**LINUX EXERCISE (LAB 07)**

Exercise 1. Manage services using systemd:

1. Check whether systemd is available in your Linux system (systemctl command)
2. List all services in your Linux OS
3. Check whether Apache server is installed or not (httpd). If not, please install Apache server
4. Start Apache server
5. Show all using files of Apache server
6. Try to access to your web server from another computer using IP address. If we want access the web server using the domain name (test.com), which steps should we do?
7. Turn off the Apache server’s service, try to access to the Apache server from another computer
8. Configure your system so the Apache server boots up along with your computer
9. Restart your computer using the command line. Login to your system. How could we know whether the Apache server is working or not?

Exercise 2. Change the run level of your system using systemd

1. Check the default run level of your system
2. Display all run levels of your system
3. Change the default run level of your system from GUI to console (or reverse).
4. What is the notification when you try to change the default run level? Is there any other way to change the default run level of your system without using systemctl?

Exercise 3. Create and manage a new service

1. Create a new file server.php with the following content. Try to guess the purpose of this file? (Hint: you can do it after executing the step 3)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | <?php | |  |  | |  | $sock = socket\_create(AF\_INET, SOCK\_DGRAM, SOL\_UDP); | |  | socket\_bind($sock, '0.0.0.0', 10000); | |  |  | |  | for (;;) { | |  | socket\_recvfrom($sock, $message, 1024, 0, $ip, $port); | |  | $reply = str\_rot13($message); | |  | socket\_sendto($sock, $reply, strlen($reply), 0, $ip, $port); | |  | } | |

1. Run “php server.php” to start the php server. (You might need to install php server – php\_cli)
2. Check the status of php server by login to another account (different terminal). Run the command “nc 127.0.0.1 10000” and then type “Hello world” after connecting with local php server. Is the printed message “Uryyb, jbeyq”?
3. To convert the PHP server to a service, create a file /etc/systemd/system/rot13.service with the following content

|  |
| --- |
| [Unit] Description=ROT13 demo service After=network.target StartLimitIntervalSec=0[Service] Type=simple Restart=always RestartSec=1 User=centos ExecStart=/usr/bin/env php /path/to/server.php  [Install] WantedBy=multi-user.target |

1. Try to start the service rot13 using systemctl
2. Use systemctl to configure rot13 so it can boot up along with the Linux OS