchomił. Symbolicznie prawo to jest oznaczane na prawie wykonania (x) dla właściciela literą `s` lub `S` jeśli prawo wykonania jest odpowiednio ustawione lub nie. Numerycznie prawo to jest oznaczane jako 4000.

Przykład:

```
1 -rws--x--x  1 root    root      12072 Apr  2  2002 /usr/bin/chfn
```

- prawo SGID może zostać ustawione dla pliku i katalogu. Dla pliku oznacza, że proces z niego uruchomiony będzie wykonywał się z efektywnym identyfikatorem grupowego właściciela pliku, z którego proces został uruchomiony, a nie podstawowej grupy użytkownika, który proces uruchomił.

Prawo to ustawione dla katalogu oznacza, że prawa dostępu do plików i katalogów w nim tworzonych będą dziedziczone od niego, a nie zależne od innych ustawień w systemie.

Symbolicznie prawo SGID jest oznaczane na prawie wykonania dla właściciela grupowego literą `s` lub `S` jeśli prawo wykonania dla właściciela grupowego jest odpowiednio ustawione lub nie. Numerycznie oznaczane jest jako 2000.

Przykład:

```
1 -r-xr-sr-x  1 root    tty      6920 Mar 14  2002 /usr/bin/wall
2 -rwxr-sr-x  1 root    tty      8584 Apr  2  2002 /usr/bin/write
```

- *sticky bit* lub SVTX. Prawo to obowiązuje jedynie dla katalogu i oznacza, że utworzone w nim pliki i katalogi może jedynie usunąć właściciel i użytkownik root.

Symbolicznie prawo to jest oznaczane na prawie wykonania dla pozostałych użytkowników systemu literą **t** lub **T** jeśli prawo wykonania jest odpowiednio ustawione lub nie. Numerycznie jest ono oznaczane przez 1000.

Klasyczny przykład:

```
1 drwxrwxrwt  9 root    root      4096 Aug  5 06:18 /tmp
```

11. Znajdź w manualu do komendy `chmod` sposób na ustawianie praw SUID, SGID oraz SVTX. Przejdź do katalogu `/tmp`. Utwórz w nim katalog o nazwie `sgid_test` i pełnych prawach dostępu. Przejdź do katalogu `sgid_test` i utwórz w nim komendą `touch` plik o nazwie `regularny` i katalog o nazwie `podkatalog`. Jakie są prawa dostępu do utworzonych plików? Gdzie stosuje się tego typu ustawienia?

Administrując systemem szczególnej kontroli należy poddawać pliki z ustawionymi prawami SUID i SGID. Zwróćmy uwagę, że w większości przypadków właścicielem pliku jest użytkownik root lub któraś z grup systemowych. Stwarza to duże możliwości dla procesu utworzonego z takiego pliku. Faktycznie otwiera to możliwość nadużycia lub włamania. Jak zatem znajdować w systemie takie pliki? Służy do tego komenda `find`. Wymaga ona podania jako argumentu ścieżki dostępu do katalogu w systemie plików od którego rozpoczniemy przeszukiwanie oraz wyrażenia (schematu) według którego poszukujemy plików lub katalogów. Komenda wypisuje na standardowe wyjście ścieżkę dostępu do pliku lub katalogu dla którego wyrażenie jest prawdziwe. Dla znalezienia interesujących nas plików komenda `find` przyjmie postać:

```
1 find /-perm +6000 -type f -print > setuid.txt
2 find /-perm +6000 -type f -exec ls -l {} \; > setuid_long.txt
```

Dodatkowo możemy dla każdego ze znalezionych plików wykonać polecenie np. polecenie `ls` patrz druga linia przykładu powyżej.

Komenda ta utworzy w bieżącym katalogu plik tekstowy o nazwie `setuid.txt` zawierający bezwzględne ścieżki dostępu do plików regularnych znajdujących się w systemie plików od katalogu głównego (wszystkie pliki) o prawach dostępu z ustawionymi SUID i SGID (4000+2000=6000). Drugie wywołanie komendy `find` listuje je w postaci długiej. W praktyce należy przyrzeć się tej liście i usunąć te pliki, które nie są w systemie wykorzystywane. Drugim, praktykowanym rozwiązaniem jest usunięcie bitów SUID/SGID z plików lub katalogów, co do których jesteśmy pewni że będą wykorzystywane jedynie przez użytkowników będących administratorami.

12. Znajdź w katalogu `/usr` oraz jego wszystkich podkatalogach pliki z ustawionymi prawami SUID i SGID i zapisz ich długi listing do pliku `setuid_long.txt` w katalogu bieżącym. Które pliki z tej listy można z systemu usunąć?

Jednym z zasobów, które w systemie komputerowym wyczerpują się najszybciej jest przestrzeń dyskowa. Z praktyki wynika, że przepełnienie systemu plików ma zawsze miejsce w najmniej pożądanym momencie. Istnieje kilka typowych systemów plików, które dobrze jest systematycznie sprawdzać pod kątem zajętości lub wręcz założyć system ograniczeń (quota). Jak jednak „personalnie” znaleźć tych użytkowników, którzy w systemie plików `home` zajmują najwięcej miejsca lub katalogi o największej zajętości w innych systemach plików? Służy do tego komenda `du` (*ang. disk usage*). Użyjmy jej w powiązaniu z komendą `sort` znajdując się w katalogu `/home` w następującej postaci:

```
1 [root@thorin home]# du -cks * | sort -nr | head
2 1083096 razem
3 583944 bory
4 499096 winda.img
5 40 test
6 16 lost+found
```

13. Podłącz się do systemu jako użytkownik root. Przejdź do katalogu `/home`. Sprawdź ile miejsca zajmują znajdujące się tam pliki i katalogi. Wypisz je w kolejności od najmniejszej do największej. Jakich opcji komendy `sort` należy użyć? Co oznaczają opcje `c`, `k` i `s` komendy `du`?

14. Skopiuj plik ze strony <http://messy.icsr.agh.edu.pl/sysopy/lab3.img>. Plik ten jest obrazem dyskietki, skopuj go na dyskietkę za pomocą polecenia `dd` a następnie sprawdź co zawiera dyskietka. Ponieważ dostarczony obraz jest obrazem systemu plików FAT zawartość dyskietki można wyświetlać za pomocą polecenia `mdir`