

Systemy operacyjne

Laboratorium 5

Archiwa, kompresja danych. Komunikacja z użytkownikami.

Powtórzenie:

cd /ścieżka do nowego katalogu – zmiana katalogu

cd .. – zmiana katalogu (piętro wyżej)

cd – przeniesienie do katalogu domowego

ls – zwraca zawartość katalogu roboczego

ls -l – zwraca zawartość katalogu z dodatkowymi informacjami

ls -a – zwraca zawartość katalogu razem z plikami ukrytymi

man – manual

date – wyświetla datę i godzinę

cp – kopiuje pliki i katalogi

mv – przenosi pliki i katalogi lub zmienia ich nazwy

mkdir – tworzy katalogi

rm – usuwa pliki i katalogi

chmod – zmienia ustawienia dostępu do pliku/katalogu

chown – zmienia właściciela pliku

chgrp – zmienia grupę do jakiej należy plik

umask – nadawanie domyślnych ustawień praw dostępu dla nowych plików i katalogów

df – zwraca informacje na temat zajętego i dostępnego miejsca

uname – zwraca podstawowe informacje o systemie

hostname – zwraca nazwę hosta oraz

free – zwraca informacje na temat zajętego i wolnego miejsca w pamięci

lslogins – wypisuje informacje na temat wszystkich użytkowników systemu

grep – szukanie plików według ich zawartości

find – szukanie plików według atrybutów

1. Archiwizacja

Archiwizacja to proces łączenia wielu plików i katalogów w jeden plik (archiwum). Najpopularniejszym narzędziem archiwizującym w Linux jest tar (tape archive).

Najczęściej używane parametry polecenia tar:

c (create) – tworzenie archiwum

x (extract) – rozpakowanie archiwum

v (verbose) – wyświetla szczegółowe informacje

f (file) – wskazanie nazwy archiwum

t (list) – wypisanie zawartości archiwum

z (gzip) – kompresja/dekompresja gzip

j (bzip2) – kompresja/dekompresja bzip2

Przykłady użycia:

```
tar -cvf pliki.tar plik1.txt plik2.txt <- tworzenie archiwum
```

```
tar -xvf pliki.tar <- rozpakowanie archiwum
```

```
tar -tvf pliki.tar <- lista plików
```

2. Kompresja plików

Przez całą historię informatyki, nieustannie dążono do maksymalizacji wykorzystania dostępnej przestrzeni przez umieszczanie w niej jak największej ilości danych - niezależnie od tego, czy chodzi o pamięć komputerową, urządzenia do przechowywania danych, czy sieci o ograniczonej przepustowości. Wiele dzisiejszych urządzeń i usług, które traktujemy jako oczywiste, takie jak przenośne odtwarzacze muzyki, telewizja wysokiej rozdzielczości czy internet szerokopasmowy, owe swoje funkcjonowanie zaawansowanym metodami kompresji danych.

Algorytmy kompresji, czyli matematyczne metody stosowane do redukcji rozmiaru danych, dzielą się na dwie główne kategorie: bezstratną i stratną. Kompresja bezstratna umożliwia zachowanie całkowitej zawartości oryginalnych danych, co oznacza, że po dekompresji otrzymujemy plik identyczny z oryginałem. Z kolei kompresja stratna polega na eliminacji niektórych danych w celu osiągnięcia większego stopnia kompresji. Efektem tego jest plik, który po dekompresji jest tylko przybliżeniem oryginału, jak np. w przypadku plików JPEG dla obrazów i MP3 dla muzyki. W naszych dalszych rozważaniach skupimy się wyłącznie na kompresji bezstratnej, ponieważ w kontekście większości danych komputerowych, utrata jakichkolwiek informacji jest niedopuszczalna.

2.1 GZIP – kompresowanie i wyodrębnianie plików

Program gzip jest narzędziem przeznaczonym do kompresji pojedynczych plików lub ich zbiorów. Po uruchomieniu, zastępuje on każdy oryginalny plik jego wersją skompresowaną. Natomiast dedykowany mu program gunzip umożliwia dekompresję, przywracając skompresowane pliki do ich pierwotnej, nieskompresowanej formy.

2.2 BZIP2 – kompresja wyższego poziomu

Program bzip2 działa podobnie jak gzip, lecz korzysta z innego algorytmu kompresji. Pozwala osiągnąć wyższy poziom kompresji jednak potrzebuje do tego więcej czasu.

Przykłady użycia:

```
gzip plik.txt    <- tworzy plik.txt.gz
```

```
gzip -d plik.txt.gz <- dekompresja pliku
```

```
bzip2 plik.txt   <- tworzy plik.txt.bz2
```

```
bzip2 -d plik.txt.bz2 <- dekompresja pliku
```

Ćwiczenie 1

Z katalogu domowego użytkownika dbud skopiuj katalog Cwiczenie5 razem z zawartością do swojego katalogu domowego.

Ćwiczenie 2

1. Utwórz zarchiwizowany i skompresowany plik:
 - Stwórz katalog projekt i utwórz w nim kilka plików tekstowych oraz podkatalogów.
 - Używając jednego polecenia, utwórz archiwum projekt.tar.gz, które zawiera cały katalog projekt i jest skompresowane za pomocą gzip.
2. Dodaj plik do istniejącego archiwum:
 - Utwórz nowy plik dodatkowy.txt poza katalogiem projekt.
 - Dodaj dodatkowy.txt do istniejącego archiwum projekt.tar.gz
3. Usuń plik z archiwum:
 - Usuń plik plik1.txt z archiwum projekt.tar.gz.

Ćwiczenie 3

1. Skompresuj folder Cwiczenie5 ze swojego katalogu domowego używając gzip, bzip2 oraz xz
2. Porównaj rozmiary wynikowych plików oraz czas potrzebny na kompresję dla każdej z metod.
3. Zapisz która metoda była najszybsza a która zapewniła najlepszy stopień kompresji.

3. Komunikacja z innymi użytkownikami

Jeśli chcemy wysłać wiadomość do innego użytkownika systemu, musimy najpierw sprawdzić, kto jest zalogowany i na jakim terminalu. Służy do tego polecenie **who**.

Znając listę dostępnych użytkowników możemy wysłać wiadomości do dowolnego z nich. Używamy w tym celu polecenia **write**, które przyjmuje dwa parametry: nazwę użytkownika oraz terminal. Po poprawnym wywołaniu komendy przechodzimy do trybu pisania wiadomości. Aby zakończyć tryb pisania wiadomości musimy wcisnąć kombinację klawiszy **Ctrl+D**, która wywoła sygnał EOF (End of File).

Ćwiczenie 4

Otwórz drugie okno terminala i przećwicz wysyłanie wiadomości między terminalami.

Oprócz wysyłania wiadomości do konkretnego użytkownika istnieje również możliwość wysyłania wiadomości do wszystkich użytkowników na raz. W tym celu używamy polecenie **wall**. Działa podobnie jak **write**, z tą różnicą że nie przyjmuje adresata jako argumentu.

Ćwiczenie 5

Sprawdź działanie komendy wall.

4. Komenda mail – wysyłanie i odbieranie wiadomości w systemie Linux

mail to prosta, tekstowa aplikacja pocztowa, dostępna w wielu dystrybucjach Linuxa. Pozwala na wysyłanie i odbieranie wiadomości lokalnie, pomiędzy użytkownikami systemu (bez udziału internetu, SMTP itp.).

Aby działała, musi być zainstalowany pakiet mailutils lub bsd-mailx, a system powinien mieć aktywną usługę lokalnej poczty (np. postfix lub sendmail).

Ćwiczenie 6

Spróbuj wysłać wiadomość mail do dbud. (komenda 'mail dbud').

Spróbuj wysłać wiadomość do kogoś innego oraz odebrać i odczytać mail.

Usuń wiadomości jeśli jakieś są w skrzynce.

5. Edytor tekstu NANO.

nano to **prosty, tekstowy edytor plików**, dostępny praktycznie we wszystkich dystrybucjach systemu Linux. Działa w terminalu i pozwala na szybkie tworzenie oraz edycję plików tekstowych – bez potrzeby znajomości skomplikowanych komend.

Jest to doskonałe narzędzie dla początkujących użytkowników Linuxa, którzy chcą nauczyć się pracy z plikami konfiguracyjnymi, skryptami czy dokumentami tekstowymi bez używania myszki i środowiska graficznego.

Ćwiczenie 7

Stwórz plik w katalogu domowym ze swoim krótkim CV za pomocą nano. Zapisz imię, nazwisko, krótkie bio (2-3 zdania). Zapisz plik pod nazwą cv.txt.

Ćwiczenie 8

Otwórz plik cv.txt a następnie dodaj nowy akapit (Umiejętności). Wypisz ich co najmniej 5. Zapisz i zamknij plik.

Ćwiczenie 9

Przećwicz wszystkie podstawowe komendy edytora nano.

ZADANIA (zapisz odpowiedzi oraz użyte komendy w notatniku)

1. *Jak wyłączyć możliwość otrzymywania wiadomości z wall'a?*
2. Jak zakończyć wpisywanie wiadomości w mail?
3. Jak sprawdzić, kto jest zalogowany do systemu?
4. Jak wysłać wiadomość z pliku jako treść?
5. Jak usunąć wiadomość ze skrzynki odbiorczej?

Nano:

6. Jak zapisuje się plik w nano?
7. Jak wyszukać konkretne słowo?
8. Jak przejść do linii numer 42?
9. Jak skopiować i wkleić linię tekstu?

