

EXERCISE 1: OPERATION SYSTEM LINUX AND FILE TRANSFER PROTOCOL (FTP)

INTRODUCTION

In our laboratory experiments a server will occasionally be used. The domain name of this server computer is »**protokoli.lkn.fe.uni-lj.si**«. The Linux operating system (Ubuntu distribution) runs on it. On this server computer, http, ssh, ftp, smtp, pop3... server processes (daemons) run. This operating system does not offer a graphical user interface (GUI) for usual work, therefore a character-oriented user interface will be used via the ssh application-layer protocol, using the »putty« application. The most important linux commands are shown in the following table.

command	functionality
passwd	set or change user password
pwd	display working directory
cd <i>directory</i>	change working directory to <i>directory</i>
cd ..	change working directory to parent directory
cd \$HOME	change working directory to user's home directory
mkdir <i>directory</i>	create (make) directory <i>directory</i>
rmdir <i>directory</i>	delete (remove) directory <i>directory</i>
ls -l	list the contents of the working directory
ls -l <i>directory</i>	list the contents of the directory <i>directory</i>
cp <i>file1 file2</i>	copy file <i>file1</i> into file <i>file2</i>
mv <i>file1 file2</i>	rename (move) <i>file1</i> into <i>file2</i>
cat <i>file</i>	list contents of a text file <i>file</i>
more <i>file</i>	list contents of a text file <i>file</i> , one screen at a time
chmod <i>ugo +/- rwx file</i>	change access permissions of user/group/others, adding/removing read/write/execute permission, for <i>file</i>
exit	exit the session

Every student, that attends the laboratory exercises will get his/hers username for the server »protkoli.lkn.fe.uni-lj.si«.

VIRTUAL TERMINAL

Virtual terminal is a distributed application that allows a client to work interactively on a remote server just as if the user's terminal were directly connected to the server. A virtual terminal provides for a user interface between a user on a client computer and the operating system on a distant server computer and can be either character- or graphic- oriented. A virtual terminal distributed application is supported by an application-layer protocol. The protocol relays data between the user's i/o devices (keyboard, mouse, screen) and the operating system of the distant computer. The most frequently used application-layer protocols are telnet (character-oriented), ssh (character-oriented, secure), rlogin (character-oriented, developed for unix), x windows (graphic-oriented).

YOUR TASK

Use the virtual terminal application putty.exe and the ssh protocol to **connect** to the server »protokoli.lkn.fe.uni-lj.si« with your username and password. *Use* the above listed linux commands to **explore** the linux environment.

Username: Your name and the first letter of your (first if more) surname, all in lowercase (eg. Jonh Newman: johnn).

Password: the same as your username.

FILE TRANSFER PROTOCOL

To transfer files from a client to a server computer or vice versa, special application-layer protocols are used that support file transfer applications. One of the most frequently used application-layer protocols supporting file transfer in the Internet is FTP (File Transfer Protocol). A usual arrangement of usage of file transfer application over the FTP protocol consists of two computers, a client and a server. In both of them, two layers of communication can be considered. In the above layer the distributed application runs, including two different applications in client and server computers, respectively (the application in server computer is usually a part of the server operating system); additionally, the operation of the distributed application is often controlled on the client side by a human user over a user interface between the user and the client application. In the lower layer the FTP protocol runs; an ftp protocol entity consists of two parts. One of them is the control function which controls an ftp session in which a user can browse the file system on the server, modify it (e. g. by renaming or deleting files) or transfer files between file systems of client and server computers; however, when setting up a new session, a user must be authenticated by the server with his/her username/password pair. Normally, the language used by a user to interact with the application over the user interface is different (although not necessarily) from the protocol used for communication between the two application-layer protocol entities, and depends on the application at the client side. The other part of the ftp protocol entity is the file transfer function which controls the transfer of files between the two file systems. For session control and file transfer respectively, two different channels (implemented as separate TCP connections) are used (the control TCP connection uses port no. 21, and the data connection uses port no. 20 on the server side. As an application-layer protocol of the TCP/IP protocol stack, the ftp protocol also provides for data format translation between the two computers. Apart from translation of character encoding, ftp also must translate the encoding of end-of-line indications in text files: in Windows operating system, the end-of-line marker consists of two control characters (CR-LF), while in unix and linux operating systems, the end-of-line marker is represented by a single \n (newline) character (hence text files must be transferred in the ASCII mode); binary files need no translation during transfer (hence binary files must be transferred in binary mode).

YOUR TASK

Use the ftp application on your personal computer to connect to the server »protocoli.fe.uni-lj.si«.

Try out the ftp application commands that are listed in the lower table. **Use** the »debug« option to see which ftp protocol data units correspond to different user interface commands.

command	functionality
ascii	set transfer mode to ASCII (default)
binary	set transfer mode to binary
cd <i>directory</i>	change remote working directory
debug	set debugging mode on/off (default is off)
delete <i>file</i>	delete remote <i>file</i>
dir	display the contents of remote working directory
disconnect	close ftp session
get <i>file</i>	copy <i>file</i> from server to client machine (both working dirs)
lcd <i>directory</i>	change working directory on local machine
mkdir <i>directory</i>	create a new <i>directory</i>
open <i>server</i>	open a new ftp session, connecting to <i>server</i>
put <i>file</i>	copy <i>file</i> from client to server machine (both working dirs)
pwd	display the name of remote working directory
quit	close a session end exit from application

Find or generate a simple text file and a simple »jpeg« image and copy them into the working directory on your PC. **Transfer** them to Linux machine and back to PC in both correct and incorrect transfer mode and **control** file integrity and length!

Use the application WireShark to monitor all the communication steps between the PC and the server. **Use** the following commands:

- get,
- put,
- dir.

Use the application FileZilla to transfer the text file you created earlier and **compare** the communication steps in Wireshark. **Write** about your findings in your report.