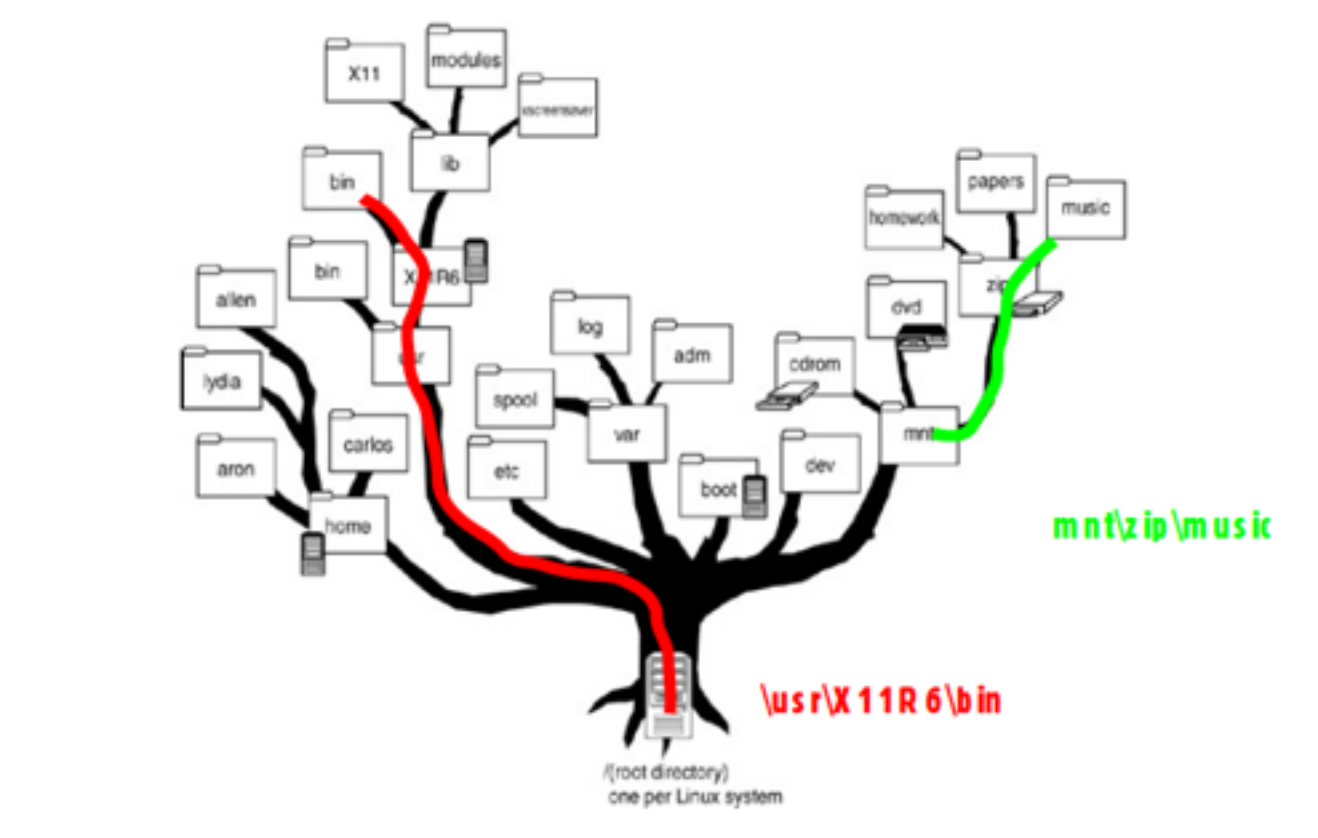
Linux

dr Magdalena Posiadała-Zezula (magdalena.Posiadala@fuw.edu.pl)
mgr Magdalena Grzeszczyk (mgrzeszczyk@student.uw.edu.pl)

www.fuw.edu.pl/~mposiada/pk15/

Struktura katalogów w postaci drzewa



Katalog domowy (1)

- Każdy użytkownik w systemie Linux ma przypisany swój katalog domowy.
- Jest to miejsce przeznaczone na wszystkie nasze dane, indywidualne pliki konfiguracyjne itp.
- Ponieważ często odwołujemy się do tego katalogu, dla wygody wprowadzono **oznaczenie ~/** jako katalog domowy.
- Przykład: cd ~mzpos – mój katalog domowy

Katalog domowy (2)

- Niech naszym katalogiem domowym będzie np.

  /home/studenci/Magda

- Niech w naszym katalogu domowym istnieje folder o nazwie przyklad.

- Do tego folderu możemy się odwołać albo:

  - /home/studenci/Magda/przyklad albo ~/przyklad

- W pewnym sensie znak ~ zastępuje ciąg znaków /home/studenci/Magda .

Katalogi i pliki- podstawowe pojecia (1)

WAŻNE!!!

- pwd – wyświetla aktualny katalog roboczy
- cd – pozwala zmienić katalog roboczy
- ls – komenda pozwalająca na wyświetlenie zawartości katalogu. Istotne jest przyswojenie kilku opcji, które można dla polecenia ls zastosować:
  - ls -a – pozwala na wyświetlenie plików „ukrytych” tzn. takich, ktorych nazwa zaczyna sie od kropki .
  - ls -l – wyświetla szczegółowe dane plików (o tym szerzej później)
  - ls -R – listuje katalogi „rekurencyjnie”

Katalogi i pliki- podstawowe pojecia (2)

- mkdir nazwa_kat- tworzenie katalogu o nazwie nazwa_kat
- rmdir nazwa_kat– usuwa pusty katalog o nazwie nazwa_kat

Polecenie man

- Polecenie man pozwala na przeglądanie dokumentacji wszystkich programów na naszym linuksie. Wystarczy wpisać man nazwa_polecenia np man ls
- Z man'a wychodzimy przyciskiem q.
- W man'ie szukamy przyciskiem /.
- Następny wynik wyszukiwania jest dostępny po kliknięciu n.

WAŻNE!!!

Polecenie cp- kopiowanie plikow i katalogów

- cp ścieżka1 scieżka2 - kopiuje plik z scieżki 1 do scieżki 2.
- Istotna opcja:
  - -r – rekurencyjnie (katalog wraz z zawartością)

Przykład

- Skopiuj prezentacje dotyczącą linuxa do swojego katalogu ~/_work_/linux
  Prezentacja znajduje się w

- /dmj/2000/mzpos/_work_/linux/linux_prac_komp_cw.pdf

Przykład- rozwiązanie

- Przykład rozwiązania:

1. wchodzimy do katalogu _work_ u siebie cd ~/_work_
2. tworzymy katalog linux: mkdir linux i do niego wchodzimy cd linux
3. kopiujemy prezentacje:
   - cp /dmj/2000/mzpos/_work_/linux/linux_prac_komp_cw.pdf ~/_work_/linux/
4. lub krócej cp ~mzpos/_work_/linux/linux_prac_komp_cw.pdf .
5. gdzie kropka na końcu oznacza “skopiuj tutaj gdzie jestes” czyli do katalogu bieżącego tzn. do katalogu ~/_work_/linux/
6. poleceniem ls sprawdzamy czy prezentacja sie skopiowała

Przykład cd.

- Otwieramy prezentacje z terminala poleceniem:
- ooffice linux_prac_komp_cw.pdf &
- Uruchamiamy program, który obsługuje pliki z rozszerzeniem pdf.
- Znak & (ampersant) oznacza, że proces otwierania pliku linux_prac_komp.pdf odbywa się w “tle” i okno terminala jest nadal aktywne i gotowe do pracy.
- Sprawdź co sie stanie jak nie dodasz znaku &?
- ooffice linux_prac_komp_cw.pdf

Znak & i praca w “tle”

- Brak znaku & przy otwieraniu dowolnego programu komendą wpisana w terminalu spowoduje “zablokowanie” terminala. Aby “odblokowac” terminal można zastosować klawisze:

1. Ctrl C- zamyka aktualnie uruchomiony program
2. Ctrl Z- zawiesza aktualnie uruchomiony program i przywraca terminal. Można wtedy wpisać w terminalu komendę bg – background, która wprowadza zawieszony program do pracy w “tle”.

Polecenie rm i mv

- Polecenie rm nazwa_pliku powoduje usunięcie pliku o nazwie nazwa_pliku
- Istotne opcje (jak w przy poleceniu cp):
  - -r – rekurencyjnie (katalog wraz z zawartością)
  - -f – wymuszenie usunięcia pliku WAZNE!!!.
- mv ścieżka1 scieżka2 przenosi plik z położenia 1 do 2. Wykorzystuje się również do zmiany nazwy (wtedy przeniesienie odbywa się w tym samym katalogu). Opcje jak dla rm, f nie ostrzega przed nadpisaniem istniejącego pliku.

Ćwiczenie

1. W swoim katalogu domowym ~/ stwórz katalog o nazwie “drzewo”, a następnie cztery podkatalogi wewnątrz: “lipa”, “klon”, “dab”, “brzoza” ☺
2. W katalogu “klon” stwórz 2 podkatalogi “lisc” i “kora”
3. W katalogu “lisc” stwórz plik “kolory.txt”. Otwórz ten plik (np poleceniem gedit kolory.txt &). Edytor tekstowy gedit tworzy nowy plik, jeżeli jego ścieżkę wpisze się po komendzie wywołującej program lub otwiera juz istniejący plik.
4. Wpisz do pliku cztery nazwy dwolnych kolorów w jednej kolumnie i zapisz plik.

Ćwiczenie– c.d.

1. Obserwuj jak działają komendy **cd i cd ..**
2. Wciśnięcie klawisza **tab** powoduje, że system usiłuje uzupełnić aktualną ścieżkę lub polecenie. Spróbuj sam jak to działa.
3. Skopiuj plik "kolory.txt" do katalogów "dab" i "brzoza"

Wzorce

1. Znak * zastępuje dowolną liczbę dowolnych znaków
2. Znak ? zastępuje dokładnie jeden dowolny znak.

- Używając [] można określić zakres znaków które mogą się pojawić. Przykłady:
  - [abc] – zastępuje a lub b lub c.
  - [a-c] – zastepuje od a do c
  - [0-9] – zastępuje dowolną cyfrę.

Wzorce II

- [!a-c] – dowolny znak poza wymienionymi
- {koleś1,koleś2} – jeden z ciągów znaków oddzielonych przecinkami.

Wzorce - przykłady

1. cp -r /usr/share/doc/{x11,xserver}* ~/Documents/ skopiuje wszystkie pliki i katalogi zaczynające się od x11 lub xserver do katalogu Documents w Twoim katalogu domowym.
2. ls –l [a-n]* - lista plikow zaczynajacych sie od a do n
3. ls –l [an]* - lista plikow zaczynajacych sie od a lub n

Ćwiczenie

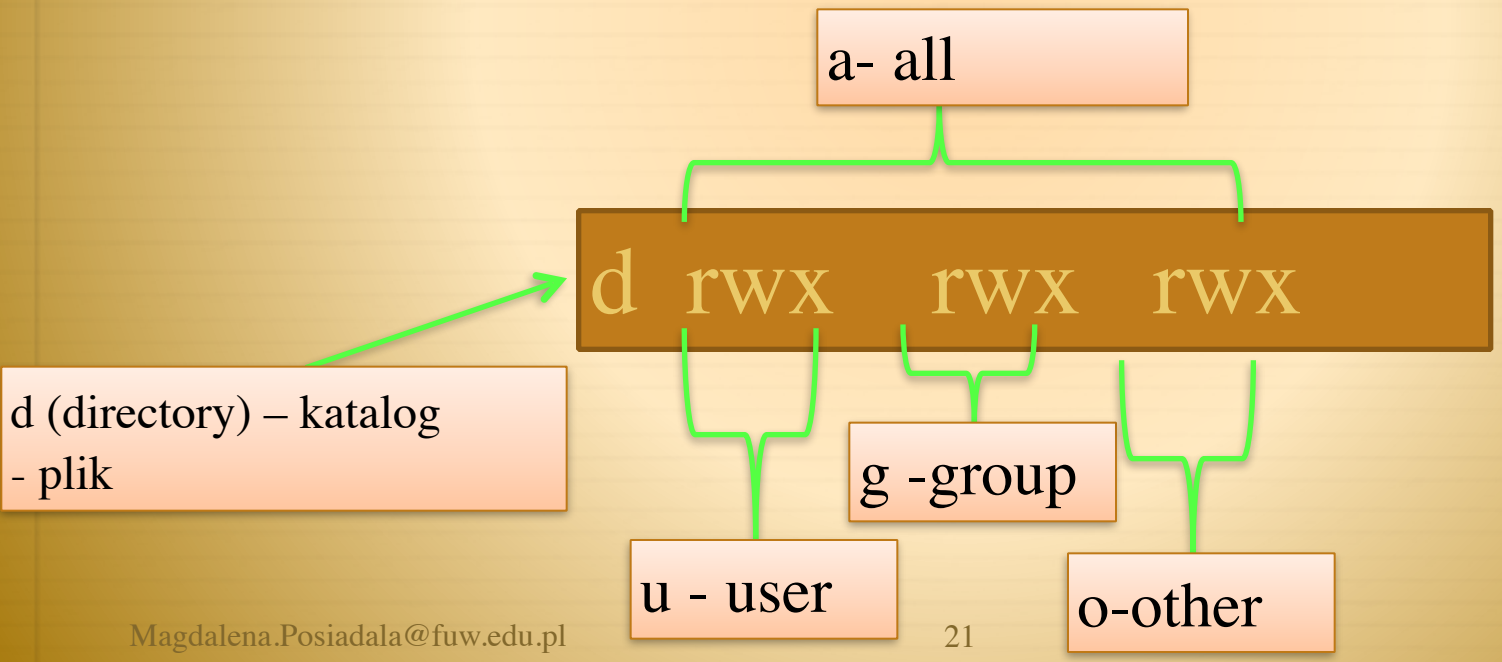
1. Stwórz w swoim katalogu domowym katalog o ścieżce ~/_work_/linux/Pracownia/cwiczenia/2/proste_cwiczenie/film/kadr/gnome
2. Skopiuj do tego katalogu wszystkie pliki i katalogi z /usr/share/doc zaczynające się od gnome lub od x11 lub od xserver
3. Stwórz katalog o ścieżce ~/_work_/linux/Pracownia/cwiczenia/2/proste_cwiczenie/pliki
4. W tym katalogu utwórz pliki koles1, koles2, koles7 i koles.txt
5. Następnie usuń pliki koles1 i koles2
6. Przemianuj poleceniem mv plik koles.txt na koles1.txt

Prawa dostępu

- Każdy plik w systemie linux ma określone prawa dostępu.
- Istnieją trzy podstawowe prawa (poniżej w zapisie symbolicznym):
  - r – read – pozwala na przeczytanie pliku
  - w – write – na zapis
  - x – execute – na wykonanie
- Każdy z tych atrybutów można ustawić dla właściciela pliku (u-user), innych z grupy (g-group) lub wszystkich innych użytkowników (o-others). Każdy użytkownik może należeć do wielu grup! Aby poznać swoje grupę użyj polecenia id.
- Dla katalogów ‘x’ pozwala na wejście do katalogu lub dowolnego podkatalogu, a ‘r’ na zlistowanie zawartości.

Prawa dostępu- polecenie ls -l

- Przykład użycia polecenia ls –l
- drwxr-xr-x 17 magdap wheel 578 12 Apr 2013 programowanie_dydaktyka
- -rw-r--r-- 1 magdap wheel 166 6 Jun 13:40 untitled.C



Zapis numeryczny

- Prawa dostępu można opisać z pomocą liczb całkowitych z zakresu 0-7.
- W takim zapisie mamy odpowiednie przyporządkowania:
  - x = 1 – eXecute – prawo do wykonywania
  - w = 2 – Write – prawo do zapisu
  - r = 4 – Read – prawo do odczytu
- Konkretne prawa dostępu uzyskuje się dodając do siebie 1,2 i 4. Np:
  - 1+2 = 3 – eXecute + Write
  - 1+4 = 5 – eXecute + Read
  - 1+2+4 = 7 – eXecute + Write + Read

Polecenie chmod

- Polecenie chmod pozwala na ustawienie praw dostępu dla pliku lub katalogu.
- chmod używamy w postaci:
  - chmod <przywileje> nazwa_pliku
  - np. chmod u+x,g+x,o+x plik.txt
- W zapisie numerycznym przywileje określają 3 cyfry – po kolei dla właściciela, grupy i wszystkich innych. Na przykład:
  - chmod 744 nazwa_pliku ustawia pełen prawa dla właściciela i prawa odczytu dla innych.

Polecenie chmod- przykłady

- chmod a+w plik.txt — nadaje wszystkim uprawnienia do zmiany 'plik.txt',
- chmod o-x plik.txt — usuwa możliwość wykonywania 'plik.txt' przez pozostałych użytkowników,
- chmod go=rx plik.txt — grupa oraz pozostali użytkownicy będą mogli tylko czytać i wykonywać.
- chmod -R 777 /home/user — wszyscy będą mogli zmieniać zawartość katalogu /home/user oraz jego podkatalogów, jak też czytać go i wykonywać zawarte w nim pliki

chmod – UWAGA!

- chmod 744 pliki
- chmod u=rwx,go=r pliki

Polecenia dają ten sam wynik!!!!!!!

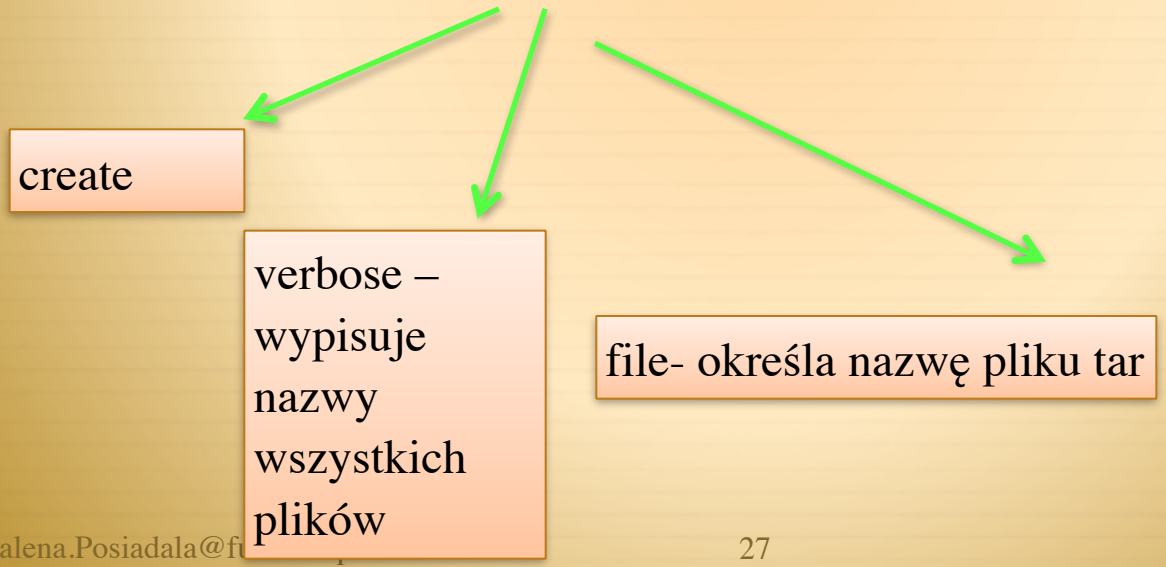
- Opcja -R pozwala (jak zwykle) działać rekurencyjnie na podkatalogach.

Ćwiczenie

- Stwórz katalog ~/_work_/linux/cw, w nim stwórz podkatalog tmp.
- W ~/_work_/linux/cw/tmp stwórz plik cos.txt.
- Poeksperymentuj z uprawnieniami pliku cos.txt – spróbuj ustawić je tak, abyś nie mógł go obejrzeć, nie mógł zmienić itp. czy możesz ustawić takie uprawnienia, aby móc usunąć plik, ale nie móc obejrzeć?
- Poeksperymentuj z uprawnieniami katalogu tmp. Co się dzieje, gdy odbierasz uprawnienie r, a co kiedy x? Za każdym razem zobacz, czy możesz wyświetlić zawartość katalogu i czy możesz otworzyć plik.

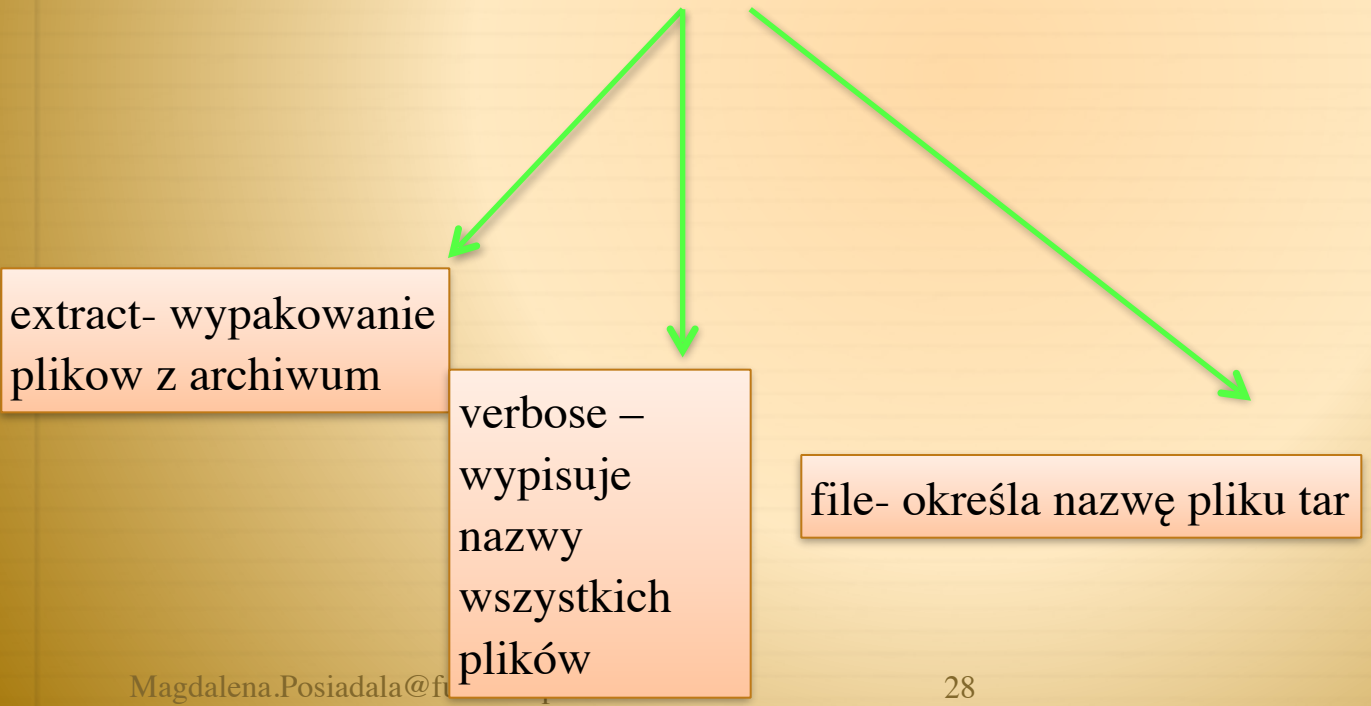
Archiwizacja plików- tar

- tar- umieszczanie grupy plików w jednym zbiorczym pliku (tzw. archiwum), który następnie może zostać skompresowany programem gzip.
- Składnia : tar –cvf nazwa_pliku.tar nazwa_katalogu



Wypakowanie plików- tar

- tar-wypakowanie plików z jednego zbiorczego pliku tar
- Składnia : tar –xvf nazwa_pliku.tar

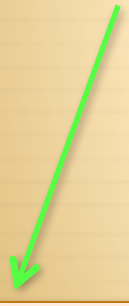


Kompresja i dekompresja zarchizowanych plików – polecenie gzip

- Plik typu nazwa.tar mozna skompresować przy uzyciu polecenia gzip.
- Składnia :
  - gzip nazwa_pliku.tar – powstaje plik nazwa.pliku.tar.gz. Oryginal nazwa_pliku.tar przestaje istniec po tej operacji.
  - Dekompresja: gzip –d nazwa_pliku.tar.gz

Dekompresja zarchizowanych plików – polecenie tar

- Plik typu nazwa.tar mozna skompresować przy uzyciu polecenia tar:Składnia :
- tar –xvzf nazwa_pliku.tar.gz



włączanie (de)kompresji programem gzip

Ćwiczenie

1. W swoim katalogu domowym ~/_work_/linux/ stwórz katalog o nazwie “drzewo”, a następnie cztery podkatalogi wewnątrz: “lipa”, “klon”, “dab”, “brzoza” ☺
2. W katalogu “klon” stwórz 2 podkatalogi “lisc” i “kora”
3. W katalogu “lisc” stwórz plik “kolory.txt” poleceniem touch. Składnia: touch kolory.txt
4. Z katalogu /etc skopiuj do katalogu kora wszystkie pliki i katalogi zawierające ciąg liter “ssh”.
5. Stwórz archiwum drzewo.tar i skompresuje je poleceniem gzip.

Standardowe wejście / wyjście

- Znaki które wpisujemy z klawiatury trafiają w systemie do tzw. „standardowego wejścia”. Program odpowiada na tzw. „standardowe wyjście”, które wyświetlane jest na ekranie.

Operatory >, >>

- Dane ze standardowego wyjścia można zapisac do pliku używając znaków > lub >>
- Istnieje subtelna różnica:
  - > tworzy nowy plik i zapisuje do niego wynik działania programu. Jeżeli plik już istnieje zostaje zastąpiony
  - >> działa podobnie, ale jeżeli plik już istnieje, to wynik zostaje dopisany.
  - Np. wykonaj polecenie ls –l >plik.txt

Wyświetlanie zawartości pliku tekstowego txt

- Polecenia służące do wyświetlania zawartości plików:
  - less
  - more
  - cat
  - np cat plik.txt lub less plik.txt lub more plik.txt

Polecenie paste

- Jeżeli istnieją dwa pliki o podanych niżej treściach - plik1: 1 2 3 (w kolumnach)
- oraz plik2: 2 4 6 (w kolumnach
- to rezultatem wykonania polecenia
- paste plik1 plik2 > plik3

Zobacz jak wyglada plik3!

Operatory <, |

- Operator < podaje na std wejście zawartość pliku.
- Operator | pozwala przekierować std wyjście na std wejście.

Przykład |

- Rozważmy polecenie
  - ls -R | grep jeż | less
  - - polecenie ls -R przeszukuje aktualny katalog rekurencyjnie i listuje wszystkie pliki i katalogi
  - - program grep przeszukuje dane na wejściu w poszukiwaniu linii ze słowem jeż – dane które wcześniej wyrzucił ls.
  - - linie które zawierały dane słowo są dalej przekazywane do programu less, który wyświetla je strona po stronie.

Polecenie wc

- Polecenie wc (word count)- drukuje liczbę linii, słów i znaków w tekście
- Np wc plik.txt daje wynik
- 22 23 224 gdzie 22 to liczba linii, 23 liczba słów, 224 liczba znaków
- wc –l plik.txt – drukuje liczbę wierszy w pliku plik.txt
- wc –w
- wc -c

sprawdź!

Polecenia head i tail

- head plik.txt- drukuje poczatek pliku (domyslnie pierwszych 10 wierszy)
  - head -1 plik.txt drukuje 1 wiersz w pliku
- tail plik.txt drukuje koniec pliku (domyslnie ostatnich 10 wierszy)
  - tail -1 plik.txt drukuje ostatni wiersz w pliku

Polecenie find

- find -P/-L <ŚCIEŻKA> <WARUNKI>, gdzie:
- -P i –L określają traktowanie linków symbolicznych (-P – nie podążaj za linkami, -L – podążaj)
- -<ŚCIEŻKA> - w tym katalogu i jego podkatalogach zostanie dokonane przeszukanie.
- - <WARUNKI> - zestaw warunków precyzujących jakie pliki mają być wyszukane.

find- warunki polecenia

- Warunki polecenia find to na przykład:
  - -name pattern – pozwala sprecyzować nazwę (działają znaki specjalne *,? I []). Pattern trzeba podać w „”!
  - -iname – jak wyżej, ale działanie bez rozróżnienia na wielkie i małe litery.
  - -size n[ck] – rozmiar, c – w bajtach, k – w kilobajtach. (+n - rozmiar większy niż, -n – mniejszy niż)

Find- przykłady

- Szukanie pliku o nazwie README w całym drzewie katalogowym
  - find / -name README
  - find ~ sprawdź znaczenie!
- Szukanie tylko zwykłych plików we fragmencie drzewa katalogów, poczynajac od katalogu bieżącego
  - find . -type f
- Szukanie plików majacych w nazwie ciag liter conf, poczynajac od katalogu /etc
  - find /etc –name “*conf*”
- Szukanie wszystkich plików w /usr/share/doc ze słowem TODO o rozmiarze większym niż 5Kbytów.
  - find -P /usr/share/doc -name ”*TODO*” -size +5k

Find przyklady 2

- Opcja exec !!! This command changes the permissions of all files with a name ending in *.mp3* in the directory */var/ftp/mp3*. The action is carried out by specifying the option -exec chmod 644 {} \; in the command.
- For every file whose name ends in .mp3, the command chmod 644 {} is executed replacing {} with the name of the file. The semicolon (backslashed to avoid the shell interpreting it as a command separator) indicates the end of the command.
  - find /var/ftp/mp3 –name "*.mp3" -type f -exec chmod 644 {} \;

Polecenie grep

- Polecenie to na podstawie podanego wzorca szuka w pliku tekstowym wierszy, które dany wzorzec zawieraja i wyswietla je.
- Szukanie w pliku nazwa_pliku linii zawierajacych ciag liter abc, bez zwracania uwagi na małe i duze litery
  - grep -i abc nazwa_pliku
- Znajduje linie zawierające wyraz 'Ala' lub 'Aga'.
  - grep 'A[lg]a' nazwa_pliku

Polecenie du

- Polecenie du wyświetla nam **rozmiar pliku/katalogu** podanego w wierszu poleceń. Domyślnie wyświetla i podaje rozmiar bierzącego katalogu i jego zawartości.
- Wybrane opcje:
- **-h** - wyświetlenie w czytelniejszy sposób w MB
- **-s** - podaje wartość sumaryczną

Polecenie sort

- sort nazwa_pliku- sortuje wiersze z danego pliku w porządku alfabetycznym, rosnąco i drukuje je na terminal.
- sort –r nazwa_pliku – jak wyżej ale malejąco.
- sort –n nazwa_pliku – sortowanie numeryczne, wiersze w pliku sa traktowane jako liczby, sortowanie wedlug 1 kolumny.
- sort –n –k2 nazwa_pliku -sortowanie numeryczne, wiersze w pliku sa traktowane jako liczby, sortowanie wedlug 2 kolumny.

Zadanie

- Korzystajac z polecen **du i sort** (z odpowiednimi opcjami) znajdz podkatalog w twoim katalogu domowym, który zajmuje najwiecej miejsca. Zapisz wynik do pliku max.txt.

Polecenie tr

- tr- zmienia lub usuwa znaki ze standardowego strumienia wejścia. Przykłady:
  - echo "wikimedia" | tr "mw" "pW”
    - Wikipedia
- echo "wiki wiki" | tr -d "ki"
  - w w
- cat plik.txt | tr A-Z a-z # zamienia wielkie litery na małe
- Parametry tr:
  - tr -d ‘:’ # usuwa wszystkie znaki ‘:’ z tekstu
  - tr ‘ ‘ ‘\‘ # spacje sa zastepowane znakiem przejscia do nastepnego wiersza

Zadanie

- Stwórz plik o nazwie tekst.txt i zapisz w nim pozniższe zdanie:
  - “Linuks - Rodzina Uniksopodobnych Systemów Operacyjnych Opartych Na Jądrze Linux.”
- i zamień małe litery na duże w tym pliku.

Powłoki - rodzaje

- W Linux'ie mamy kilka powłok do wyboru:
  - sh : Bourne Shell, oryginalna powłoka systemu unix
  - csh : C shell, nowa składnia polecen, udogodnienia w pracy interakcyjnej
  - ksh : Korn shell, zgodnosc składniowa z powłoka Bourne'a +m.innymi udogodnienia jak w powłoce csh
  - bash : Bourne Again Shell, połaczenie najlepszych cech csh i ksh
  - tcsh : udoskonalona wersja csh

Powłoki

- Typ powłoki definiuje administrator systemu w pliku z hasłami
- Obecnie każdy użytkownik w OKWF ma standardowo ustawiona ¸powłokę bash (zmienna $SHELL)
- Uruchamianie innej powłoki, np. tcsh (jednorazowo): polecenie tcsh
- Standardowo uruchamiane skrypty:
  - przy logowaniu: /etc/profile i ./bash_profile
  - start powłoki: ./bashrc

Skrypty

- Skrypt powłoki = plik tekstowy zawierajacy jedno lub wiele polecen
- Powłoka zakłada, ze każda linia to osobne polecenie
- Komentarze zaczynaja ˛ się od znaku #
- Uruchamianie, powłoka bash: ./skrypt
- Wypisywanie komunikatu na ekran:
  - echo "Z nowym wierszem"
  - echo -n "Bez nowego wiersza"

Uruchamianie skryptów

- Uruchomienie skryptu jak zwykłego programu:
- Zmiana praw dostepu:
  - chmod u+x skrypt.sh
- Wywołanie:
- ./skrypt.sh

Pisanie skryptów- przykład 1

Utworzyć skrypt pt przyklad1.sh:

```
#!/bin/bash
# (Tu jest komentarz) definuje w jakiej powłoce bedzie uruchamiany
    skrypt

    echo "Pierwszy program”
    pwd
```

- *Zmienic prawa dostepu pliku przyklad1.sh aby móc go wykonywać*
- *Uruchomić w terminalu prog.sh -> ./przyklad1.sh*

Pisanie skryptów- przykład 2

- *#!/bin/bash*
  echo "Witam. Twoj login to $USER"
  echo "Lista plików w bieżącym katalogu, $PWD"
  ls # wypisz listę plików

Pisanie skryptów- przykład 3 pętla for

- #!/bin/bash
- for ((i=1; $i <= 10; i++)) ; do
- echo " Iteracja nr: $i"
- done

Pisanie skryptów- przykład 4 parametry

- Kod skryptu przyklad4.sh
  - #!/bin/bash
  - mkdir $1 $1 to pierwszy parametr podany podczas uruchamiania skryptu zaraz po jego nazwie
  - mkdir $2
  - ls –ltr
  - Uruchamiamy skrypt z parametrami wejściowymi
  - ./sprzyklad4.sh nazwa_kat1 nazwa_kat2

Przykład 5

- Prześledź poniższy skrypt, zapisz go w pliku przyklad5.sh i uruchom.

```
#!/bin/bash
KONIEC="x";
function wybor
{
   case $KONIEC in
        s) echo "Jestes w katalogu : $PWD" ;;
        p) echo "teraz wypisze wszystkie twoje procesy";
            ps -e ;;
        w) echo "Oto lista plikow w tym katalogu";
            ls ;;
  esac;
}
function menu # poczatek menu
{
    until [ $KONIEC = k ];do
        echo "Wcisnij Enter aby kontynuowac";
        read; # czytanie danych z klawiatury
        clear; # czysci ekran
        echo $KONIEC;
        echo " Proste Menu ";
        echo;
        echo "s - wyswietl sciezke w ktorej znajduje sie skrypt.";
        echo "p - wyswietl liste procesow uzytkownika.";
        echo "w - wyswietl wszystkie pliki w tym katalogu.";
        echo "k - zakoncz skrypt.";
        read KONIEC;
        wybor;
    done;
}
menu;
```

1) 2) 3) 4) 5)

Magdalena.Posiadala@fuw.edu.pl

Przykład 5- wyjaśnienia

1. W funkcji wybor instrukcja case wybiera naszą opcje i wywołuje ją.
2. Zaczyna się funkcja, wyświetla menu i następnie "wchodzi" do funkcji wybor
3. W menu **pętla until** sprawdza czy jest spełniony warunek..
4. Dzięki komendzie **read** możemy wpisywać z klawiatury dane do zmiennych.
5. Powrót do menu

Koniec