

14.1 chargen

In this part you will get to know the chargen protocol (RFC 864). chargen listens on port 19 and supports TCP and UDP.

14.1.1 Enable chargen on your Device

- Ensure that xinetd is enabled on your system, by editing `/etc/xinetd.d/chargen`. Ensure that you have the following line for TCP and UDP:

```
disable = no
```

- Afterwards, restart the service (if you had to change it):

```
sudo service xinetd restart
```

- Finally, use the following command to test its working:

```
netcat localhost chargen
```

14.1.2 Clients

- Develop a UDP client that connects to the server running on your PC. It should be able to connect to retrieve a set of random characters from the server and print it to the console.
- Now develop another client for chargen, but this time using TCP.

14.1.3 Servers

- Write a minimal server in TCP that continuously sends the sequence (1234567890) as long as the client is connected.
- Extend your server in a way that multiple concurrent requests are served (no use-once-server).
- Extend your server in a way that multiple connections can be established and served in parallel (using `fork()` or `threading`).
- *HONour Task: Check out the `random` module to generate really random sequences of printable characters.*
- *HONour Task: Check out the data syntax section in the RFC and implement the same behaviour.*
- *HONour Task: Write a `chargen` server using UDP sockets.*

14.2 DNS

In this task you will develop a minimal DNS client.

- Start by composing an empty DNS message and send it on a UDP socket. You can check if you did it correctly by using Wireshark.

- You client should send a name query for "uni-saarland.de" to 134.96.252.20 (the university's name server). Start phrasing your question as defined in the RFC.
- Parse the response to get the IP address of the university's web page out.
- *HONour Task: Parse additional fields from the response.*