**mkdir** – tworzenie katalogu

**rm –r nauka** – usnunięcie katalogu nauka UWAGA musimy być w jego rodzicu, żeby go usunąć

**cat>nazwaPliku** – tworzymy plik i od razu wypełniamy go treścią CTRL+D przerywa i zamyka plik

**cp sciezkaCoSkopiowacNazwaPliku gdzieSkopiowacSciezka** – kopiuje plik

Pulpit

|

-------- kopia

| |

| -------- info1

|

-----------test

|

---------- info2

z poziomu pulpit chcemy zmienić nazwę pliku w kopia

**mv kopia/info1 info2** - zmiana nazwy z info1 na info2 **i umieszczenie go w nowej ścieżce**

**mv kopia/info1 kopia/info2** – zmiana nazwy pliku w ścieżce Nazwy ścieżek muszą być te same by zmieniło tylko nazwę pliku

**cat ścieżka do pliku** – wyświetlenie zawartości pliku

**cat>>sciezka do pliku i/lub rozszerzenie** – dodanie tekstu do pliku

Jesteśmy w kopia i kopiujemy info1 do test

**cp info1 ../test** – kopiujemy plik info1 z kopia do test z poziomu kopia

**tar –czvf nazwaArchiwum.tar sciezka do tego co ma być w archiwum** – tworzy archiwum w miejscu w którym jesteśmy aktualnie

**tar –czvf archiwum/nazwaArchiwum.tar sciezka do tego co ma być w archiwum -** tworzy nazwaArchiwum.tar w katalogu archiwum

**tar –tzvf sciezka/plik.tar** - pokazuje nam zawartośc podanego archiwum

**tar -zxvf nazwa\_archiwum.zip nazwa\_folderu** – rozpakowanie pliku tar

**useradd –m –g nazwaGrupy nazwaUzytkownika –** tworzenie nowego użytkownika do grupy

**groupadd nazwa** – dodanie nowej grupy

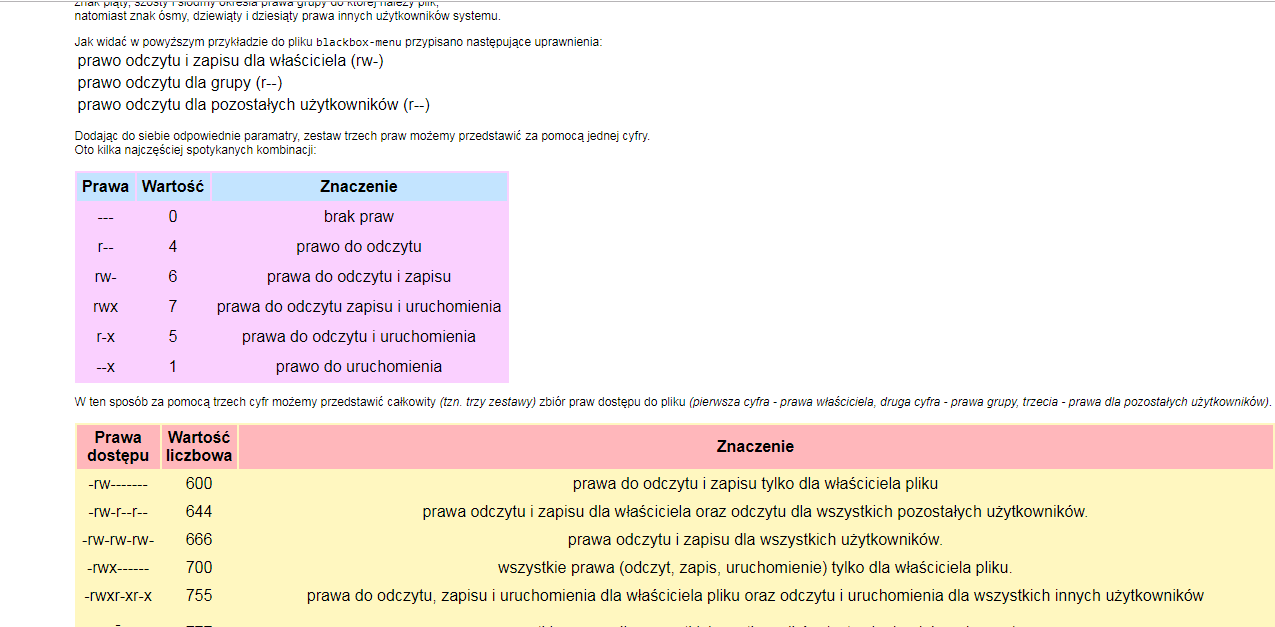
**groupdel nazwa** – usuniecie grupy

**adduser nazwaGrupy** – dodanie uzytkownika do grupy

**userdel nazwa** – usuniecie uzytkownika

**passwd nazwaUzytkownika** – zmiana hasła użytkownikowi lub jego nadanie

**logout** – wylogowanie sie

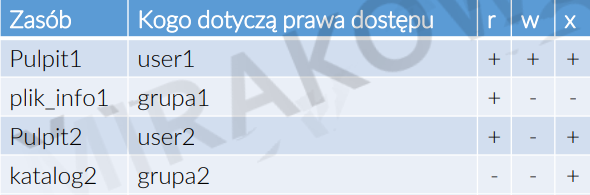
****

**chown nazwaUzytkownika nazwaPlikuLubKatalogu** – zrobienie kogoś właścicielem pliku

**ls –l nazwaPliku** – sprawdznaie uprawnien

**chown nazwaUzytkownika:nazwaGrupy nazwaPliku** – zmiana użytkownika i grupy jednocześnie

**chown :nazwaGrupy nazwaPliku** – zmiana grupy



dodac osobno uzytkownika useradd u1 i osobno groupadd g1 żeby u1 nie był w grupie g1

**groupadd g1 g2**

**useradd u1 u2**

**chown u1 pulpit1**

**chmod u+rwx pulpit1 cd pulpit1 ls –ld**

**chown :g1 pulpit1/plik\_info1** bo plik\_info1 jest w pulpit1

**chown u2 pulpit2**

**chmod u+rx pulpit2**

**chown :g2 pulpit2/katalog2**

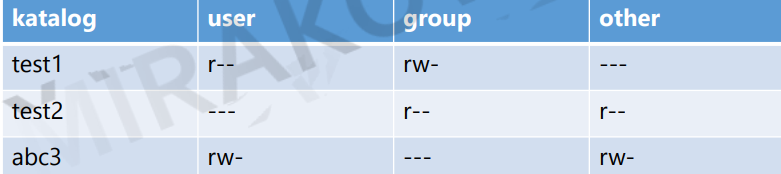
**chmod g+x pulpit2/katalog2**

**drwxr-xr-x** = oznacza nadanie praw rwx do katalogu bo d na początku

**-rw-r—r--** = oznacza nadanie praw r do pliku bo – na początku

**drwxr-xr-x** = oznacza nadanie praw rx do katalogu bo d lub samo x

**drwxrwxr-x =** oznacza nadanie praw wx lub samo w



**chown u1 test**

**chmod u+r test**

**chown :g1 test**

**chmod g+rw test**

**chmod a+r test2**

**chown :g2 test2**

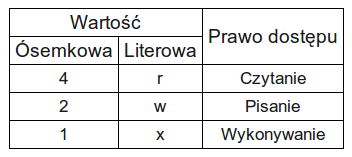
**chmod g+r test2**

**chmod a+rw test3**

**chown u3 test3**

**chmod g+rw test3**

Każdy plik posiada uprawnienia podzielone na trzy części, które odpowiednio oznaczają: **prawa dostępu właściciela pliku**, **prawa dostępu grupy**, oraz **prawa dostępu pozostałych**. W naszym przypadku pierwsza część to **rw-**, druga to **r--** oraz trzecia to **r--**. Na poniższym rysunku przedstawiony został wykaz praw dostępu z wartościami literowymi oraz liczbowymi (te przydadzą się podczas nadawania praw dostępu w dalszej części)



**zaloguj się jak użytkownik root i nadaj poniższym zasobom odpowiednie prawa:**

1. prawo uzytwkonikowi do pliku lub katalogu możemy nadać pisząc chown użytkownik nazwaPlikuKatalogu

2. uprawnienia zmieniamy za pomocą chmod

**Przykład:**

|  |
| --- |
| chmod a+w nazwa\_pliku |

Dzieki temu możemy spowodować nadanie wszystkim prawa do zapisu dla pliku o podanej nazwie.  
Wyjaśnienie składni tego polecenia:

**1. Określamy kto ma dostac te prawa:**

|  |  |
| --- | --- |
| **Litera** | **Znaczenie** |
| a | *(ang. all)* - wszyscy (użytkownik, grupa, inni) |
| u | *(ang. user)* - użytkownik - właściciel pliku |
| g | *(ang. group)* - grupa, której przypisano plik |
| o | *(ang. others)* - inni |

Napisałem wcześniej, że właściciel określany jest jako user (użytkownik), a nie owner (właściciel), oto dlaczego - nie może być dwóch typów użytkowników zaczynających się na o (owner, others), więc jeden nich jest zastępowany przez user.

Programiści wymyślili rozwiązanie, które i tak nie koliduje z prawidłowościami - (u)ser - użytkownik (któremu nadano plik - właściciel), (o)thers - inni użytkownicy w systemie - nie należą do grupy, do której przypisano plik i nie są właścicielami pliku.

**2. Określamy znaczenie polecenia:**

|  |  |
| --- | --- |
| **Znak** | **Znaczenie** |
| + | nadanie prawa (dodanie) |
| - | odebranie prawa |
| = | zastąpienie prawa - kasuje poprzednie i zastępuje nowym |

**3. Ustalamy prawa:**

* **r, w, x -**wiadomo,
* **u, g, o -**ustawie takie same prawa jak ma **(u)**ser, **(g)**roup lub **(o)**thers
* **t -**oznacza, że usunąc katalog może tylko jego właściciel(e) (tzw. Lepki bit - ang. sticky bit); oznacza też plik tekstowy
* **l -**obowiązujące zabezpieczenie (małe "L")

Argumenty możemy ze sobą łączyć, np. **chmod ug+rwx nazwa\_pliku** - nada właścicielowi i grupie prawo do czytania, zapisu i wykonywania, natomiast dla innych użytkowników *(others)*pozostaną one bez zmian.

czy chmod g+rwx plik34 nadałuprawnienia grupie do rwx ? Tak

**uprawnienia u22 do folderu test2 nadałem najpierw deklarując go wlascicielem za pomoca chown a pozniej za pomoca chmod g+r lub +rx itd. w zaleznosci od potrzeb i na koncu nazwa katalogu**

g – grupa która jest wlascicielem

Skrypty

**mcedit nazwa.sh** – tworzy skrypt

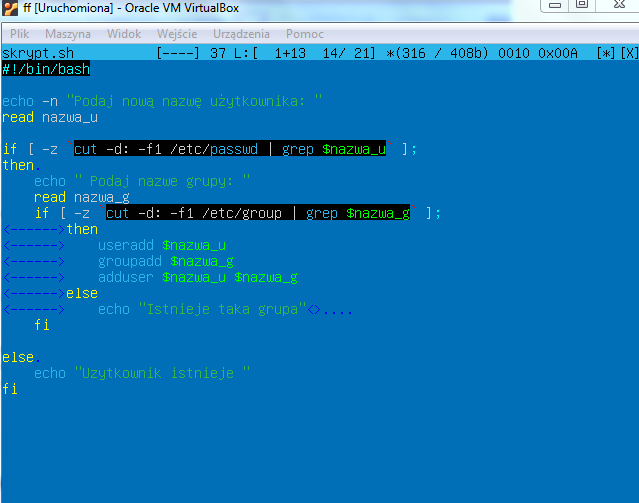
**chmod a+x nazwa.sh** – nadanie uprawnień wszystkim do podanego skryptu

**uruchamiamy skrypt w ścieżce w którym go utworzyliśmy**

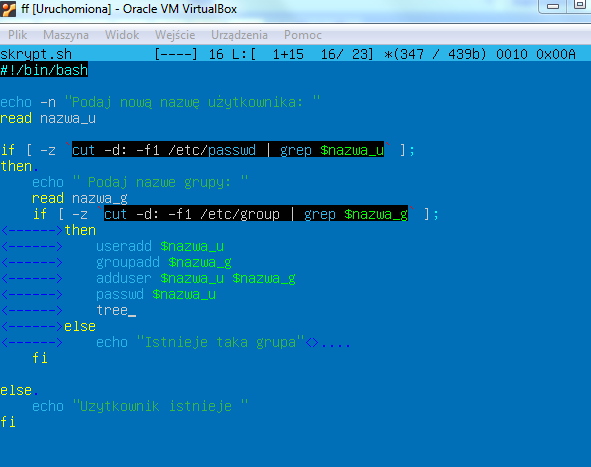
**./nazwa.sh –** uruchomienie skryptu

#!/bin/bash

1. Sprawdzanie czy istnieje użytkownik i grupa jeśli nie to dodaje użytkownika do grupy



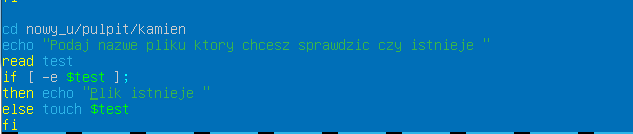
2. Sprawdzanie czy istnieje użytkownik i grupa jeśli nie to dodaje użytkownika do grupy, nadaje mu hasło i wyświetla drzewo home



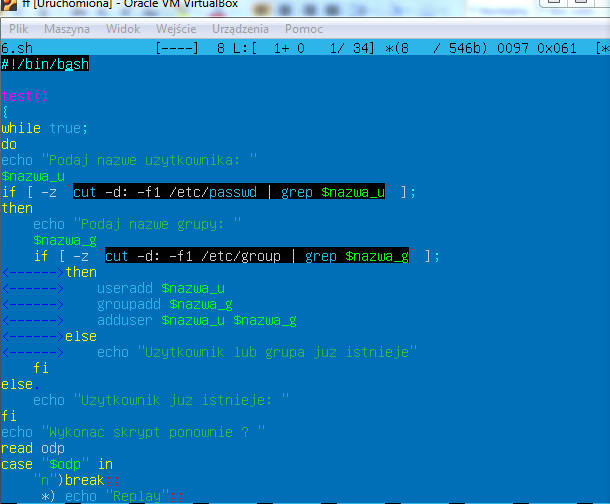
3. Tworzy podaną scieżkę, dodaje katalog o podanej nazwie jeśli jeszcze nie istnieje a nastepnie dodaje do tego katalogu plik o podanej nazwie jeśli jeszcze nie istnieje



4. Sprawdzenie czy istnieje w sciezce nowy\_u/pulpit/kamien plik o podanej nazwie, jeśli nie to utworz go



5. Funkcja z dwoma rgumentami i pętlą while



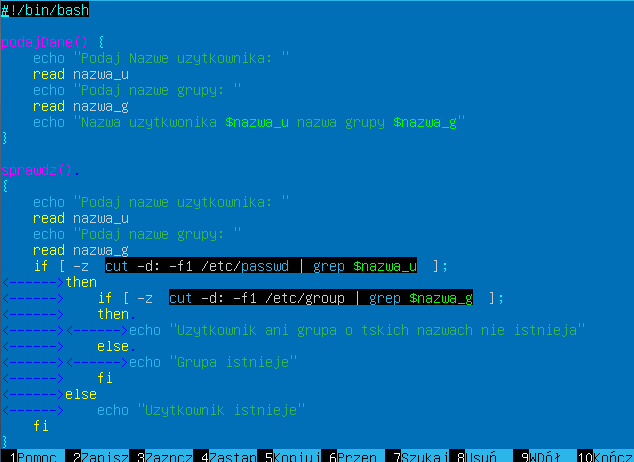
esac

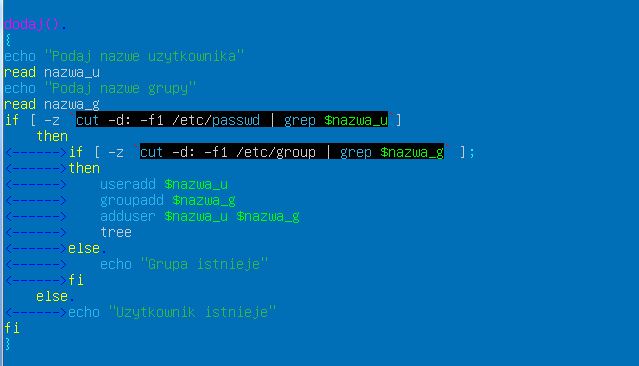
done

}

test u1 g1

6. Menu funkcjonalne – pokazywanie, sprawdzanie, dodawanie







7.