# Tests

| **Test #** | **Test procedure / description** | **Expected Outcome** | **Actual Outcome & Remarks** | **Pass / Fail** |
| --- | --- | --- | --- | --- |
| 1 | Epoll server listens for connections on its specified port, so Hercules client may connect to it.   1. make clean 2. make epoll\_svr 3. ./epoll\_svr.out -p 7000 -n 4 4. connect to server using Hercules net tool | Epoll server listens on the specified port, so Hercules client can connect to it. | As expected | Pass |
| 2 | Select server listens for connections on its specified port, so Hercules client may connect to it.   1. make clean 2. make select\_svr 3. ./select\_svr.out -p 7000 -n 4 4. connect to server using Hercules net tool | Select server listens on the specified port, so Hercules client can connect to it. | As expected | Pass |
| 3 | Thread server listens for connections on its specified port, so Hercules client may connect to it.   1. make clean 2. make thread\_svr 3. ./thread\_svr.out -p 7000 -n 300 4. connect to server using Hercules net tool | Thread server listens on the specified port, so Hercules client can connect to it. | As expected | Pass |
| 4 | Epoll server echoes back what is sent to it by a connected Hercules client.   1. perform test 1 2. send data to server via Hercules | Sent data should also be displayed in received text box. | As expected | Pass |
| 5 | Select server echoes back what is sent to it by a connected Hercules client.   1. perform test 2 2. send data to server via Hercules | Sent data should also be displayed in received text box. | As expected | Pass |
| 6 | Thread server echoes back what is sent to it by a connected Hercules client.   1. perform test 3 2. send data to server via Hercules | Sent data should also be displayed in received text box. | As expected | Pass |
| 7 | Epoll server, when under heavy load, takes advantage of multiple CPUs and spreads out work over all processors. | System monitor should display that all cores are being used. | As expected | Pass |
| 8 | Select server, when under heavy load, takes advantage of multiple CPUs and spreads out work over all processors. | System monitor should display that all cores are being used. | As expected | Pass |
| 9 | Thread server, when under heavy load, takes advantage of multiple CPUs and spreads out work over all processors. | System monitor should display that all cores are being used. | As expected | Pass |

## Screenshots

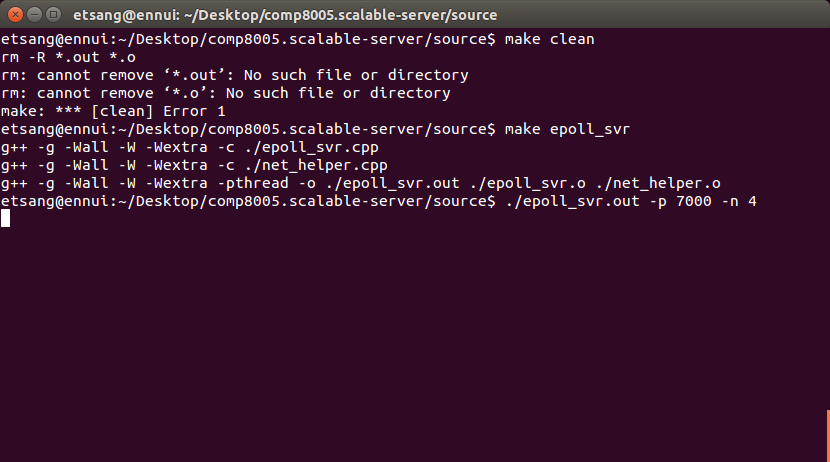


Figure Test 1, compile and run epoll server

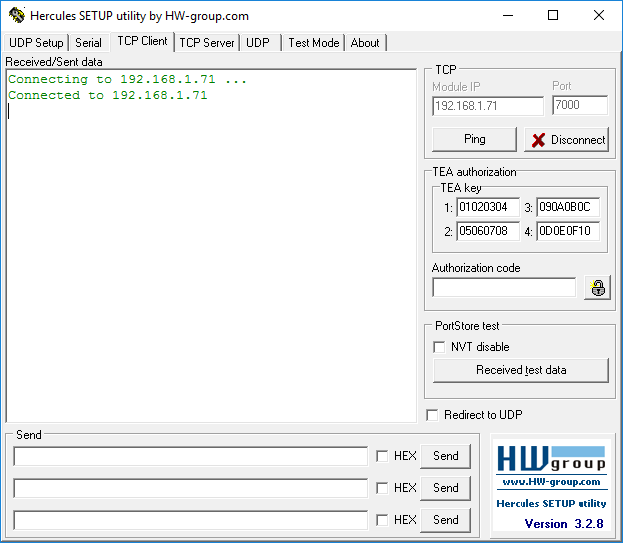


Figure Test 1, Hercules client connects to epoll server

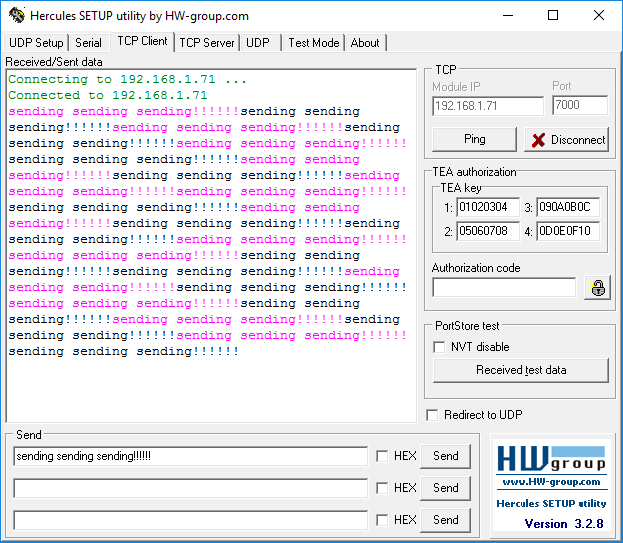


Figure Test 4, epoll server echoes back what is sent to it via Hercules client

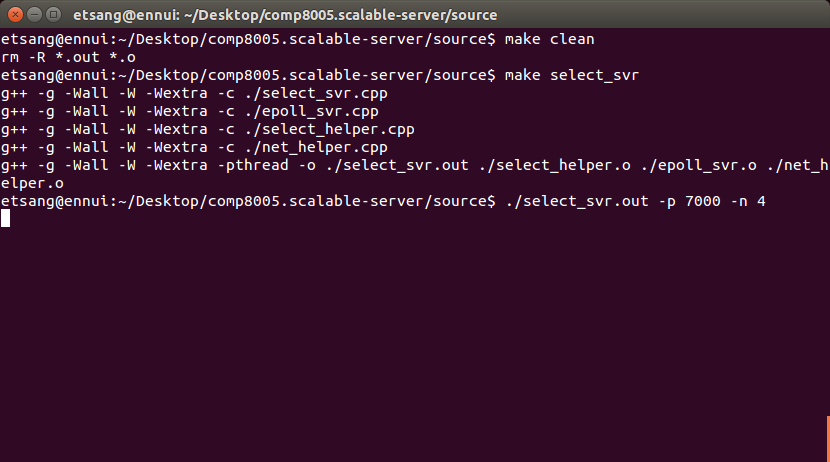


Figure Test 2, compile and run select server

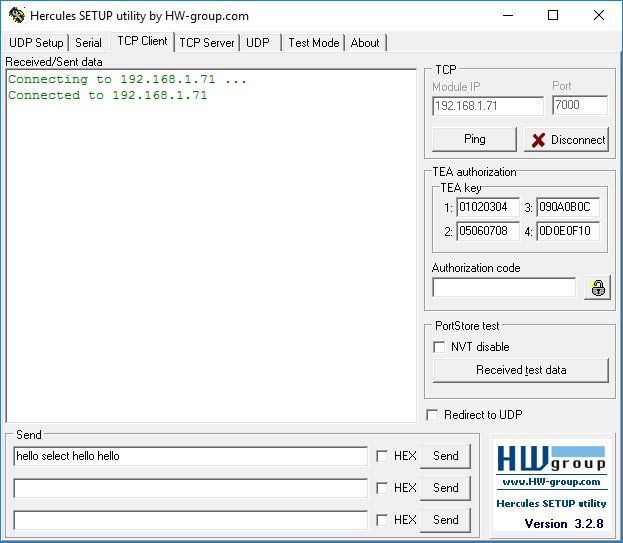


Figure Test 2, Hercules client connects to select server

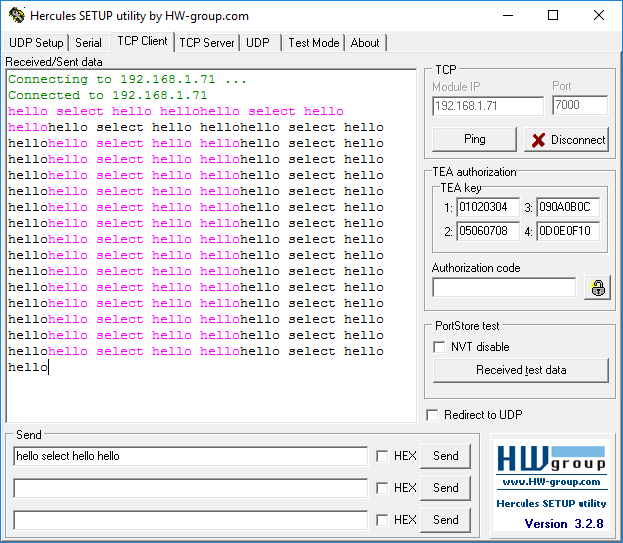


Figure Test 5, select server echoes back what is sent to it via Hercules client

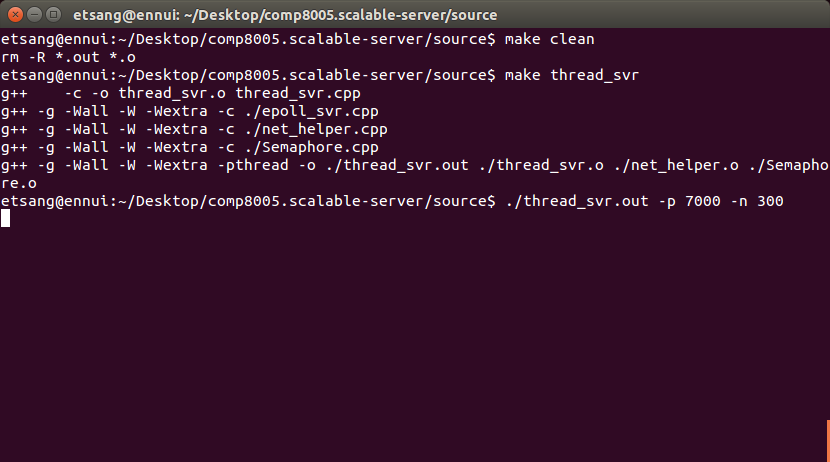


Figure Test 3, compile and run multi-thread server

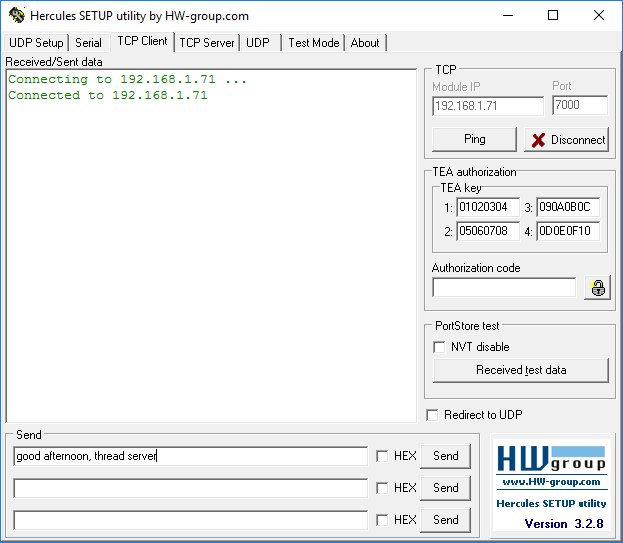


Figure Test 3, Hercules client connects to thread server

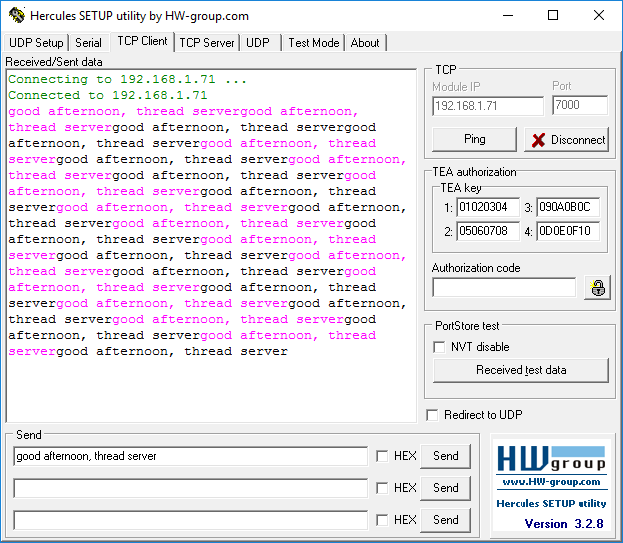


Figure Test 6, thread server echoes back what is sent to it via Hercules client



Figure Test 7, epoll server does load balancing



Figure Test 8, select server does load balancing



Figure Test 9, threaded server does load balancing