

Project topics - computer networks

1. (4) Present the next steps for installing and configuring a selected network service, e.g. a. Samba file server with a domain controller, b. WWW server with PHP, db, mod_rewrite support, for managing user passwords, c. Mail server (e.g. postfix, sendmail, etc.), d. Video streaming server (e.g. DLNA), e. WWW service based on any platform, for storing and managing user files (you can be inspired by: chomikuj, google drive, dropbox, iCloud, MediaFire, etc.), f. video transcoder in the form of a web application.
2. (3.5) Write a program in the form of a shell script to scan files on selected drives, and generate a report in the form of an HTML file containing the following information:
 - a. absolute file location
 - b. file name
 - c. file size
 - d. creation date
 - e. last modification date
 - f. MD5 checksum
 - g. SHA256 checksum
3. (5.0) Write a program in any language to synchronize directories with files from two different machines using the ICMP protocol. The program should work in a similar way to the rsync command.
4. (3.0) Write a script that counts files with a given extension for a given path (the user can provide several parameters at once, e.g. *.txt, *.jpg, *.docx). The program generates a report in the form of a text file or an html file.
5. (4.0) Write a script that creates a local repository of apt packages (the list is in sources.list) for a given platform (amd64, arm), based on a globally available server. The local copy should contain a structure that allows for local installation of packages. The script should work similarly to the apt-mirror program.
6. [4.5] Prepare a Linux system configuration that allows for regular backups of archived data from another remote Linux system. Only specified remote directories should be archived, archives should be incremental, with the possibility of restoring the full original data. Communication between machines should be encrypted.
7. [4] A set of bash shell scripts should be designed, which, based on text data from input files, will automatically configure firewall rules that allow/block traffic for given machines on given ports. The structure of the input files is as follows (each set in one line):

[ip_address][port][in/out][a/d] ; comment: "/" - only one of the options is allowed, [a/d] - accept, drop, in/out - data direction to/from the machine with the specified IP address.

Example string: [192.168.1.10][80][in][a]

Example call: network_firewall.sh machines.txt

8. [5] Write a program in any programming language (e.g. C/C++, Java, C#) that recursively downloads all files with the specified extensions for a given website and saves them on the local disk in the appropriate directories.

Example call: web_downloader.exe https://wp.pl *.jpg|logo.png

9. [5] Write a program in a selected programming language that implements a simple WWW server. The implementation should be implemented on sockets. The server should allow simultaneous access to resources for many users. Data returned by the server should be only static files (*.txt, *.html).
10. [5] Write a program in a selected programming language that implements a simple FTP server. The implementation should be implemented on sockets. The server should allow simultaneous access to resources for many users. Data returned by the server can be text or binary files. The communication mode can be passive or active.
11. [*3+] Prepare a configuration of a web server with PHP support, which contains a web page that allows you to save your favorite web page addresses. The forms should only allow you to enter the full web page address, and the final result should be adding a new page to the page_addresses.txt file located in the main directory of the web server.
12. [*3+] Prepare a configuration of a dynamic web server in Python, which contains a web page that allows you to save your favorite web page addresses. The forms should only allow you to enter the full web page address, and the final result should be adding a new page to the page_addresses.txt file located in the main directory of the web server.
13. [3.5] Prepare a web service that allows you to upload files to the server using the drag & drop method, and returns the hash function value for the uploaded file. As a hash function, you can choose e.g. MD5, SHA1, SHA-256, SHA-512.
14. [3] Write a program that scans all ports of specified IP addresses (from a *.txt file), and returns information about which ports are open. The program can be written as a shell script, or compiled into a standalone executable program.

15. [3.5] Write a program that checks for the following domain names (in a text file):

- owner,
- registration date,
- expiration date,
- calculates how many days are left to expiration from the current date. The data should be returned in the form of a report to the text file "raport.txt".

16. Other topics are possible, as modifications of the above, or as completely different proposals (agreement with the instructor required).