

mgr Maciej Wróbel

Obsługa pamięci masowej, systemy plikowe, operacje na plikach i katalogach. Struktura systemu plików w Linux.

4 październik 2010

1. Wprowadzenie – system plików Linux

1.1. Hierarchiczna struktura katalogów

Podobnie jak system Microsoft Windows, system Linux posiada system plików, który organizuje pliki w katalogach o hierarchicznej strukturze¹. W systemach opartych o UNIX istnieje jednak jedyny katalog główny, który jest nadrzędny wobec wszystkich katalogów – katalog / (nazywany *root*). Wszystkie katalogi są jego podkatalogami. Istotną różnicą w stosunku do systemów operacyjnych firmy Microsoft jest to, że nie ma odrębnych katalogów głównych dla poszczególnych partycji. Wynika to między innymi ze specyficznego dla UNIX (i Linux) podejścia do obsługi urządzeń wejścia/wyjścia, które jądro systemu operacyjnego udostępnia w systemie jako pliki. Dzięki takiemu podejściu możliwa jest obsługa urządzeń tak, jakby były (prawie) zwykłymi plikami. Poszczególne partycje *montowane* są jako katalogi gdzieś w hierarchicznej strukturze systemu plików.

Aby ustandaryzować strukturę systemu plików (co wygodne jest zarówno dla administratorów, jak i programistów) wprowadzono standard hierarchii systemu plików (FSSTND v2.3, 2004 rok). Określa ona m. in., że w katalogu / powinny się znaleźć katalogi:

- /bin — zawiera polecenia dostępne dla użytkowników,
- /boot — zawiera pliki niezbędne do uruchomienia systemu,
- /dev — jest to specjalny system plików dla urządzeń i plików specjalnych,
- /etc — powinien zawierać wszystkie pliki konfiguracyjne dla systemu,
- /home — zawiera katalogi i pliki użytkowników,
- /initrd — jest to katalog zawierający RAMdysk, wykorzystywany przy uruchamianiu systemu. Jest odmontowywany po pomyślnym uruchomieniu systemu,
- /lib — katalog ten zawiera moduły jądra i dzielone biblioteki,

¹ Choć są także próby tworzenia systemów plików opartych o bazy danych.

/lost+found — tutaj umieszczane są pliki odzyskane przez program *fsck*, które zostały utracone na skutek niewłaściwego zamknięcia systemu,

/media — katalog w którym umieszczane są przenośne nośniki danych (ang. *removable media*),

/mnt — standardowy katalog docelowy, w którym montowane są systemy plików,

/opt — tutaj umieszczane są aplikacje, które nie są elementem standardowej instalacji systemu,

/proc — specjalny wirtualny system plików, w którym umieszczane są informacje o działaniu systemu,

/root — jest to katalog domowy użytkownika root (czyli administratora),

/sbin — tutaj umieszczane są programy administracyjne i dotyczące utrzymania systemu,

/usr — tutaj znajduje się najwięcej danych (nie licząc plików użytkowników) – jest to wspólny katalog, w którym znajdują się aplikacje, dokumentacja biblioteki itd. Nazwa pochodzi od ang. *user usable programs and data*.

/var — ang *variable data*, czyli zmienne dane, takie jak logi systemowe, pliki poczty, pliki tymczasowe itd.,

/srv — jest to miejsce na dane udostępniane przez system, czyli np ftp, cvs czy rsync,

/tmp — miejsce na pliki tymczasowe, zwykle kasowane przy uruchamianiu i wyłączeniu systemu (ale niekoniecznie!).

Opis struktury katalogów dostępny jest na stronie *hier* podręcznika man (polecenie *man hier*).

Systemy UNIX udostępniają komendę **cd**, która służy do poruszania się po systemie plików. Ścieżki można podawać jako bezwzględne (zaczynające się od /). Sprawdzana jest wtedy tylko podana ścieżka. Jeżeli ścieżka nie zaczyna się od /, to powłoka systemowa sprawdza, czy ścieżka jest wskazaniem do podkatalogów katalogu bieżącego, a jeżeli nie, to przeszukuje ścieżki w zmiennej środowiskowej \$CDPATH². **cd** wywołane bez argumentów zmienia katalog na katalog domowy użytkownika, a z przełącznikiem - na poprzednio używany katalog. Polecenie **pwd** wyświetla ścieżkę bieżącego katalogu.

Katalogi możemy tworzyć przy pomocy polecenia **mkdir**, a usuwać poleceniem **rmdir**. Zwykle system zaprotestuje przy próbie usunięcia katalogu niepustego. Wtedy można wymusić jego usunięcie np. poleceniem **rm -rf nazwakatalogu**. Polecenia **mkdir** i **rmdir** z przełącznikiem -p pozwalają jednym poleceniem utworzyć całą strukturę katalogów (np. *mkdir -p a1/a2/a3/a4* utworzy strukturę katalogów *a1/a2/a3/a4*).

Zawartość katalogu sprawdzić można poleceniem **ls**. Wywołane bez żadnych przełączników wyświetli zawartość bieżącego katalogu. Można także podać katalog, którego zawartość chcemy wyświetlić. Przełącznik -R pozwala rekursywnie przeglądać podkatalogi (co jednak może być czasochłonne). Przełącznik -d pozwala wyświetlać tylko katalogi.

W każdym katalogu w systemie Linux istnieją dwa ważne dowiązania do katalogów: *./* i *../*. Pierwszy z nich jest odniesieniem do katalogu bieżącego, a drugi do katalogu nad-

² Więcej o zmiennych środowiskowych na laboratorium dotyczącym programowania w powłoce systemowej.

rzędnego. Dowiązanie do katalogu nadrzędnego pozwala korzystać ze ścieżek względnych nie tylko podrzędnych katalogowi bieżącemu. Dowiązanie do katalogu bieżącego ma istotne zastosowanie dotyczące bezpieczeństwa systemu. W przeciwieństwie do innego popularnego systemu operacyjnego, system Linux *nie przeszukuje* bieżącego katalogu w poszukiwaniu aplikacji³ (przemyślcie, czym grozi przeszukiwanie katalogu bieżącego). Jeżeli więc chcemy uruchomić program, znajdujący się w katalogu bieżącym, to musimy jego nazwę poprzedzić ścieżką do niego. Istnienie dowiązania `./` pozwala wywołać program z użyciem składni `./nazwaprogramu`.

Zadania

1. Uruchom terminal. Sprawdź bieżący katalog.
 2. Wyświetl zawartość katalogu bieżącego, katalogu `/etc`, `/boot`.
 3. Wypróbuj zmienianie katalogu z wykorzystaniem ścieżek względnych i bezwzględnych.
 4. W katalogu domowym utwórz strukturę katalogów `dir1/dir2/dir3`.
 5. Usuń stworzoną strukturę katalogów.
 6. Zapoznaj się z pomocą polecenia `ls`. Wypróbuj działanie przełączników `-l`, `-1`, `-R`.
-

1.2. Obsługa plików zwykłych

Pliki w systemie Linux są szczególne – mówi się, że w systemach UNIX/Linux ang. *everything is a file*. Pliki w Linux są trojakiemu rodzaju:

1. pliki zwykłe,
2. katalogi,
3. pliki specjalne.

Rodzaj pliku (i różne informacje o nim) można sprawdzić poleceniem **file**. Pliki zwykłe możemy utworzyć w powłoce systemowej np. poleceniem **touch**. Kopiujemy je poleceniem **cp** **zrodlo cel**, a usuwamy poleceniem **rm**. Przenieść pliki (lub zmienić ich nazwę) możemy przy pomocy polecenia **mv**.

Pliki możemy znaleźć wykorzystując polecenia **locate**, **find** i **whereis**:

1. **locate** przeszukuje periodycznie aktualizowaną bazę plików i zwraca listę pełnych ścieżek do plików pasujących do podanego wzorca,
2. **whereis** zwraca pełne ścieżki plików binarnych i pomocy systemowe pasujących do podanego wzorca,
3. polecenie **find** pozwala na zaawansowane wyszukiwanie plików. Jego podstawowe opcje to:
 - name**/**-iname** wyszukuje pliki pasujące do wzorca z uwzględnieniem/bez uwzględnienia wielkości liter,
 - size** wyszukuje pliki o podanych kryteriach dotyczących rozmiaru pliku,

³ więcej o ścieżkach przeszukiwanych podczas próby wykonania polecenia na zajęciach dotyczących obsługi powłoki Bash

-atime/-mtime/-ctime wyszukuje pliki o określonych własnościach dotyczących czasu dostępu/modyfikacji/utworzenia/

Polecenie **find** pozwala także na wykonywanie poleceń na wyszukanych plikach. Zainteresowanych odsyłam do stron podręcznika find (man find).

Wszystkie współcześnie używane systemy plików Linux umożliwiają korzystanie z dowiązań do plików. Dowiązania te dzielimy na dwa rodzaje:

1. dowiązania twarde,
2. dowiązania symboliczne.

Dowiązania twarde są bezpośrednim wskaźnikiem na węzeł pliku (ang. *inode*) w tablicy alokacji plików. Plik jest usuwany z systemu, gdy nie ma do niego żadnych dowiązań twarde. Dowiązania twarde mogą znajdować się tylko na tym samym systemie plików i nie mogą dotyczyć katalogów. Dowiązanie twarde tworzymy poleceniem:

```
ln nazwa_zrodla nazwa_celu
```

Jeżeli po utworzeniu dowiązania twardego usuniemy plik źródłowy, plik docelowy dalej będzie osiągalny.

Dowiązania symboliczne wskazują na pełną ścieżkę do pliku. Dzięki temu mogą znajdować się na innym systemie plików, mogą także wskazywać katalog. Tworzymy je poprzez polecenie:

```
ln -s nazwa_zrodla nazwa_celu
```

Dowiązania symboliczne stają się bezużyteczne po usunięciu pliku źródłowego.

Zadania

1. Stwórz plik abc.txt.
 2. Sprawdź rodzaj pliku abc.txt.
 3. Zmień jego nazwę na cba.txt.
 4. Skopiuj plik cba.txt do pliku cca.txt.
 5. Skopiuj plik cca.txt do pliku uprawnienia.txt.
 6. Przy pomocy polecenia locate znajdź pliki, które zawierają w nazwie łańcuch linux
 7. Przy pomocy polecenia find znajdź w katalogu /proc pliki. zawierające w nazwie łańcuch acpi (niezależnie od wielkości znaków).
 8. Znajdź w katalogu domowym puste pliki (o rozmiarze 0).
 9. Przy pomocy whereis znajdź polecenie locate.
 10. Utwórz link symboliczny o nazwie link.txt do pliku cca.txt
 11. Utwórz link twarde o nazwie twardy.txt do pliku cca.txt
 12. Usuń plik cca.txt. Co stało się z plikami twardy.txt i link.txt?
-

1.3. Wzorce uogólniające

Polecenia dotyczące plików i katalogów można wywoływać przez podanie wprost pełnej nazwy pliku/katalogu. W przypadku operowania na dużej liczbie plików byłoby to niewy-

godne. Podobnie jak w konkurencyjnym systemie istnieją więc wzorce uogólniające. Wyglądają one następująco:

***** — zastępuje dowolny ciąg znaków (w tym pusty)

? — zastępuje pojedynczy znak

zakres — zastępuje jeden znak z zakresu znaków

^ zakres — zastępuje jeden znak z dopełnienia zakresu znaków

Przykładowo:

1. `ls *aa*`
2. `rm ?.tmp`
3. `rm [1-3]*[xz]`
4. `ls [^a]*`

Polecenie 1 wyświetli wszystkie pliki zawierające wyrażenie aa. Polecenie 2 usunie wszystkie pliki zawierające jeden znak przed kropką i rozszerzenie tmp. Polecenie 3 usunie wszystkie pliki zaczynające się na 1, 2 lub 3 i kończące się na x lub z. Polecenie 4 wyświetli wszystkie pliki o nazwach nie zaczynających się na literę a.

Zadania

1. Wyświetl wszystkie pliki w katalogu /etc zaczynające się na literę a
 2. Wyświetl wszystkie pliki w katalogu /etc zawierające w nazwie łańcuch passwd
 3. Sprawdź, jakie pliki w katalogu domowym kończą się łańcuchem txt. Usuń pliki, które utworzyłeś w poprzednich zadaniach oprócz pliku uprawnienia.txt
-

2. Uprawnienia do plików i katalogów

Mechanizmy bezpieczeństwa w systemie UNIX pierwotnie oparte były o ograniczenia dostępu do plików. Uprawnienia użytkownika wynikały z regulacji dotyczących dostępu do plików. Administrator miał dostęp do wszystkich plików, zwykły użytkownik – do własnych plików oraz tych, które udostępnił mu administrator itd. Taki mechanizm zabezpieczeń jest stosunkowo słaby, gdyż zwykły użytkownik po otrzymaniu uprawnień do danego pliku nie był już niczym ograniczony w wykorzystaniu tych uprawnień. Chociaż model zabezpieczeń został rozbudowany, dalej uprawnienia dotyczące dostępu do plików są krytyczne w systemach Linux.

Ponieważ każdy plik ma tylko jednego właściciela, aby umożliwić szerszą kontrolę nad dostępem do plików plik posiada także przypisaną grupę. Użytkownicy zaś mogą należeć do jednej lub więcej grup. To, do jakiej grupy należy użytkownik sprawdzić można np. poleceniem **id**.

Ze względu na rolę, jaką pełniły prawa dostępu do plików, mechanizm ich obsługi w systemie Linux jest dosyć rozbudowany. Każdy plik ma określone prawa dotyczące jego odczytu (r), zapisu (w) oraz wykonania (x). Prawa te określone są niezależnie dla **właściciela** pliku, **grupy** do której należy plik oraz dla **pozostałych** użytkowników. Prawa dostępu do pliku

wyświetlane są jako ciąg 10 znaków. Pierwszy z nich określa rodzaj pliku (np *d* dla katalogu, - dla zwykłego pliku, *l* dla linków symbolicznych). Pozostałe 9 odpowiadają kolejno prawom odczytu, zapisu i wykonania dla kolejnych typów użytkowników. Przykładowo:

```
drwxr-xr-x jas przecietny abcd
```

oznacza, że *abcd* jest katalogiem (*d*), o prawach odczytu, zapisu i wykonania dla właściciela (użytkownika *jas*), oraz odczytu i wykonania dla użytkowników należących do grupy *przecietny* oraz odczytu i wykonania dla pozostałych użytkowników.

Ostatnie 9 znaków można także przedstawić przy pomocy 0 i 1. Tworzą one trzy 3-bitowe liczby. Ich dziesiętna reprezentacja, przedstawiona jako 3 cyfry jest poprawnym oznaczeniem uprawnień pliku. Przykładowo 765 binarnie to 111 110 101 i oznacza, że właściciel pliku ma wszystkie uprawnienia, członkowie grupy mają prawa odczytu i zapisu, a wszyscy użytkownicy odczytu i wykonania.

Uprawnienia plików i katalogów ustawia się przy pomocy narzędzia **chmod**, o składni:

```
chmod uprawnienia nazwa_pliku
```

gdzie uprawnienia mają postać:

użytkownicy Λ prawa

gdzie użytkowników określamy jako *u,g,o* (user, group, others) lub *a* (all), Λ to operator + (dodaje wskazane uprawnienia), - (usuwa wskazane uprawnienia) lub = (dodaje wskazane uprawnienia, a usuwa niewskazane). Prawa mają postać łańcucha znaków *r, w, i x*. Przykładowo

```
chown ug+rw abc.txt
```

dodaje uprawnienia odczytu i zapisu właścicielowi i grupie, nie zmieniając praw wykonania, zaś:

```
chown a=r abc.txt
```

powoduje, że wszyscy użytkownicy mają wyłącznie prawo odczytu pliku *abc.txt*.

Zadania

1. Nadaj plikowi *uprawnienia.txt* uprawnienia odczytu i zapisu dla wszystkich użytkowników.
 2. Korzystając z numerycznej notacji usuń uprawnienia zapisu i wykonania wszystkim użytkownikom.
-

2.1. Systemy plików Linux

Podczas instalacji (i eksploatacji) systemu Linux mamy stosunkowo dużą dowolność w wyborze systemu plików, który będzie porządkował nasze dane. Są to przede wszystkim

systemy ext2,ext3,ext4, ReiserFS i btrfs (dopiero wchodzący do użytku). Ponadto system Linux obecnie obsługuje partycje FAT, FAT32 i NTFS.

Natywne systemy plików Linux rozróżniają duże i małe litery oraz pozwalają korzystać z długich nazw plików. Umożliwiają stosowaną w Linux regulację uprawnień dostępu do plików. Wszystkie nowsze systemy plików w Linux są systemami z księgowaniem.

3. Sprawozdanie

Sprawozdanie powinno zawierać:

1. Opis 10 wybranych poleceń dotyczących obsługi plików lub katalogów lub zarządzania uprawnieniami.
2. Wyjaśnienie celowości wprowadzania standardowej struktury katalogów.
3. Opis dwóch wybranych systemów plików.
4. Omówienie zagrożeń wynikających z zabezpieczania systemu w oparciu o uprawnienia dostępu do plików **lub** omówienie zagrożeń wynikających z domyślnego poszukiwania programu do wykonania w ścieżce bieżącej.
5. Pięć przykładów użycia wzorców uogólniających w Linux.

W wersji elektronicznej instrukcji podane niżej linki są hiperłączami.

Literatura

- [1] Materiały MiM UW (j. polski) na http://wazniak.mimuw.edu.pl/index.php?title=Systemy_operacyjne dotyczące użytkowania systemu uniksopodobnego.
- [2] Każdy podstawowy podręcznik systemu Linux.
- [3] *Linux: administracja*. M. Carling, Stephen Degler, James Dennis. Wrocław : Wydaw. Robomatic, 2000.
- [4] Polskojęzyczny opis wyszukiwania plików na http://ultra.ap.krakow.pl/~bar/FIND/find_toc.html.
- [5] Opis struktury plików Linux (j. angielski) dla projektu TLDP, dostępny na http://tldp.org/LDP/intro-linux/html/sect_03_01.html.
- [6] Materiały (w j. angielskim) dotyczące certyfikatu LPI udostępniane przez IBM na <http://www.ibm.com/developerworks/linux/library/l-lpic1-v3-map/>
- [7] Materiały (także w j. polskim) dotyczące obsługi Linux udostępniane przez społeczność Gentoo na <http://www.gentoo.org/doc/pl/articles/>.
- [8] Materiały na Oopweb dotyczące administracji Linux (w j. angielskim): <http://oopweb.com/OS/Documents/SAG/VolumeFrames.html>.