

1. FileServer

Schrijf een programma in Java dat gebruik maakt van een socket connectie die een client toelaat om een bestand (text-file) te specificeren zodat de server de inhoud van dit text-bestand naar de client stuurt.

Indien het bestand niet bestaat dan wordt een passende boodschap afgedrukt op het scherm.

Vervolgens laat je toe dat de client het bestand wijzigt en het gewijzigde bestand terug stuurt naar de server die het gewijzigde bestand stockeert. De gebruiker wijzigt de inhoud van het bestand en stuurt de gewijzigde versie terug naar de server.

Indien de client de bestandsnaam niet wijzigt zal het originele bestand op de server overschreven worden.

| | |
|---|--------------------------------|
| Het protocol: File Server ↔ File Client | |
| in: | actie readFile |
| "READ" | out: "READ" |
| in:filename | out: filename |
| exists? | |
| ja | in: antwoord |
| out: "FOUND" | antwoord= FOUND |
| out: regel/regel | in: regel/regel tot EOF |
| out: EOF | |
| nee | |
| out: "NOTFOUND" | |
| | actie updateFile |
| "REWRITE" | out: "REWRITE" |
| in:filename | out: filename |
| in: regel/regel tot EOF | out: contents + EOF |

2. UDPPinger

Bij deze opdracht implementeer je een eenvoudige op UDP gebaseerde ping-client. De functionaliteit is vergelijkbaar met het standaard ping-commando dat in moderne besturingssystemen wordt gebruikt.

Standaard werkt Ping door Internet Control Message Protocol (ICMP) ECHO-berichten te verzenden die door de ontvangende machine terug gestuurd worden naar de verzendende machine.

Java heeft geen enkele functionaliteiten om ICMP-berichten te verzenden of te ontvangen.

Wij implementeren in deze opdracht echter het pingen op de applicatielaag met standaard-UDP-sockets en –berichten.

Bij het pingen zendt een client een pakket naar een server. Deze server zendt het pakket ongewijzigd naar de client terug.

In het startproject staat java-code van de PingServer die je eerst grondig bestudeert.

Bij de PingServer wordt een LOSS_RATE gebruikt om pakketten die verloren gaan in het netwerk te simuleren. Verder wordt ook gebruik gemaakt van AVERAGE_DELAY om een vertraging te simuleren.

Implementeer zelf de PingClient: de client moet 10 ping's naar de server sturen met een tussentijd van ongeveer 1 seconde. Een ping bestaat uit "ping volgordenummer tijndindicatie". Na elke ping wacht de client 1 seconde om een antwoord van de server te ontvangen. Indien het antwoord van de server niet binnen de 1 seconde bij de client toe komt dan gaat de client er van uit dat zijn pakket of het antwoordpakket van de server verloren is gegaan in het netwerk.

Bij het uitvoeren van PingClient wordt een host en een poortnummer meegegeven.

Optioneel:

- De tijd tussen het verzenden en het ontvangen (round-trip tijd) wordt reeds bepaald. Bepaal ook de minimum, maximum en gemiddelde round-trip tijd

VB run:

```
Output
OOPIII_NET_UDPPinger (run) % OOPIII_NET_UDPPinger (run) #2 %
run:
Ping reply server started: uses port 5555
Received from 127.0.0.1 : Ping #1 1480867955612 (12/04/2016 17:12:35)
    Reply not sent.
Received from 127.0.0.1 : Ping #2 1480867956612 (12/04/2016 17:12:36)
    Reply not sent.
Received from 127.0.0.1 : Ping #3 1480867957612 (12/04/2016 17:12:37)
    Reply sent.
Received from 127.0.0.1 : Ping #4 1480867957626 (12/04/2016 17:12:37)
    Reply sent.
```

```
Output
OOPIII_NET_UDPPinger (run) % OOPIII_NET_UDPPinger (run) #2 %
run:
Ping #1 No response was received from the server
Ping #2 No response was received from the server
Ping #3 1480867957612 (12/04/2016 17:12:37) Received from 127.0.0.1 (RTT=10ms)
Ping #4 1480867957626 (12/04/2016 17:12:37) Received from 127.0.0.1 (RTT=180ms)
Ping #5 1480867957806 (12/04/2016 17:12:37) Received from 127.0.0.1 (RTT=164ms)
Ping #6 1480867957970 (12/04/2016 17:12:37) Received from 127.0.0.1 (RTT=51ms)
Ping #7 1480867958021 (12/04/2016 17:12:38) Received from 127.0.0.1 (RTT=44ms)
Ping #8 No response was received from the server
Ping #9 1480867959065 (12/04/2016 17:12:39) Received from 127.0.0.1 (RTT=112ms)
Ping #10 1480867959177 (12/04/2016 17:12:39) Received from 127.0.0.1 (RTT=157ms)
min: 10, max: 180, gem: 71
BUILD SUCCESSFUL (total time: 3 seconds)
```