T: Konfiguracja usługi SSH (SECURE SHELL) w systemie Linux.

Zadanie1:

Odszukaj w serwisie internetowym Wikipedii informacje na temat usługi SSH.

SSH (ang. secure shell) to standard protokołów komunikacyjnych używanych w sieciach komputerowych w architekturze klient-serwer. SSH służący do terminalowego łączenia się ze zdalnymi komputerami. Transfer wszelkich danych jest zaszyfrowany oraz możliwe jest rozpoznawanie użytkownika na wiele sposobów. Protokoły z rodziny SSH korzystają zwykle z portu 22 protokołu TCP.

Zadanie2:

Sprawdź dostępną w systemie pomoc na temat klienta ssh (`man ssh`).

W celu odszukania informacji o zainstalowanych programach usługi ftp należy uruchomić YaST i przejść do Zarządzania oprogramowaniem lub w konsoli tekstowej wydać polecenie:

```
rpm -qa | grep ssh
```

Do połączenia się ze zdalnym serwerem możemy wykorzystać klienta konsoli tekstowej wydając polecenie:

```
ssh serwer.domena.pl
ssh 127.0.0.1
ssh -l username servername
ssh username@servername -p 22
```

Połączenie możemy również realizować przy pomocy nakładki Midnight Commander wybierając z menu Lewy/Prawy => Połączenie po powłoce i wpisując:

```
/#sh:username@hostname/etc/sysconfig
username@hostname
```

Do pobierania plików poprzez usługę ssh możemy wykorzystać polecenie konsoli tekstowej:

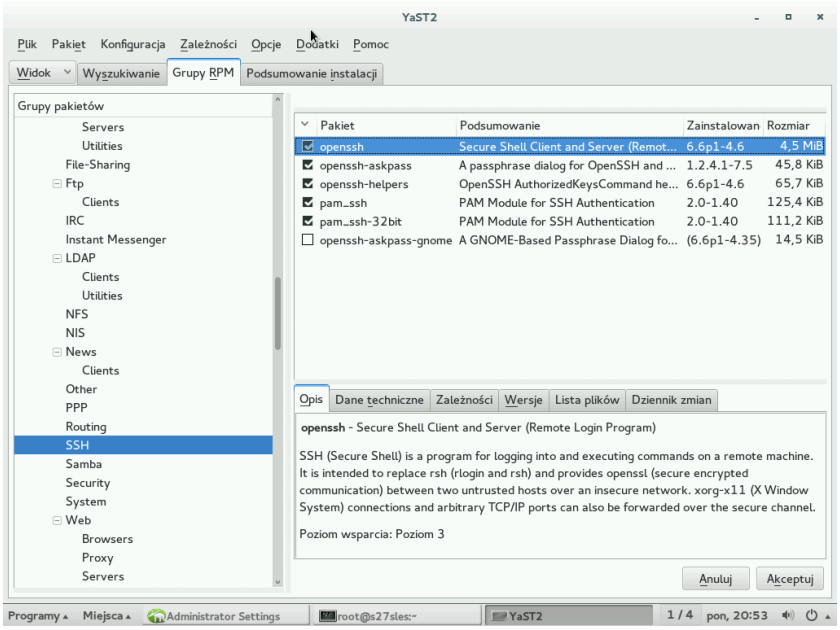
```
scp username@ip_server:/path/filename /local_path/filename
```

Do wysyłania plików poprzez usługę ssh możemy wykorzystać polecenie konsoli tekstowej:

```
scp /local_path/filename username@ip_server:/path/filename
```

Zadanie3:

Sprawdź przy użyciu konsoli tekstowej dostępność oprogramowania ssh w systemie Linux, a w razie potrzeby przeprowadź instalację pakietu, np. `openssh-server`. Uruchom centrum sterowania YaST w celu sprawdzenia dostępności oprogramowania serwera ssh.



W celu sprawdzenia w konsoli tekstowej, czy zainstalowane jest oprogramowanie ssh należy wydać polecenie:

```
rpm -qa | grep ssh
dpkg -l | grep ssh
```

Instalację usługi ssh w systemie Linux Ubuntu przeprowadzimy za pomocą konsoli tekstowej wydając polecenie:

```
apt-get update
apt-get install openssh-server
apt-get install openssh-client  #domyślnie jest zainstalwoany
```

Usługa po instalacji jest automatycznie uruchamiana na porcie 22 tcp. Stan portów w lokalnym komputerze sprawdzimy poleceniem:

```
netstat -ant | grep :22
```

Dodatkowe polecenia konsoli tekstowej:

```
chkconfig sshd on
iptables -L | grep ssh     #lub numer portu 22
man sshd
cat /etc/hosts.allow
cat /etc/hosts.deny
     sshd : all except s27nau
~/.ssh/
/etc/init.d/sshd status
/etc/init.d/sshd start
service ssh restart        #sshd
initctl reload-configuration    #wykonać, gdy występuje błąd Unknown job:ssh
```

Pliki konfiguracyjne serwera (`sshd_config`) i klienta (`ssh_config`) `ssh` znajdują się w katalogu `/etc/ssh`. W celu zapoznania się z dostępnymi opcjami konfiguracji serwera możemy w przeglądarce Konquerror w polu adresu wpisać `#sshd_config` lub w konsoli tekstowej podać polecenie `man sshd_config`.

Zadanie5:

Przeanalizuj zawartość pliku `/etc/ssh/sshd_config`.

Generowanie kluczy publicznego i prywatnego na potrzeby demona sshd:

```
ssh-keygen -t rsa   (ssh_host_rsa_key - nazwa pliku)
ssh-keygen -t dsa   (ssh_host_dsa_key - nazwa pliku)
ssh-keygen -t rsa1  (ssh_host_key - nazwa pliku)
```

Uzyskanie informacji na temat odcisku palca (fingerprint) klucza:

```
ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key
```

Sekwencja poleceń dokonujących podpisania klucza `user@poczta.pl` oraz wyświetlający informacje o odcisku (należy wydać w katalogu `/home/user/.ssh`):

```
gpg --edit-key user@poczta.pl
Polecenie> sign
Polecenie> check
Polecenie> save
Polecenie> quit
```

W celu zablokowania usługi ssh dla wszystkich komputerów za wyjątkiem komputera nauczyciela należy w pliku /etc/hosts.deny dokonać następującego wpisu:

```
sshd : all except s27nau
sshd : all except 192.168.19.35
sshd : lamer.pl
ssh : all
rsh : all
```

Przykładowe opcje konfiguracji serwera sshd w pliku /etc/ssh/sshd_config:

```
# $OpenBSD: sshd_config,v 1.56 2002/06/20 23:37:12 markus Exp $
########################################
# Jest to plik konfiguracyjny serwera sshd.
# Przegladnij sshd_config(5) dla uzyskania dodatkowych informacji.
# Ten sshd zostal skompilowany ze sciezka:
# PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin
#
# UWAGA: po zainstalowaniu sshd popraw plik /etc/hosts.allow
# (sshd: IPklienta oraz w nowym wierszu ssh: IPklienta)
# Pamietaj o potrzebie dopuszczenia sshd w firewallu
########################################
# Nr portu na ktorym nasluchuje Twoj serwer (demon sshd)
Port 22
# Dostepne protokoly: ssh2 i ssh1
Protocol 2,1
# Na jakim IP Twojego serwera będzie nasluchiwac sshd
# (wazne w przypadku kilku kart sieciowych na serwerze)
# TU nasluchuje na wszystkich dostępnych adresach serwera
# czyli wszystkich kartach sieciowych i ew. modemie.
ListenAddress 0.0.0.0
#ListenAddress ::
#
# Klucz HostKey dla protokolu version 1
HostKey /etc/ssh/ssh_host_key
# Klucz HostKeys dla protokolu version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
#
# Dlugosc zycia (w sekundach) klucza "version 1 server key"
KeyRegenerationInterval 3600
# Dlugosc klucza (w bitach) klucza "version 1 serwer key"
ServerKeyBits 768

# Logging
# obsoletes (przestarzaly) QuietMode (cichy tryb)
# and FascistLogging
# SyslogFacility czyli SyslogUdogodnienie
#SyslogFacility AUTH
#LogLevel INFO
#
#################################
# Parametry autentykacji. (Authentication)
#################################
#
# Czas oczekiwania (w sekundach)
LoginGraceTime 600
# Czy mozna logowac sie zdalnie na konto roota.
# Wpisz "no" i jak root loguj sie poprzez konto
# zwyklego uzytkownika i komendy su lub su -l
PermitRootLogin no
StrictModes yes
#################################
# Czy zgadzasz sie na autentykacje RSA ? (TAK!!!)
# Klucz RSA mozna uzyc zamiast lub rownoczesnie z haslem.
# Zaraz wybierzesz wlasciwe opcje.
# Teraz chwila wyjasnien: Nalezy odroznic ssh (narzedzie klienckie)
# i demona sshd (czyli serwer).
# Tutaj konfigurujemy co prawda sshd, ale pamietac nalezy, ze odlegly
# klient tez musi prawidlowo sie przygotowac.
# Do poprawnej pracy OPENSSH musimy dokonac konfiguracji demona
# sshd oraz klienta ssh.
# Jezeli zdecydowalismy sie na uzywanie ssh z kluczami RSA (zamiast
# lub rownoczenie z haslami), kazdy user (czyli Ty oraz twoi kumple)
# przed uzyciem powinien wygenerowac swoja wlasna pare kluczy komenda:
# $ ssh-keygen (w Mandrake może byc zrobiony automatycznie).
# W takiej chwili będziesz musial podac tzw. paszport (zapisz sobie na kartce)
# W podfolderze /home/antek/.ssh zostana utworzone dwa pliki:
```

```
# Identity - ktory zawiera PRYWATNY KLUCZ i nie powinien byc udostepniany
# nikomu (pamiętaj o restrykcyjnych prawach do tego pliku)
# identity.pub - czyli klucz publiczny.
# Jezeli zamierzamy jako klient pracowac przez ssh na odleglym serwerze,
# to zawartosc tego klucza powinnismy skopiowac do katalogu odleglego
# uzytkownika, nza ktorego konto logujemy sie - do pliku np.
# /home/antek/.ssh/ w pliku authorized_keys.
# Oczywiscie wowczas proby logowania na odlegly serwer musza
# byc podejmowane na konto antek i uwaga: wskazane jest na poczatku
# aby logowac sie z konta antek na konto tez antek.
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
#
# Autentykacja rhosts (czyli dopuszczanie do logowania wg listy tzw.
# "zaufanych" maszyn zamiast zmuszania ich do podawania hasel)
# nie powinna byc uzywana ze wzgledow bezpieczenstwa.
# Wystarczy bowiem, ze ktos sie wlamie i dobierze do plikow
# ~/.rhosts .shorts...
# JEZELI WYBRALES "no" (a to polecam) TO MOŻESZ ZAHASZOWAC KILKA
# KOLEJNYCH OPCJI, gdyz nie maja one w takim razie znaczenia.
# Wszak zrezygnowales z tego rodzaju autentykacji.
RhostsAuthentication no
#
# Ignorowanie plikow ~/.rhosts i ~/.shosts, ktore maja wykaz "zaufanych"
# stacji, z ktorych odlegly uzytkownik może sie zalogowac bez podania hasla.
# Oczywiscie wpisz "yes" i nie pozwol (ze wzgledu bezpieczenstwa) na
# czytanie plikow ~/.rhosts i ~/.shosts w zastepstwie autoryzowania haslem.
IgnoreRhosts yes
#
# Jezeli w autentykacji rhosts wybrales opcje "RhostsAuthentication yes"
# to możesz zmusic klientow zapisanych w w/w plikach ~/.rhosts i ~/.shosts by
# dodatkowo legitymowaly sie kluczem "host keys" znajdujacym
# sie w /etc/ssh/ssh_known_hosts (pamietasz jak opisalem koniecznosc
# skonfigurowania ssh u odleglego klienta poprzez wygenerowanie
# kluczy komenda: [ $ ssh-keygenktory ]) ?
#RhostsRSAAuthentication no
# i podobnie j.w. ale wobec protokolu version 2
#HostbasedAuthentication no
#
# Zmien ponizsze na NIE (!!!) jezeli nie masz zaufania do klienckich
# kluczy zapisanych w ~/.ssh/known_hosts (w autentykacji
# RhostsRSAAuthentication i HostbasedAuthentication).
# Czy serwer ma ignorowac nadzor nad komputerami w/w uzytkownikow?
# Inaczej mowiac - czy serwer ma zadowolic sie tescia kluczy od
# klientow zapisanych w ~/.ssh/known_hosts ?
IgnoreUserKnownHosts no
#
# Aby wylaczyc tunelowanie (co jest fatalnym pomyslem) - wyczysc wpisy
# hasel i zmien na "no"
PasswordAuthentication yes
# Zezwolenie na puste hasla
PermitEmptyPasswords no
#
# NIE WIEM O CO TUTAJ CHODZI.
# Uncomment to disable s/key passwords czyli
# Wyhaszuj po to, aby wylaczyc s/klucz hasel
#ChallengeResponseAuthentication no
#
# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#AFSTokenPassing no
# Kerberos TGT Passing only works with the AFS kaserver
#KerberosTgtPassing no
#
# Set this to 'yes' to enable PAM keyboard-interactive authentication
```

```
# Warning: enabling this may bypass the setting of 'PasswordAuthentication'
#PAMAuthenticationViaKbdInt yes
#
# Szyfrowanie przez ssh polaczenia graficznego za pomoca X-Window
X11Forwarding yes
X11DisplayOffset 10
X11UseLocalhost yes
#
# Pojawi sie komunikat powitalny o tresci pobranej z pliku /etc/motd
PrintMotd yes
#
# Pojawi sie komunikat powitalny o tresci z pliku tekstowego /etc/issue
#Banner /etc/issue
# Polecam jednak utworzyc inny plik np. /etc/ssh/banner z jakims tekstem
# Oczywiscie nalezy wowczas wpisac ponizej odpowiednia sciezke dostepu
Banner /etc/ssh/baner
#
# Pojawi sie informacja z data ostatniego logowania
PrintLastLog yes
#
KeepAlive yes
#
#UseLogin no
#
#Wiele kodów w OpenSSH które działały wyłącznie pod rootem, obecnie funkcjonują pod
nieuprzywilejowanym użytkownikiem. Ponieważ znacząco podnosi to bezpieczeństwo
OpenSSH, powinno się udostępnić cechę UsePrivilegeSeparation . Niestety, opcja ta
nie działa zbyt dobrze (w wersji OpenSSH 3.4p1) z innymi systemami unixowymi, można
jednak się spodziewać, że następna wersja OpenSSH będzie pozbawiona błędów
UsePrivilegeSeparation yes
#
# Zezwolenie na kompresje danych podczas połączenia
Compression yes
#
MaxStartups 10
#
# Sprawdzanie zgodnosci nazwy odległego klienta
# (pełnej domeny) z IP. Domyślnie niedostępne.
#VerifyReverseMapping no
#ReverseMappingCheck yes
#
#CheckMail yes
#
#UseLogin no
#
# Zezwolenie na szyfrowane polaczenie sftp.
Subsystem sftp /usr/lib/ssh/sftp-server
```

Konfiguracja klienta SSH zapisana jest w pliku /etc/ssh/ssh_config. Opcje konfiguracyjne sprawdzane są w następującej kolejności:

- opcje podane w linii komend,
- plik konfiguracyjny użytkownika ($HOME/.ssh/config),
- plik ogólnosystemowy.

Opis ustawień w pliku konfiguracyjnym klienta ssh (/etc/ssh/ssh_config)

`Hosts *` - otwiera sekcję dotyczącą połączeń do danego hosta - * oznacza wszystkie hosty,

`ForwardAgent yes` – określa, czy agent autentykacyjny ma być przekazywany na kolejne systemy na które następuje logowanie,

`ForwardX11 yes` – zezwala na automatyczne przekazywanie połączeń X11 ponad szyfrowanym kanałem SSH,

`RhostsAuthentication no`

`RhostsRSAAuthentication no` - zezwalanie na autentykację za pomocą mechanizmu rhosts,

`PasswordAuthentication yes` - autentykacja za pomocą haseł,

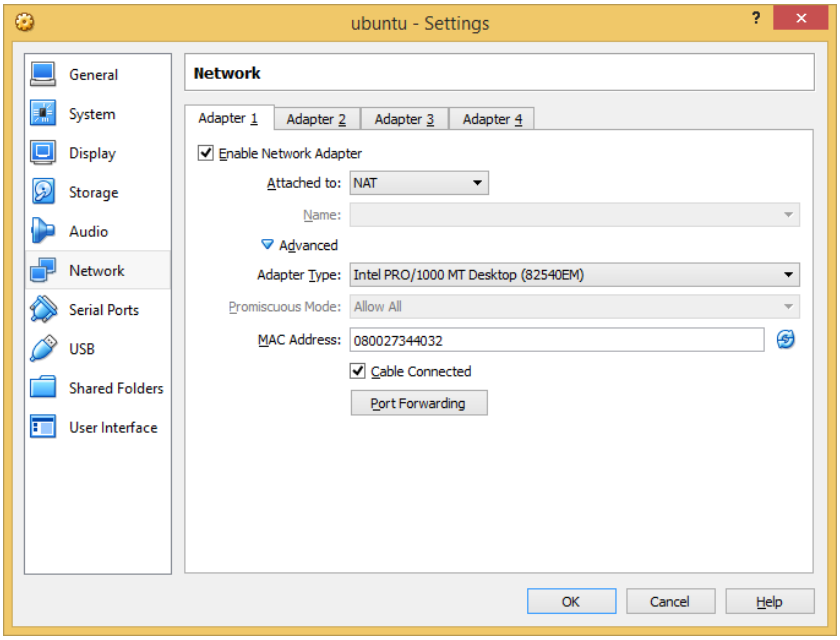
`RSAAuthentication yes`

`TISAuthentication no` - wybór metody autentykacji (wybrać tylko RSA),
`PasswordPromptHost yes`
`PasswordPromptLogin yes` - czy program ma pytać o hasła,
`FallBackToRsh no`
`UseRsh no` - możliwość użycia rsh w przypadku niepowodzenia połączenia za pomocą ssh. Można włączać, ale administrator zdalnej maszyny prawie na pewno to wyłączył,
`BatchMode no` - możliwość użycia ssh w trybie wsadowym,
`EscapeChar ~` - jaki znak powoduje wyjście z połączenia (jak w telnecie **ctrl+]**),
`Cipher 3DES` - algorytm stosowany do szyfrowania przy połączeniu ze zdalną maszyną,
`Compression yes` - czy włączona jest kompresja,
`CompressionLevel 9` - poziom kompresji, 0 - wyłącza,
`IdentityFile ~/.ssh/identity` - położenie i nazwa pliku identyfikacji.

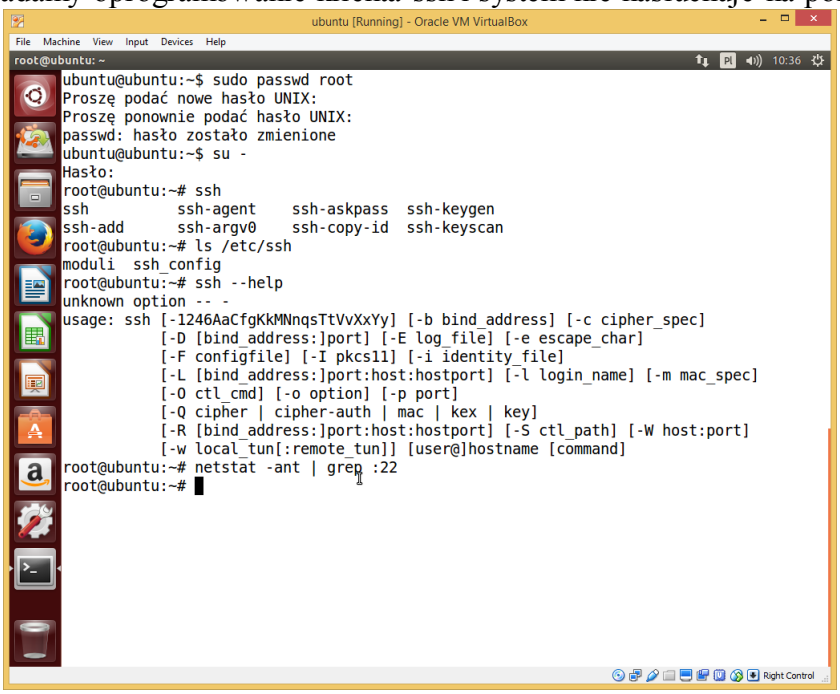
Zadanie6:

Wykorzystując dowolny procesor tekstu wykonaj sprawozdanie na temat instalacji, konfiguracji i sprawdzenia poprawności działania usługi SSH w systemie Linux Ubuntu. Na jednej stronie o rozmiarze A4, orientacji pionowej, umieść dwa opisane zrzuty z ekranu, o szerokości minimum 14 cm. Każdą stronę podpisz swoim imieniem i nazwiskiem w nagłówku strony a w stopce oznacz numer strony wg schematu Strona X z Y. Pracę zachowaj w pliku pod nazwą **$nazwisko_ssh** i prześlij pocztą elektroniczną do nauczyciela na adres greszata@zs9elektronik.pl.

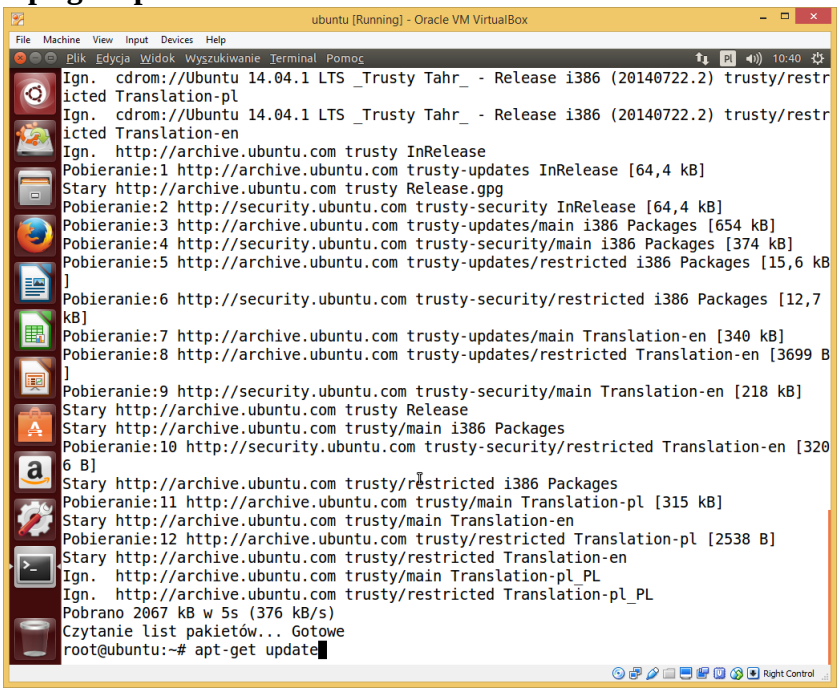
Przed przystąpieniem do instalacji usługi SSH należy sprawdzić, czy w maszynie wirtualnej systemu Linux włączyliśmy kartę sieciową i odpowiednio ją skonfigurowaliśmy. I tak dla maszyny wirtualnej **ubuntu** wybieramy ustawienia (**Settings**), przechodzimy do kategorii **Network** i w zakładce **Adapter 1** kartę podłączamy w trybie **NAT**.



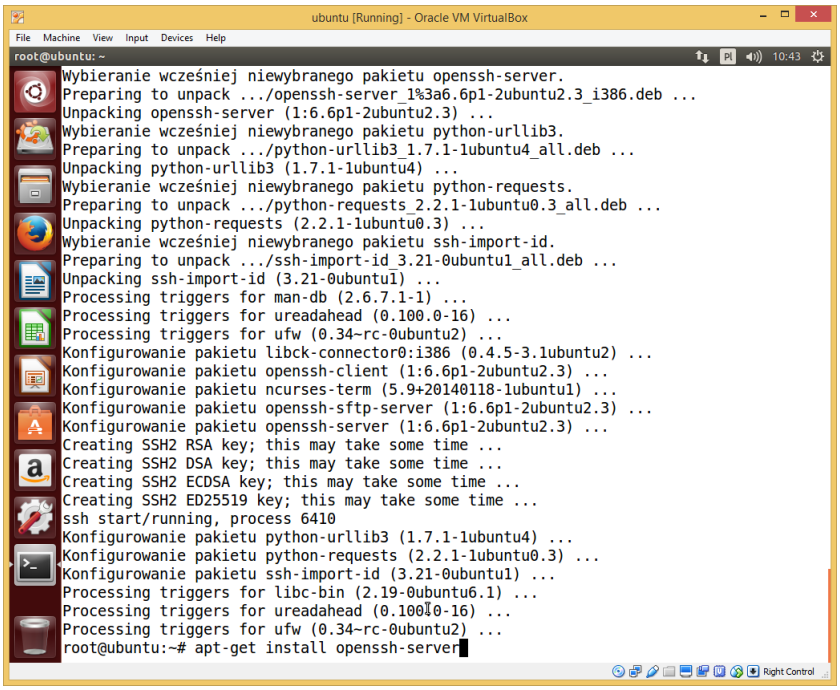
W systemie operacyjnym Linux Ubuntu uruchamiamy konsolę terminala. Domyślnie w systemie Linux Live Ubuntu zainstalowane jest oprogramowanie klienta ssh (openssh-client). Uprawnienia do zarządzania systemem posiada administrator systemu, dlatego należy w oknie podać hasło użytkownika root. W oknie terminala upewniamy się, że posiadamy oprogramowanie klienta ssh i system nie nasłuchuje na porcie 22 protokołu TCP:



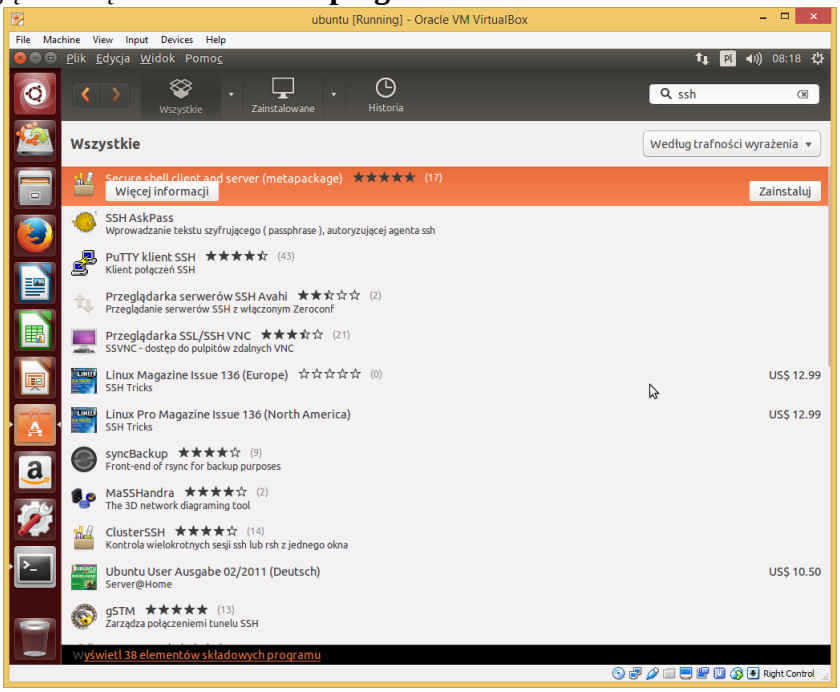
Przed instalacją niezbędnego oprogramowania aktualizujemy adresy serwerów źródłowych i listy dostępnych programów poleceniem **apt-get update**:



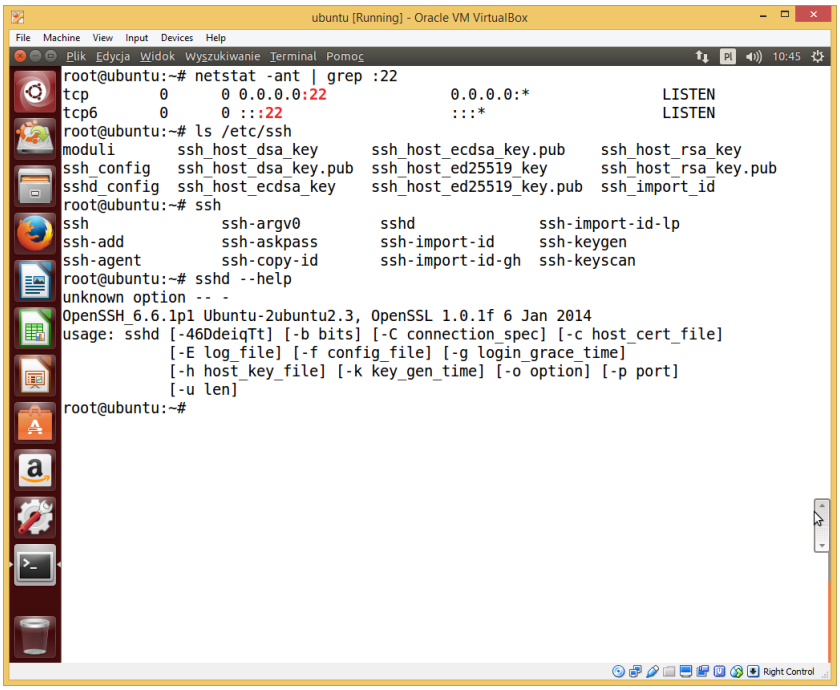
Następnie przeprowadzamy instalację oprogramowania serwera ssh wydając w konsoli polecenie **apt-get install openssh-server**:



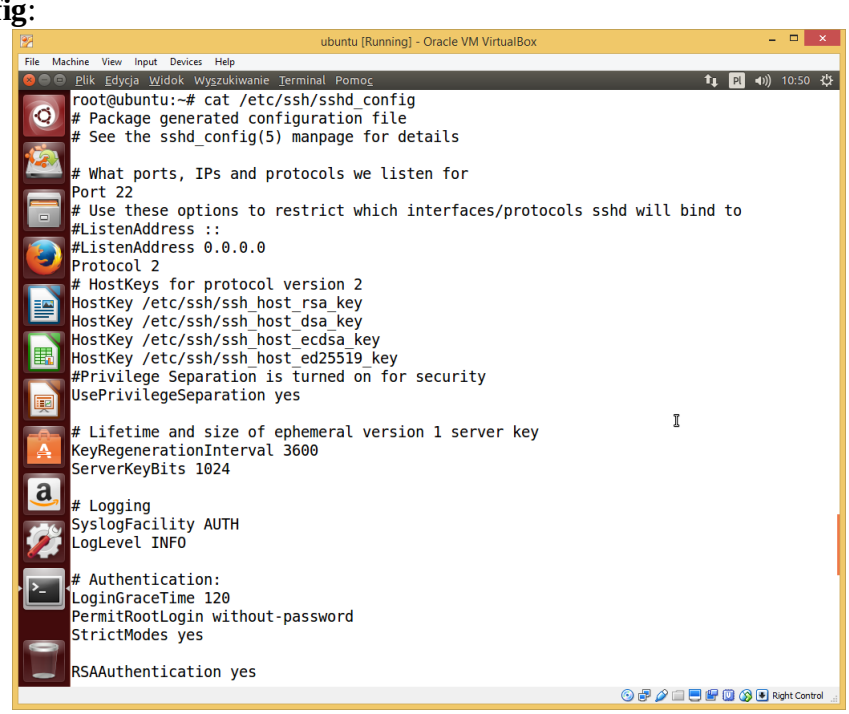
W systemie Linux Ubuntu możemy dokonać instalacji niezbędnego oprogramowania również w trybie graficznym wykorzystując narzędzie **Centrum oprogramowania Ubuntu**:



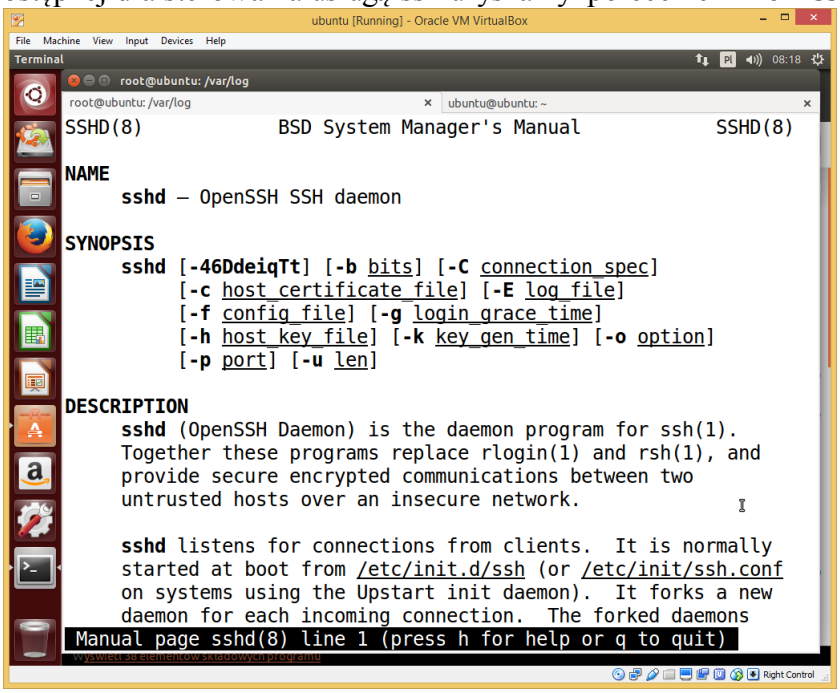
Po instalacji serwera ssh sprawdzamy bieżący stan usługi ssh (service sshd status) oraz zmiany programowe dokonane w systemie:



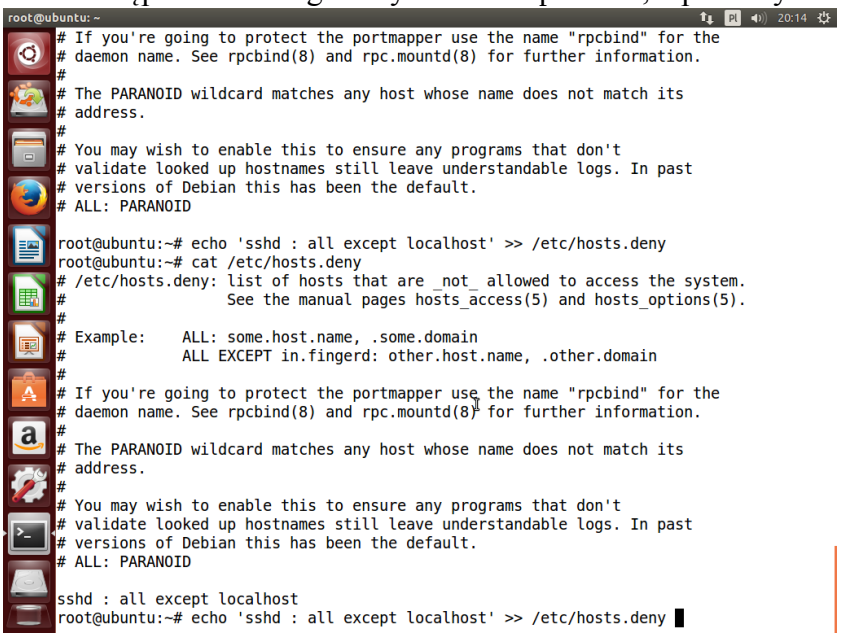
Sprawdzenie zawartości pliku konfiguracyjnego usługi ssh dokonamy wydając polecenie **cat /etc/ssh/sshd_config**:



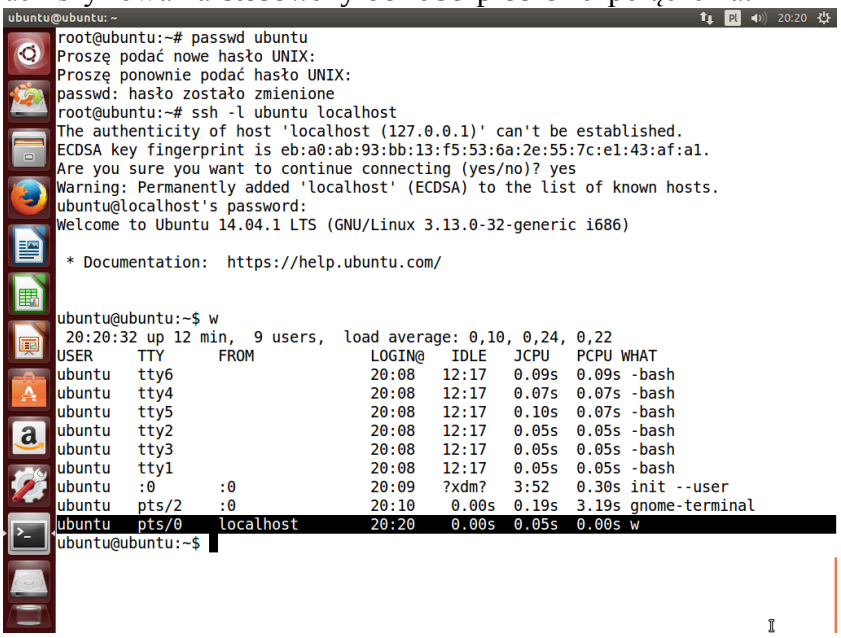
Sprawdzenie pomocy dostępnej dla sterowania usługą ssh uzyskamy poleceniem **man sshd**:



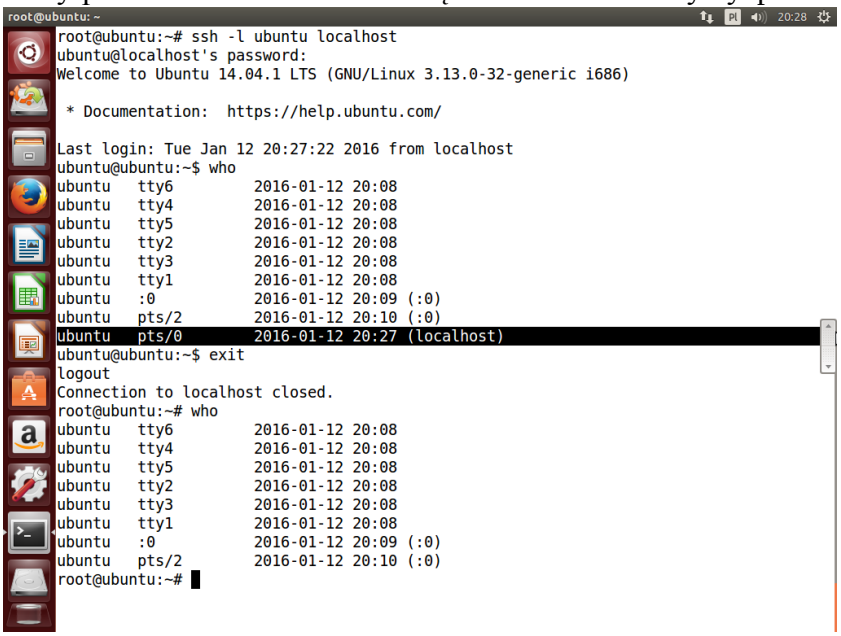
Restartujemy usługę ssh wydając polecenie **service ssh restart**. Dostęp do usługi możemy kontrolować poprzez pliki **/etc/hosts.allow** oraz **/etc/hosts.deny**. Dodając wpis **sshd : all except localhost** do pliku **/etc/hosts.deny** spowodujemy zablokowanie dostępowi do usługi wszystkim komputerom, oprócz systemu lokalnego:



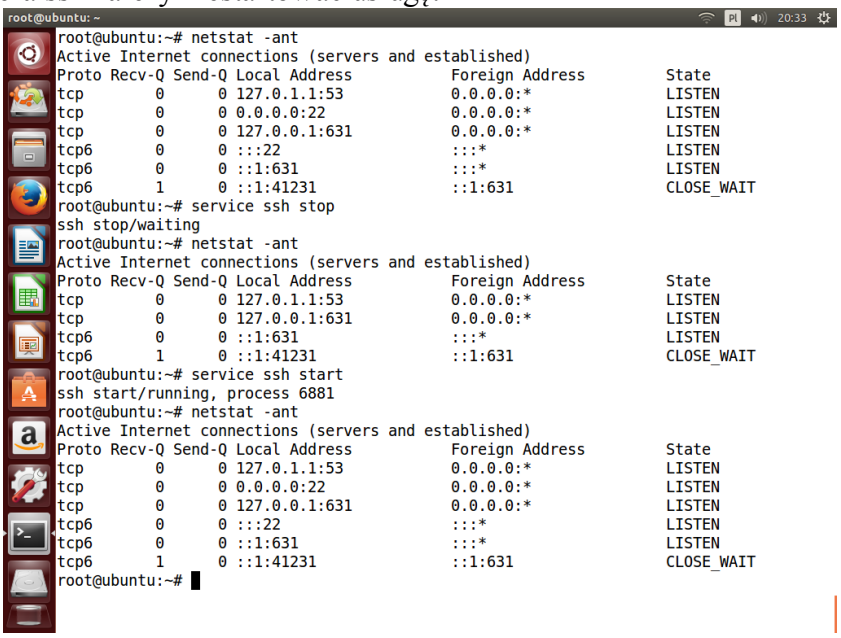
Połączenie z usługą nawiążemy podając polecenie **ssh -l ubuntu localhost**. Podczas pierwszego logowania należy zaakceptować klucz szyfrowania stosowany do zabezpieczenia połączenia:



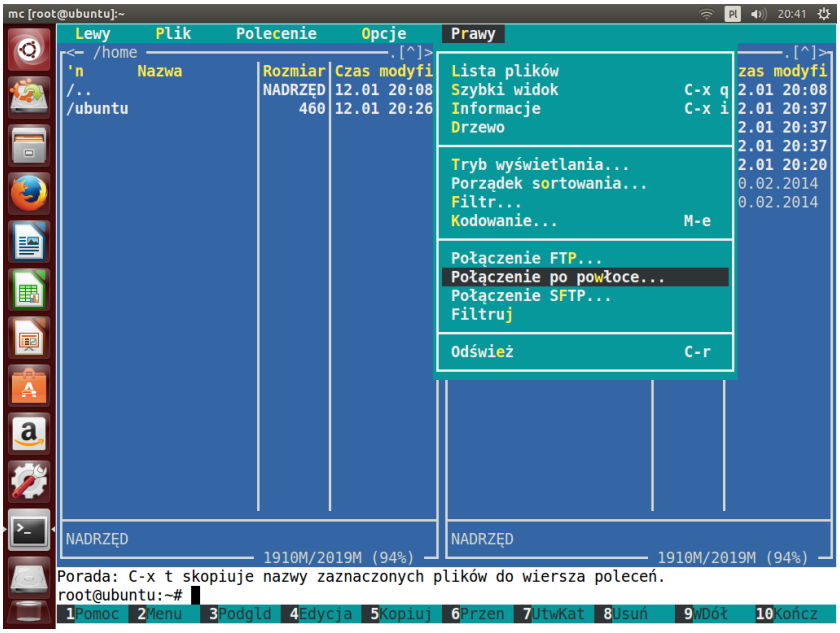
W trakcie połączenia z serwerem ssh użytkownik będzie wyświetlony na liście zalogowanych użytkowników w systemie. Listę wywołamy poleceniem **w** lub **who**. Połączenie ssh zakończymy poleceniem **exit**:



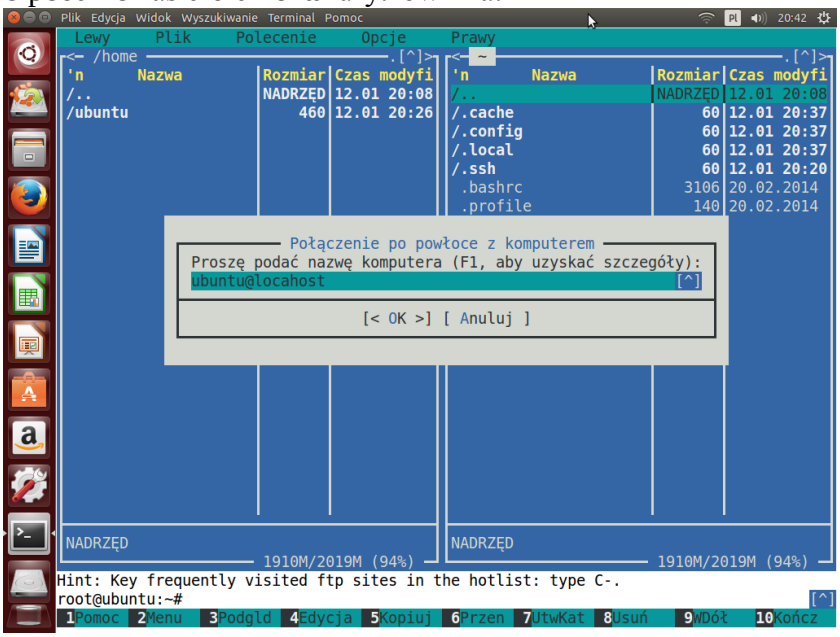
W dowolnym momencie pracy systemu możemy sterować usługą ssh. Wykorzystujemy do tego polecenia **service ssh stop** lub **service ssh start** lub **service ssh restart**. Pamiętajmy, że w przypadku zmian w zawartości pliku konfiguracyjnego serwera ssh należy zrestartować usługę:



Z usługą ssh możemy również połączyć się poprzez menedżera plików Midnight Commander. Po uruchomieniu programu poleceniem **mc** wybieramy z menu górnego (**F9**) w kategorii **Lewy**/**Prawy** polecenie **Połączenie po powłoce…**:



W wyświetlonym oknie podajemy nazwę użytkownika i adres serwera ssh i po kliknięciu przycisku **OK** zostaniemy poproszeni o podanie hasła dla konta użytkownika:



Pozostaje już tylko korzystać z ułatwień gwarantowanych przez menedżer plików MC: