**1.POLECENIA OGÓLNE**

**man costam** – info o danym poleceniu  
**cd nazwa\_katalogu** – wejście do katalogu  
**cd ..** – wyjście z katalogu  
**mkdir nazwa\_katalogu** – tworzy katalog  
**rmdir nazwa\_katalogu** – usuwa katalog  
**rm nazwa\_pliku** – usuwa plik  
**rm \*** – usuwa wszystkie pliki z danego katalogu  
**rm \* -i** – usuwa wszystkie pliki z danego katalogu z potwierdzeniem  
**rm \* -f** – usuwa wszystkie pliki z danego katalogu i wyłącza potwierdzenia (nawet zabezpieczone przed usunięciem)  
**rm -f** – usunięcie plików zabezpieczonych przed kopiowaniem  
**rm -r** – usunięcie plików również w podkatalogach  
**rm -rf** – usuwa cały system plików  
**chmod u+r jeden.doc** – mogę czytać dany plik  
**w** – mogę zapisywać i edytować dany plik  
**x** – mogę wykonywać dany plik  
**ls -a** – pokazuje ukryte pliki  
**ls -l** – pokazuje więcej info o pliku:

W WYNIKACH POWYŻSZEJ KOMENDY NA POCZĄTKU SĄ:  
**„-„** – oznacza zwykły plik  
**„d”** – oznacza katalog  
Potem pierwsze 3 znaki oznaczają dostępność do pliku właściciela, następne 3 znaki dla grupy, a następne 3 dla innych użytkowników

**ls > wynik** – wysłanie spisu zawartości katalogu do pliku wynik jeżeli taki plik istnieje to skasuje się jego poprzednia zawartość  
**ls >> wynik** – dopisanie zawartości katalogu do zawartości pliku wynik  
**mount /mnt/floppy** – montuje flopa  
**umount /mnt/floppy** – rozmontowuje flopa  
**cat jeden.doc > wszystkie.doc** – nadpisanie zawartości w pliku wszystkie.doc zawartościami innych plików  
**cat jeden.doc dwa.doc >> wszystkie.doc** – dopisanie do pliku wszystkie.doc zawartości innych plików  
**cat nazwa\_pliku** – czytanie zawartości pliku  
**cat \*.doc** – znajdzie np.: dwa.doc sratytaty.doc  
**cat jeden.doc dwa.doc > wszystkie.doc** – wrzuca zawartość plików do jednego  
**cat < jeden.doc > dwa.doc** – wypisze zawartość jeden.doc na ekran (standartowe wyjście) i nadpisze zawartość pliku jeden.doc do pliku dwa.doc  
**cat parowka.doc | more** – wyjście programu cat (parowka.doc) jest przekazywane na wyjście programu more  
**head -42 nazwa\_pliku** – wyświetla 42 pierwszych wierszy zawartych w pliku  
**tail -42 nazwa\_pliku** – wyświetla 42 końcowych wierszy zawartych w pliku (przydaje się w obserwowaniu świeżo dopisanych logów do pliku)  
**cp plik\_zrodlowy plik\_docelowy** – kopiuje zawartość z pliku do pliku  
**cp plik(i)\_zrodlowy katalog(i)\_docelowy** – kopiuje pliki do katalogu  
**mv stara\_nazwa nowa\_nazwa** – zmiana nazwy pliku  
**mv nazwa\_pliku nazwa\_katalogu** – przenoszenie pliku do katalogu  
**lp nazwa\_pliku** – drukowanie pliku  
**lpr nazwa\_pliku** – wstawienie pliku do koleiki drukowania  
**pwd** – pokazuje w jakim jesteś katalogu  
**.** – oznacza katalog bieżący  
**..** – oznacza katalog nadrzędny  
**touch nazwa\_pliku** – tworzy pusty plik  
**stty -a** – wypisanie skrótów do np. zatrzymania procesu  
**mail** – otwiera plik z poczta  
**printenv albo export (powłoka C)** – zobaczenie zawartości zmiennych środowiska  
**nazwa=zawartość** – pierwsze przypisze zmiennej powłoki odpowiednia zawartość  
**export nazwa** drugie utworzy zmienna o tej samej nazwie i zawartości co zmienna powłoki  
**MOJA\_ZMIENNA=”jakas\_wartosc”** – tworzy zmienna środowiska z wartością „jakas\_wartosc”  
**echo $MOJA\_ZMIENNA** wypisuje zawartość MOJA\_ZMIENNA (znakiem dolara poprzedzamy nazwy zmiennych środ. które to nazwy powłoka ma zastąpić zawartością odpowiedniej zmiennej  
**find ścieżka/nazwa\_pliku lub kryterium** – wyszukiwanie plików  
**grep** – służy do wyszukiwania plików według zadanego wzorca

**2. KRYTERIA SZUKANIA OKREŚLONYCH STRINGÓW**

**\*** – zastępuje wiele znaków np.: ls \*.doc  
**\*.\*** – oznacza pliki z kropka w nazwie  
**\*xyz** – oznacza pliki w rodzaju xyz albo fghxyz  
**?** – zastępuje jeden znak  
**[abc]** – oznacza a lub b lub c  
**[a-z]** – oznacza od a do z  
**doc[0-9]** – oznacza doc0 do doc 9  
**\*[Aa]dd\*** – oznacza useradd lub User-Add itd  
**\*[Aa] [Dd] [Dd]\*** – oznacza userADD lub xxxAdd itd  
**– R** – wszystkie podkatalogi

**3. UŻYTKOWNICY, PRAWA DOSTĘPU ITD**

**newgrp grupa** – zmiana grupy  
**chmod g+r jeden.doc** – dodanie użytkownikom grupy do której należy plik jeden.doc  
prawa do jego czytania (dołączanie praw):  
**u** – użytkownik  
**o** – pozostali użytkownicy systemu  
**–** – odebranie danego uprawnienia  
**chmod u=rw jeden.doc** – nadanie tobie wyłącznie czytanie i pisania do pliku jeden.doc (zastępowanie praw)  
**chmod u=rw /katalog** – masz prawo do przeszukiwania katalogu jeżeli nie miałbyś tych praw możesz znaleźć tylko plik którego znasz nazwę (zastępowanie praw)  
**chmod 777 nazwa\_pliku** – następny sposób zastępowania praw, numeryczny 777 oznacza akurat przyznanie wszystkich praw wszystkim użytkownikom do danego pliku a 666 prawo do odczytu i zapisu pliku dla wszystkich użytkownikow

PRAWA DOSTĘPU I ICH ODPOWIEDNIKI NUMERYCZNE:

**---   0**  
**--x   1**  
**-w-   2**  
**-wx   3**  
**r--   4**  
**r-x   5**  
**rw-   6**  
**rwx   7**  
  
**umask** – w momencie tworzenia nowego pliku otrzymuje on domyślne prawa dostępu określone za pomocą parametru umask  
**4 2 1** – wartości liczbowe odpowiadające określonym uprawnieniom  
**r w x** każda liczba odpowiada koleino użytkownikowi -> grupie -> „światu” (reszcie użytkowników)  
**umask 421** – oznacza: właściciel może plik czytać grupa może do pliku pisać, a pozostali użytkownicy mogą plik wykonywać  
**umask 700** – oznacza pełne prawa dla właściciela i żadnych praw dla grupy i reszty użytkowników  
**umask 077** – wyłączy prawa do czytania pisania i wykonywania dla grupy i całego „świata” (reszty użytkowników)

PO WPISANIU **UMASK** I WARTOŚCI OD TEJ PORY KAŻDY UTWORZONY PLIK BĘDZIE MIAŁ TAKIE PRAWA DOSTĘPU

**chown nowy\_wlasciciel nazwa\_pliku** – daje prawa do pliku innemu użytkownikowi  
**chown -r nowy\_wlasciciel nazwa\_pliku** – daje prawa do wszystkich plików w katalogu innemu użytkownikowi  
**chgrp nowa\_grupa nazwa\_pliku** – daje prawa do pliku określonej grupie  
**who** – info o wszystkich zalogowanych do systemu, obejmuje nazwę użytkownika  
**finger** – to samo co who tylko jeszcze jest imię terminal, i czas zarejestrowania się w systemie  
**who am i** – info tylko o tobie

**4. KOMUNIKACJA MIĘDZY UŻYTKOWNIKAMI**

**write nazwa\_uzytkownika** – wysyłanie wiadomości do określonego użytkownika, tekst pojawia się od razu po wpisaniu, teksty obu użytkowników mogą się mieszać  
**mesg n** – zablokowanie przychodzących wiadomości write  
**mesg y** – odblokowanie przychodzących wiadomości write  
**talk nazwa\_uzytkowika** – następny program tutaj ekran dzieli się na dwie części a rozmowa się nawiązuje jeśli drugi rozmówca odpowie „talk twoja\_nazwa\_uzytkownika”  
**conrol-c** – zakańcza program talk

**5. MAIL**

**mail** – otwiera twoja skrzynkę odbiorcza  
**mail nazwa\_uzytkownika1 nazwa\_uzytkownika2** – wysłanie maila do danego(ych) użytkownika np.: mail user@wp.pl, mail grzes  
**control-d lub („.” sama w wierszu)** – zakończenie listu i wysłanie  
**control-c** – opuszczenie programu bez wysyłki listu  
**/usr/spool/mail nazwa\_uzytkownika** lub **/usr/mail/ nazwa\_uzytkownika** – tu znajduje się plik w którym są zapisywane nadchodzące listy (nazwa pliku jest tez napisana w zmiennej środowiskowej MAIL można się tam dostać wpisując export i znaleźć MAIL=…)  
**d2** – wrzuca 2 list do „kosza”  
**q** – usuwa listy z „kosza” i zamyka program  
**x** – opuszczenie programu bez usuwania listów  
**–** – poprzedni list  
**+** – następny list  
**p** – powtórne wyświetlenie bieżącego listu  
**w 1 /home/zachowana\_tresc** – zachowanie treści listu nr 1 w pliku zachowany\_list w katalogu home  
**s 1 /home/zachowany\_list** – zachowanie listu (nagłówka i treści) nr 1

**6. SIEĆ**

**rlogin nazwa\_kompa** – rejestracja w inny systemie Unix  
**telnet nazwa\_kompa** – rejestracja w innych systemach np.: winNT  
**rcp nazwa\_kompa:plik\_zrodlowy plik\_docelowy** – program do przesyłania plików pomiędzy systemami Unix  
**ftp plik\_zrodlowy nazwa\_kompa:plik\_docelowy** – program do przesyłania plików pomiędzy dowolnymi systemami

KOMENDY FTP:

**get plik** – pobranie pliku z innego kompa  
**mget plik(i)** – pobranie wielu plików  
**put plik** – wysłanie pliku do innego kompa  
**mput plik(i)** – wysłanie wielu plików  
**bye** – zamkniecie połączenia z innym kompem  
**ls** – spis plików na drugim kompie  
**cd** – zmiana katalogu na drugim kompie  
**binary** – tryb przesłania plików binarnych  
**ascii** – tekstowy tryb przesłania (z tłumaczeniem znaków końca wiersz)

**control-]** – nie przerywając pracy na zdalnym kompie powrót do swojego systemu lokalnego (W PRZYPADKU SESJI TELNET)  
**return~control-z** – to co wyżej ale dla RCP (działa jeśli twoja powłoka ma wbudowany mechanizm wielozadaniowości zwany „job-control”

**7. PROCESY**

**ps** – wypisuje stan procesów  
**ps -l** – dokładniejsze wypisanie stanów procesów  
**ps -aux | more** – wypisanie włącznie z procesami nie należącymi do ciebie (| more – udogodnienie)

OPIS WYPISANYCH KOLUMN:

**PID** – identyfikator procesu  
**TTY** – terminal do którego jest podłączony proces  
**STAT** – stan procesu (W=czekający, R=wykonywany lub gotowy do wykonania S=uśpiony, Z=zabity ale wciąż przebywający w systemie)  
**TIME** – czas jego wykonywania  
**COMMAND** lub **CMD** – nazwa programu który wykonuje proces  
**UID** – nazwa użytkownika który jest właściciele procesu  
**PPID** – proces nadrzędny który utworzył dany proces  
**C** lub **CP** – wykorzystanie procesora – używane do szeregowania procesów  
**PRI** – priorytet  
**NI** – wartość używana do obliczania priorytetu  
**ADDR** – adres procesu  
**SZ** – rozmiar procesu w blokach  
**WCHAN** – zdarzenie na które oczekuje proces

**nice o\_ile\_nizszy polecenie**

– zmiana priorytetu procesu i mniejsza liczba tym wyższy priorytet (automatycznie niższy priorytet otrzymują procesy intensywnie wykorzystujące CPU) (tylko administrator może podawać liczby ujemne)

**cat \*.doc > wszystkie.doc &**

– znaczek & na końcu polecenia oznacza ze powłoka utworzy dany proces i uruchomi program ale nie będzie czekała na jego zakończenie

**kill identyfikator\_procesu**

– zakończenie procesu

**kill -9 identyfikator\_procesu**

– zakończenie procesów które ignorują zwykłą komendę kill

**kill -v nazwa\_procesu**

– zabija proces po podaniu jego nazwy

**at czas „polecenie” control-d**

– wykonanie procesu o określonej godzinie np.: „cat 0900a „cat \*.doc > razem.doc” control-d” co oznacza ze proces się uruchomi o godzinie 9:00 rano (a=a.m., p=p.m.)

**8. EDYTOR VI**

**vi nazwa\_pliku** – otwiera edytor z plikiem do edycji  
**vi** – uruchamia edytor potem:  
**:** – wierszowy tryb pracy (wiersz staje się poleceniem po naciśnięciu return)  
**i** – drugi tryb pracy gdzie znaki są wstawiane do bufora aż do naciśnięcia klawisza esc  
**10x** – usunie 10 znaków pod kursorem  
**10X** – usunie 10 znaków na lewo od kursora  
**:w** – zachowanie pliku na dysku  
**:q** – opuszczenie programu  
**:wq lub ZZ** – zapisuje plik i opuszcza program  
**:q!** – wyjście bez zapisania pliku  
**10dd** – wycina do schowka 10 wierszy od kursora w dół  
**10yy** – kopiuje do schowka 10 wierszy od kursora w dół  
**p** – wklejenie zawartości schowka na lewo od kursora  
**P** – wklejenie zawartości schowka na prawo od kursora  
**/klocek** – znajduje string klocek  
**/** – następne wyszukanie tego samego stringu  
**:s/klocek/badylek** – zamiana klocek na badylek w bieżącym wierszu  
**:1,$s/klocek/badylek/g** – zamiana klocek na badylek w całym pliku  
**:r nazwa\_pliku** – wczytanie do bufora innego pliku  
**:set number** – numerowanie wierszy  
**:set no number** – nie ponumerowane wiersze  
**u** – cofnięcie skutków ostatniego polecenia

**9. EDYTOR EMACS**

**control-h** – wypisanie pomocy emacs  
**na przykład: control-h a file** – wypisanie info o funkcji w której występuje słowo file  
**control-h o** – przełączanie miedzy oknami  
**control-h 0** – usunięcie okna pomocy gdy kursor się w nim znajduje  
**control-h 1** – usunięcie okna pomocy gdy kursor znajduje w oknie z redagowanym tekstem  
**control-x b (i podać nazwę pliku)** – wyświetlenie określonego redagowanego pliku  
**emacs nazwa\_pliku (w wierszu poleceń)** – otwarcie pliku w edytorze emacs  
**control-x control-s** – zapisanie pliku  
**control-x control-c** – opuszczenie emacsa  
**nazwa\_pliku~** – nazwa kopii zapasowej pliku  
**esc x overwrite-mode** – tryb zastępowania tekstu  
**control-k** – wycięcie do schowka tekstu od kursora do końca wiersza  
**control-@** – zaznaczenie początku przenoszonego bloku tekstu  
**control-w** – zaznaczenie końca bloku przenoszonego tekstu  
**control-y** – wklejenie zawartości schowka  
**control-s** – wyszukiwanie stringa  
**10. INNE INFORMACJE**

**1)** gdzie co jest „standardowo”:  
**– /** – katalog główny  
**– /root** – katalog administratora (root-a)  
**– /home** – katalogi użytkowników  
**– /usr/bin** – programy dla użytkowników, pliki wykonywalne (standartowe programy instalowane na początku z systemem)  
**– /usr/local/bin** – programy dodane przez administratora dla konkretnego hosta lub lokalnej sieci  
**– /usr/local** – oddziela pliki dostarczone przez „sprzedawcę” i późniejsze rozszerzenia od programów pochodzących z systemu  
**– /usr/lib** – pliki pomocnicze dla standartowych programów linuksowych  
**– /usr/man** – pliki podręcznika systemowego man  
**– /usr/src** – kody źródłowe programów  
**– /var/log** – pliki dziennika  
**– /var/spool** – katalogi koleiek dla drukarek, poczty itd.  
**– /bin lub /sbin** – standartowe programy będące częścią systemu, niezbędne do zapewnienia minimalnej funkcjonalności systemu  
**– /tmp** – pliki tymczasowe (usuwane przy starcie systemu)  
**– /var/tmp** – pliki tymczasowe (nie usuwane przy starcie systemu)  
**– /etc** – programy konfiguracyjne, nazwy pozostałych użytkowników (w pliku passwd)  
**– /dev** – jądro systemu, pliki odpowiadające urzadzeniom  
**2)** wykonywany program można zakończyć klawiszami: **control-c** lub zastopować klawiszami **control-** wtedy zostaje stworzony plik **core** w którym jest zapisany stan programu w momencie stopu (plik core może być duży)  
**3)** wypisywanie danych na ekran można zatrzymać: **control-s** i wznowić: **control-q**  
**4)** skrypt inicjujący: **.profile**  
**5)** przy tworzeniu plików z dziwnymi znaczkami w nazwie trzeba je poprzedzać lub umieścić nazwę w „” wtedy znaczki stracą swoje specjalne znaczenia ale wtedy trzeba uważać przy kasowaniu żeby np.: nie wpisać **rm \*** tylko **rm”\*”** bo można skasować wszystkie pliki z katalogu  
**6)** wyrejestrowanie się: **control-d** – powłoki bourne i korn, **logout** – powłoka c  
**7)** odświeżanie ekranu: **control-l**  
**8)** każdy plik ma dwóch właścicieli użytkownika i grupę  
– użytkownik może być członkiem więcej niż jednej grupy  
– nowo utworzony plik należy do tej grupy, do której należy katalog, w którym jest tworzony plik  
– gdy chcesz cos robić z plikiem system najpierw bada czy jesteś właścicielem pliku, jeśli tak to są sprawdzane prawa dostępu użytkownika, w przeciwnym wypadku jest porównywana grupa do której należy plik gdy odp. jest pozytywna system sprawdza grupowe prawa dostępu gdy nie, prawa dostępu pozostałych użytkowników  
**9)** po zmontowaniu np.: cdromu jest „tworzony” punkt zamontowania który nie musi mieć nic wspólnego z fizycznymi nośnikami, podrzędny system plików jest „doczepiany” do systemu głównego w punkcie zamontowania  
**10)** Kod źródłowy systemu Linux znajduje się w katalogu **/usr/src/linux-2.4** w tym katalogu jest tez katalog „kernel” gdzie jest plik „sched.c” dotyczy on zarządzania procesami  
**11)** Domyślna lokalizacja pakietów na płycie z instalka linuksa jest: **/mnt/RedHat/RPMS**  
**12)** Programem lpq możemy opróżnić bufor drukarki

**shutdown -r now lub reboot** – ponowne uruchomienie kompa  
**shutdown -h now lub halt** – zatrzymuje prace systemu  
**gnomecc** – control-panel dla gnome  
**uname -a** – informacje m.in. o wersji jądra którą mamy  
**df** – ilość wolnego miejsca na dysku  
**du** – wielkość katalogów i ich podkatalogów w kilobajtach  
**file ścieżka** – podaje rodzaj pliku

**11. MONTOWANIE CD-ROM, FLOPPY I PARTYCJI Z WINDOWS**  
PAMIĘTAJ ŻE ŚCIEŻKA: **/mnt/jakis\_katalog** MUSI ISTNIEĆ PRZED ZMONTOWANIEM CZYLI MUSISZ NAJPIERW STWORZYĆ KATALOG W **/mnt** DO KTÓREGO BĘDZIE PODMONTOWYWANY JAKIŚ SYSTEM PLIKÓW

**mount /dev/cdrom/ mnt/cdrom** – zamontowanie cdrom-u  
**umount /dev/cdrom** albo **umount /mnt/cdrom** – odmontowanie cdromu  
**mount /dev/fd0 /mnt/floppy** – zamontowanie flopa  
**umount /dev/fd0** albo **umount /mnt/floppy** – odmontowanie flopa  
**mount /dev/hda1 /mnt/windows** – montowanie dysku z windowsem (jeśli masz 2 dyski na pierwszym jest win a na drugim linux jeśli masz inaczej np.: win jest na tym samym dysku co linux może ścieżka wyglądać inaczej)  
**umount /dev/hda1 /mnt/windows** – odmontowywanie partycji z windowsem

MOŻESZ SKONFIGUROWAĆ LINUKSA TAK ABY MONTOWAŁ NP: PARTYCJĘ WINDOWSOWĄ ZARAZ NA STARCIE – MUSISZ ZNALEŹĆ TYLKO PLIK **/etc/fstab** OTWORZYC GO NP: PROGRAMEM **gedit** I NA SAMEJ GÓRZE WPISAĆ PO KOLEI:  
**/dev/hda1 /mnt/windows fat auto 0 0**  
OCZYWIŚCIE SCIEŻKA **/dev/hda1** MOŻE BYĆ INNA W TWOIM KOMPIE

**12. AUTOMATYKA, SKRYPTY W LINUKSIE**

**at** – można tym poleceniem zlecić systemowi wykonywanie jakiegoś polecenia o określonej godzinie ( jeśli ma być wykonywane cyklicznie można je dodać do pliku cron danego użytkownika)

(CHCEMY ABY O 16:05 BYŁ WYSYŁANY LIST NA OKREŚLONY ADRES) PO kolei:  
**at teatime + 5 minutes** – zaznaczmy o której ma to nastąpić (teatime oznacza 16:00) czas można tez określić np. tak: at 2:14pm Jan9, at now + 5 minutes (hour, week, day, year), at teatime next day  
**mail -s „temat wiadomości” dzony@poczta.onet.pl** – piszemy co ma być wykonywane i wciskamy enter  
**„co tam słychać”** – wpisujemy treść listu i wciskamy enter  
naciskamy **control-d** – i tym samym kończymy

**mail dzony@poczta.onet.pl <~/nazwa\_pliku** – wysyła na podany adres zawartość pliku  
**atq** – wyświetla wszystkie polecenia oczekujące na wykonanie  
**atrm numer** – usuwa polecenie o określonym numerze  
**cron** – automatyczne uruchamianie poleceń w określonych odstępach czasu (np.: przypomnienie co tydzień o zrobieniu kopii zapasowej)  
**/var/spool/cron** – pliki cron wszystkich użytkowników. Pliki cron można stworzyć w linux configuratorze

**history n > nazwa\_skryptu** – „tworzy” skrypt z ostatnich n komend  
**chmod u+x nazwa\_skryptu** – PAMIĘTAJ ŻEBY NADAWAĆ SKRYPTOM MOŻLIWOŚĆ WYKONYWANIA

TWORZENIE SKRYPTÓW POWŁOKI (PRZYKŁADOWE SPOSOBY):

1) SPOSÓB – Gnotepad  
Uruchom jakikolwiek edytor np. gnotepad  
Na samej górze wpisz (musi zawierać pełną ścieżkę do „programu” w którym będzie wykonywany):  
**#!/bin/bash**  
potem np.:  
**echo -e „dzwonek! a”**  
następnie zapisz plik np.: pod nazwą: **skrypt** aby uruchomić skrypt np.: wejdź do katalogu gdzie jest i wpisz:  
**bash skrypt**

2) SPOSÓB – vi  
Wpisz w linii komend:  
**vi nazwa\_skryptu (np.: skrypt)** naciśnij:  
**i**  
wciśnij:  
**control-c** wpisz:  
**:wq** i aby uruchomić skrypt wpisz:  
**bash skrypt** (ale jeśli jesteś w tym samym katalogu co skrypt)

**licznik = 0** – deklaracja zmiennej  
**nazwa = „Stefan Telefan”** – również deklaracja w tekście wystąpiła spacja wiec jest otoczony cudzysłowem  
**licznik = $mojazmienna** – przypisanie jednej zmiennej wartości drugiej (mojazmienna przejęła wartość)  
**$1, $2 itd.** – dzięki tym argumentom ich wartości można przekazywać po uruchomieniu skryptu z wiersza poleceń

**SKRYPT – WYŚWIETLANIE ARGUMENTÓW Z WIERSZA POLECEŃ**

if [$# -eq 0]

then

echo "Podaj identyfikator użytkownika"

else

echo "Witaj" $1

fi

$# – oznacza wszystkie argumenty które można podać z wiersza poleceń  
$# -eq 0 – porównanie (jeśli $# jest równe 0 )  
fi – zakończenie instrukcji warunkowej if

$# – oznacza wszystkie argumenty które można podać z wiersza poleceń  
$0 – nazwa programu powłoki  
$\* – pojedynczy tekst utworzony ze wszystkich argumentów przekazanych w wierszu poleceń  
= – równe (łańcuchy znaków)  
!= – nierówne (łańcuch znaków)  
-eq – równe (liczby)  
-ge – większe lub równe (liczby)  
-le – mniejsze lub równe (liczby)  
-ne – rożne (liczby)  
-gt – większe niż (liczby)  
-lt – mniejsze niż (liczby)  
for…do…done  
while…do…done  
until…do…done  
select…element…in…lista\_elementów…do…done  
if…elif…else…fi  
case

JĘZYK AWK

Poniższy skrypcik jest pisany w linii komend i ma wypisać ilość liter w każdym wierszu pliku z tekstem (język Awk najlepszy jest właśnie do przetwarzania tekstu)  
**cat nazwa\_pliku | gawk ‚{print NF „: ” $0}’**

JĘZYK PERL

Perl może być uruchamiany na prawie każdym systemie operacyjnym, służy m.in. do obsługi stron WWW za pomocą mechanizmów CGI lub np.: automatyzacji zadań administracyjnych.

**SKRYPT – WYŚWIETLA NUMER IDENTYFIKACYJNY UŻYTKOWNIKA**

Wszystko podobnie jak w tworzeniu skryptów powłoki tylko że tu wpisujemy po kolei w pliku:

#! /usr/bin/perl

print"Twój numer identyfikacyjny to $< n"

print "-------------------------------------- n"

Uruchom powyższy skrypt wpisując po prostu jego nazwę

$< – zmienna która zawiera numer identyfikacyjny użytkownika

JĘZYKI PYTHON I TCL

Python obsługuje klasy wyjątki i dynamiczne typy danych może pracować interaktywnie lub interpretując kod zapisany w modułach. Programy w tym języku mogą korzystać ze środowiska graficznego np.: Gnome.

TCL jest prostym językiem skryptowym przeznaczonym głownie do osadzania w innych językach, ma za zadanie spajać biblioteki i języki programowania o większych możliwościach.

JĘZYK C

PONIŻEJ OPISUJĘ CO TRZEBA ZROBIĆ ŻEBY STWORZYĆ PROGRAM W POJEDYNCZYM PLIKU (oczywiście są na to inne sposoby):  
**wpisujemy: vi progs.c** – otwiera nowy plik „progs.c” w edytorze vi  
naciskamy **i** – wchodzimy w tryb wpisywania programu  
wpisujemy np. taki program:

#include

int main()

{

printf("Pararara n");

exit(0);

}

po wpisaniu programu naciskamy **ctr-c** – dzięki temu wejdziemy w tryb wpisywania na terminalu  
wpisujemy: **:wq** – czyli zapisz i zamknij edytor  
wpisujemy: **touch program** – tworzymy pusty plik  
wpisujemy: **gcc progs.c -o program** – dzięki temu nasz program z pliku progs.c zostaje skompilowany do pliku wynikowego **program**  
wpisujemy: **./program** – uruchamia nasz program

**13. INSTALACJA PAKIETÓW – GnoRPM**

Wejdź do **Main Menu / System / GnoRPM**  
naciśnij przycisk **Install**  
naciśnij przycisk **Add**  
teraz (jeśli chcesz cos zainstalować z cdromu) wejdź do (nie koniecznie musisz mieć taka sama ścieżkę) **/mnt/cdrom/…** i znajdź pliki z rozszerzeniem **.rpm**, który chcesz zainstalować, jak już cos wybierzesz naciśnij **Add** i zamknij to okno  
na tym oknie co pozostało powinieneś widzieć ikonkę paczki i nazwę wybranego przez siebie pakietu teraz już po prostu naciśnij **Install**  
Jeśli chcesz uzyskać dokładniejsze info o pakiecie naciśnij na niego prawym klawiszem i wybierz **Query**, jeśli chcesz sprawdzić czy nie ma błędów naciśnij **Verify**. Możesz tez instalować pakiety z internetu – do tego służy przycisk **Web Find**.