

№4 Лабораторная работа

Серверді қашықтан басқару. SSH протоколы

Secure SHell-UNIX / Linux жұмыс істейтін желілік компьютерлерді қашықтан басқарудың негізгі құралы. Көптеген хостинг провайдерлері сайт иелеріне SSH протоколы арқылы серверлерге қол жеткізуге мүмкіндік береді.

Жұмыстың мақсаты: SSH протоколы арқылы қашықтағы хостпен жұмыс істеудің бастапқы дағдыларын алу.

Жұмысқа тапсырмалар

1. SSH пәрменін пайдаланып, сервер қабығын жүктеместен амалдық жүйенің түрін, аппараттық платформаны және сервер ядросының нұсқасын (тиісті параметрлері бар uname пәрмені) анықтаңыз.
2. SSH клиентін пайдаланып, edu серверіне қосылыңыз.asoіu студенттік есептік жазбамен.
3. Қашықтағы серверде қай каталог ағымдағы екенін анықтаңыз. Егер ол /home/student-тен өзгеше болса, онда /home/student-ке ауысыңыз.
4. Серверге қандай пайдаланушылар, қандай терминалдар және қандай ір мекенжайлары қосылғанын анықтаңыз (кім командасы)
5. Ағымдағы каталогта каталог жасаңыз сіздің атыңыз транслитпен (мысалы, ivanov).
6. Мс файл менеджерін іске қосыңыз, ағымдағы каталогтың мазмұнын қараңыз.
7. Мс файл менеджерімен жұмысты аяқтаңыз
8. Ssh сеансын аяқтаңыз.
9. Ssh сеансына кірмей-ақ, серверде бұрын жасалған каталогқа ерікті файлды (SCP пәрмені) жүктеңіз.
10. Ssh сеансына кірмей-ақ, /home/student сервер каталогының мазмұнын қараңыз .
11. Осы жұмыстың орындалуы туралы есеп жасаңыз. SSH көмегімен есеп файлын осы жұмысты орындау барысында жасаған сервер каталогына жүктеңіз.Методические указания

Серверді қашықтан басқару құралдары жергілікті және ғаламдық желілерде кеңінен қолданылады. Мұндай қаражаттың негізгі мақсаты - серверді басқару командаларын беру арнасын ұйымдастыру (жалпы жағдайда-шлюз арқылы) және осы командалардың нәтижесін клиентке қайтару. Сондықтан қашықтан басқару бағдарламаларына қойылатын негізгі талаптардың бірі-пайдаланушының ашықтығы және аз трафик. Бұл аумақтық бөлінген түйіндерді бір жұмыс орнынан орталықтан басқаруға немесе баяу байланыс желілері арқылы қашықтағы терминал клиенттеріне кіруді ұсынуға мүмкіндік береді..

Желілік шлюздер

Жергілікті желілерді Интернетке қосу әдетте байланыс сервері - шлюз (gateway) арқылы жүзеге асырылады. Мұндай сервер әдетте UNIX-тің кейбір нұсқаларында жұмыс істейді (FreeBSD, Debian GNU/Linux, RHEL және т.б.) және кем дегенде 2 желілік интерфейсі бар: біреуі ішкі, оған жергілікті желі қосылған, екіншісі Интернетке қарайтын сыртқы. Шлюзде маршруттау орнатылған, Nat және прокси қызметтері көтерілген.

Қосымша қызметтер (баннерді кесу, трафикті есепке алу және т.б.) көтерілуі мүмкін. С.Лазаренконың "Debian-да интернет шлюзін Орнату және конфигурациялау" нұсқаулығында iptables (firvol+Nat) және SQUID (прокси) көмегімен Debian GNU/Linux (Lenny және Squeeze) негізіндегі шлюзді орналастыру процедурасы егжей-тегжейлі сипатталған.

Басқару үшін символдық протоколдар (telnet, rlogin, ssh) және графикалық мүмкіндіктерді қолдайтын екілік (әр түрлі) қолданылады. Арнайы серверлер үшін графикалық құралдар әдетте пайдаланылмайды, өйткені мұндай серверлер оларды жұмыс станциялары ретінде пайдалануды білдірмейді. Бұл ресурстарды бөлудің қажеті жоқ дегенді білдіреді (маңызды!) графикалық пайдаланушы интерфейсі үшін.

Telnet және rlogin мәтіндік протоколдары қарапайым және функционалды, бірақ қауіпсіз емес. Ағымдағы жұмыста көрсетілген ойларға сүйене отырып, SSH протоколы бойынша UNIX-серверді қашықтан басқаруды игеру көзделеді. Айта кету керек, SSH протоколы графикалық режиммен жұмыс істеуді қолдайды (X серверінің туннелі). Сонымен қатар, SSH протоколы TCP протоколын көлік ретінде пайдаланатын кез келген желілік трафикті туннельдеуге мүмкіндік береді. (Толығырақ-the Secure Shell (SSH) Protocol Assigned Numbers, RFC 4250, 2006; the Secure Shell (SSH) Protocol Architecture, RFC 4251, 2006; The Secure Shell (SSH) Authentication Protocol, RFC 4252, 2006 және т. б.)

SSH протоколы бойынша серверді басқару үшін оның сервер қолдауы және клиенттік SSH қосымшасы қажет. UNIX серверлері SSH протоколын стандартты түрде қолдайды. Клиент ретінде әдетте OpenSSH қолданылады (SSH командасы шақырады). Windows үшін әртүрлі өндірушілердің

клиенттері бар, ең танымал PuTTY және SecureCRT. Зертханалық жұмыс OpenSSH клиентін пайдалануды көздейді. Бұл бағдарламаны пайдалану туралы мәліметтер man нұсқаулығында (man 1 ssh) қол жетімді.

Linux файлдық жүйесінің кейбір командалары "ашық бағдарламалық жасақтама"пәні бойынша №3 зертханалық жұмыста қарастырылған

Серверге қосылу

Серверге қосылу үшін SSH пәрменін сервердің пайдаланушы аты мен атауын немесе желілік мекен-жайын параметрлер ретінде көрсету керек:

```
aag@stilo:~> ssh -l student edu.asoiu // -l student - логин; edu.asoiu - ssh-сервер
```

Немесе солай ма:

```
aag@stilo:~> ssh student@edu.asoiu
```

Егер Сіз SSH-ді пайдаланушы атын көрсетпестен орындасаңыз, серверге ағымдағы Жергілікті Пайдаланушының аты жіберіледі.

Қашықтағы құрылғыға бірінші SSH қосылғанда, сіз осындай хабарды көресіз:

```
The authenticity of host 'edu.asoiu' can't be established.  
DSA key fingerprint is 94:68:3a:3a:bc:f3:9a:9b:01:5d:b3:07:38:e2:11:0c.  
Are you sure you want to continue connecting (yes/no)?
```

Жалғастыру үшін *yes* деп теріңіз. Бұл жағдайда сервер сіздің белгілі серверлер тізіміне қосылады, бұл туралы келесі хабарлама айтады:

```
Warning: Permanently added 'edu.asoiu' (DSA) to the list of known hosts.
```

Содан кейін қашықтағы компьютердің құпия сөзін енгізуге шақыру шығады. Серверде авторизациядан кейін пайдаланушы UNIX қабығына түседі (әдетте бұл shell нұсқаларының бірі, сервер параметрлеріне байланысты) және жергілікті машинада жұмыс істегендей командаларды енгізе алады.

Командаларды орындау

Пайдаланушы үшін қашықтағы серверде командаларды орындау жергілікті командалармен және файлдармен жұмыс істеуден айтарлықтай ерекшеленбейді. Бірнеше мысал келтірейік:

1. Ағымдағы каталогтың анықтамасы:

```
aag@stilo:~> ssh student@edu.asoiu  
Password: // пароль при вводе не отображается  
Last login: Thu Feb 21 11:08:13 2008  
Have a lot of fun... // авторизация прошла успешно  
student@stilo:~> pwd // ввод команды  
/home/student // вывод результатов выполнения  
student@stilo:~> // приглашение оболочки
```

2. Каталог файлдарының тізімін қарау:

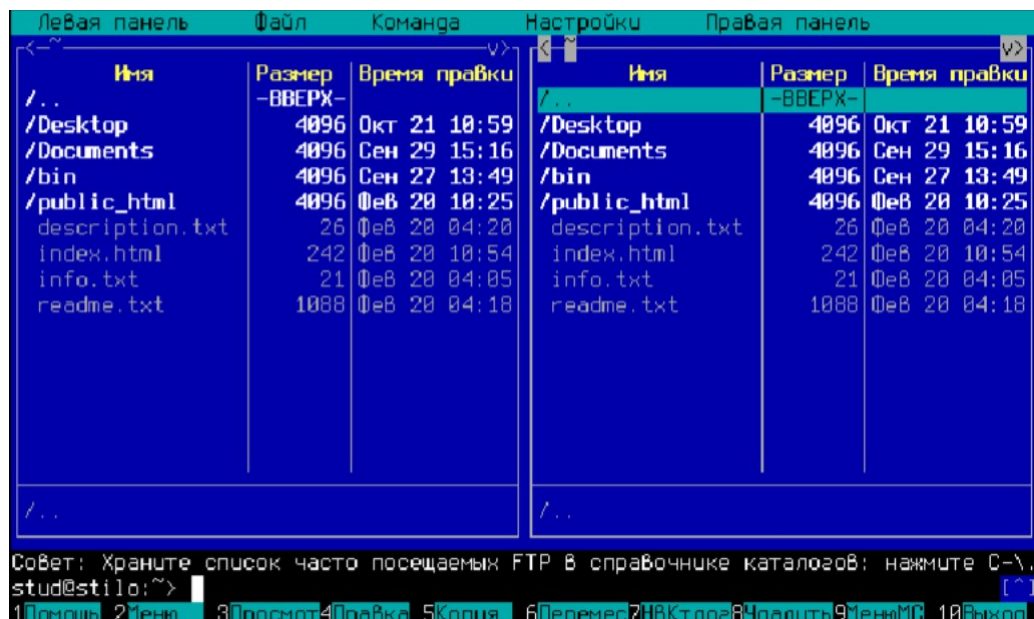
```
student@stilo:~> ls -  
ABl итогo 212  
-rw----- 1 student users 782 Фев 20 12:33 .bash_history  
-rw-r--r-- 1 student users 1177 Сен 27 13:49 .bashrc  
drwx----- 7 student nobody 4096 Окт 21 11:09 .beagle  
drwxr-xr-x 2 student users 4096 Сен 27 13:49 bin  
drwx----- 2 student nobody 4096 Окт 21 10:59 .config  
-rw-r--r-- 1 student nobody 26 Фев 20 04:20 description.txt  
drwxr-xr-x 2 student nobody 4096 Окт 21 10:59 Desktop  
-rw----- 1 student nobody 24 Окт 21 10:59 .dmrc
```

```

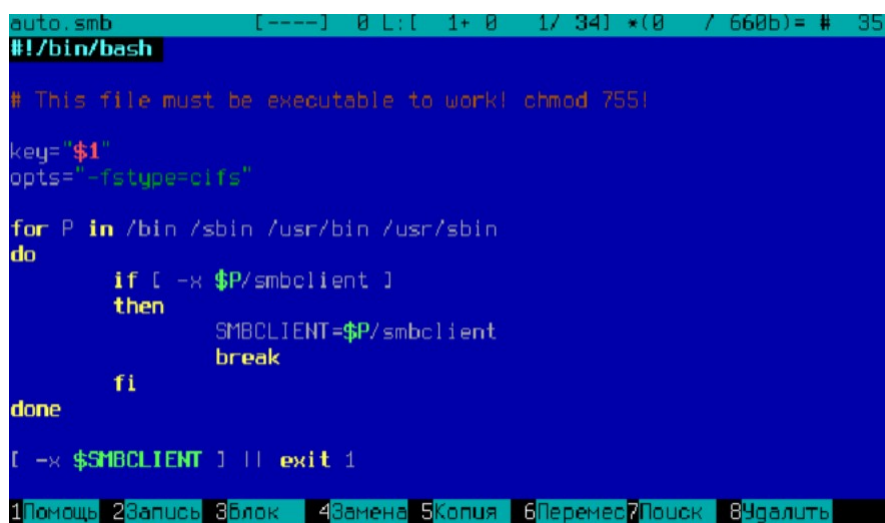
drwx----- 2 student users 4096 Сен 29 15:16 Documents
-rw-r--r-- 1 student users 208 Сен 27 13:49 .dvipsrc
-rw-r--r-- 1 student users 1637 Сен 27 13:49 .emacs
...

```

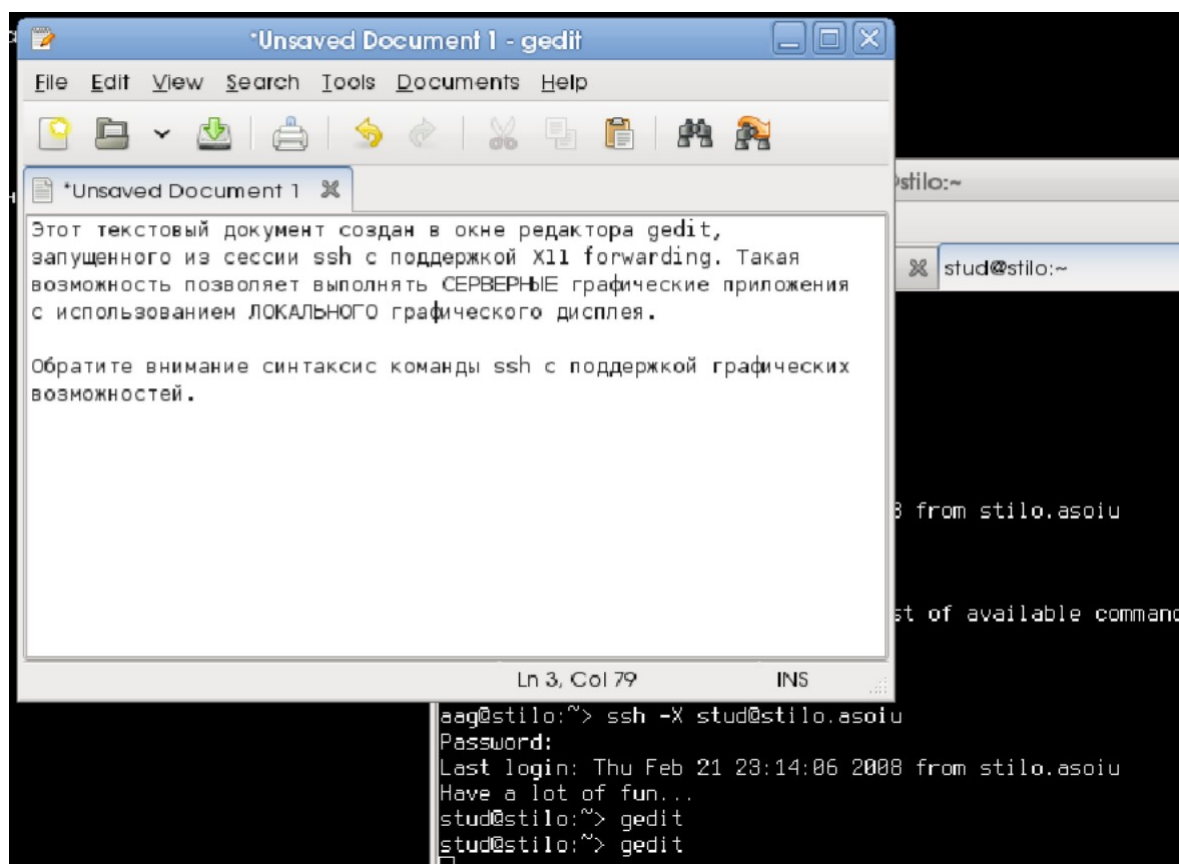
3. Мс файл менеджерін орындау



4. mcedit-те файлды өңдеу



5. SSH сеансынан іске қосылған gedit редакторы (сервермен байланыс орнатқан кезде-X параметрін көрсету керек және ssh серверінің конфигурациясын одан әрі конфигурациялау қажет болуы мүмкін)



SSH пәрмені қашықтағы компьютерде және қашықтағы қабықты жүктемей командаларды орындау үшін қолданыла алады. Бұл жағдайда қажетті пәрмен SSH параметрі ретінде беріледі:

```
ssh <hostname> <command>
```

Мысалы, қашықтағы edu компьютерінде `ls /usr/share/doc` пәрменін орындағыңыз келсе `edu.asoiu`, шақыруына теріңіз:

```
ssh edu.asoiu ls /usr/share/doc
```

Дұрыс құпия сөзді енгізген кезде экранда `/usr /share/doc` мазмұны пайда болады және сіз қабық шақыруына ораласыз. Айта кету керек, барлық командаларды осылай орындау мүмкін емес. `mc` іске қосудан бас тарту мысалы:

```
aag@stilo:~> ssh student@edu.asoiu mc
Password:
Cannot get terminal settings: Недопустимый аргумент (22) // Ошибка! Невозможно получить параметры терминала
TERM environment variable needs set.
aag@stilo:~>
```

Файлдарды көшіру

Қорғалған қосылым арқылы компьютерлер арасында файлдарды тасымалдау үшін `scp` (`secure copy`) пәрменін пайдаланыңыз.

Жергілікті файлды қашықтағы компьютерге жіберу үшін бұл пәрмен келесідей қолданылады:

```
scp localfile username@tohostname:/newfilename
```

`localfile` бастапқы файлды және комбинацияны білдіреді `username@tohostname:/newfilename` тағайындауды анықтайды.

Жергілікті `somefile` файлын `edu` компьютеріне жіберу үшін `edu.asoiu` пайдаланушы атымен шақыруға қабықшаларды енгізіңіз (`username`-ді өз атыңызбен ауыстырыңыз):

```
scp somefile username@edu.asoiu:/home/username
```

Бұл жағдайда жергілікті *somefile* файлы edu серверіндегі */home/username/somefile*-ге көшіріледі *edu.asoiu*.

Қашықтағы файлды жергілікті компьютерге жіберу үшін пәрмен қолданылады

```
scp username@tohostname:/remotefile /newlocalfile
```

remotefile бастапқы файлды, ал *newlocalfile* тағайындауды білдіреді.

Бастапқы файлдар ретінде бірнеше файлдарды тізімдеуге болады. Мысалы, каталог /жүктеу мазмұнын қашықтағы edu компьютеріне жүктеу деп аталатын бар каталогқа жіберу үшін *edu.asoiu*, қабық шақыруына келесі пәрменді енгізіңіз:

```
scp /downloads/* username@edu.asoiu:/uploads/
```

SSH сеансын аяқтау

SSH протоколының жұмысын аяқтау үшін пайдаланушы сеансын аяқтайтын және байланысты үзетін *exit* пәрменін орындау қажет:

```
...
-rw-r--r-- 1 student nobody 1088 Фев 20 04:18 readme.txt
-rw-r--r-- 1 student nobody 2194 Окт 21 11:04 .recently-used.xbel
drwxr-xr-x 2 student nobody 4096 Окт 21 10:59 .skel
drwx----- 2 student nobody 4096 Сен 27 15:13 .ssh
student@stilo:/etc> exit
logout // сеанс пользователя завершен
Connection to edu.asoiu closed. // соединение закрыто

aag@stilo:~> // возврат в локальную оболочку
```