Event-Driven Single Process Server

Submitted By
Chetan Arora, Neethu Mariya Joy
2016A8PS0346P, 2016A7PS0119P

Usage

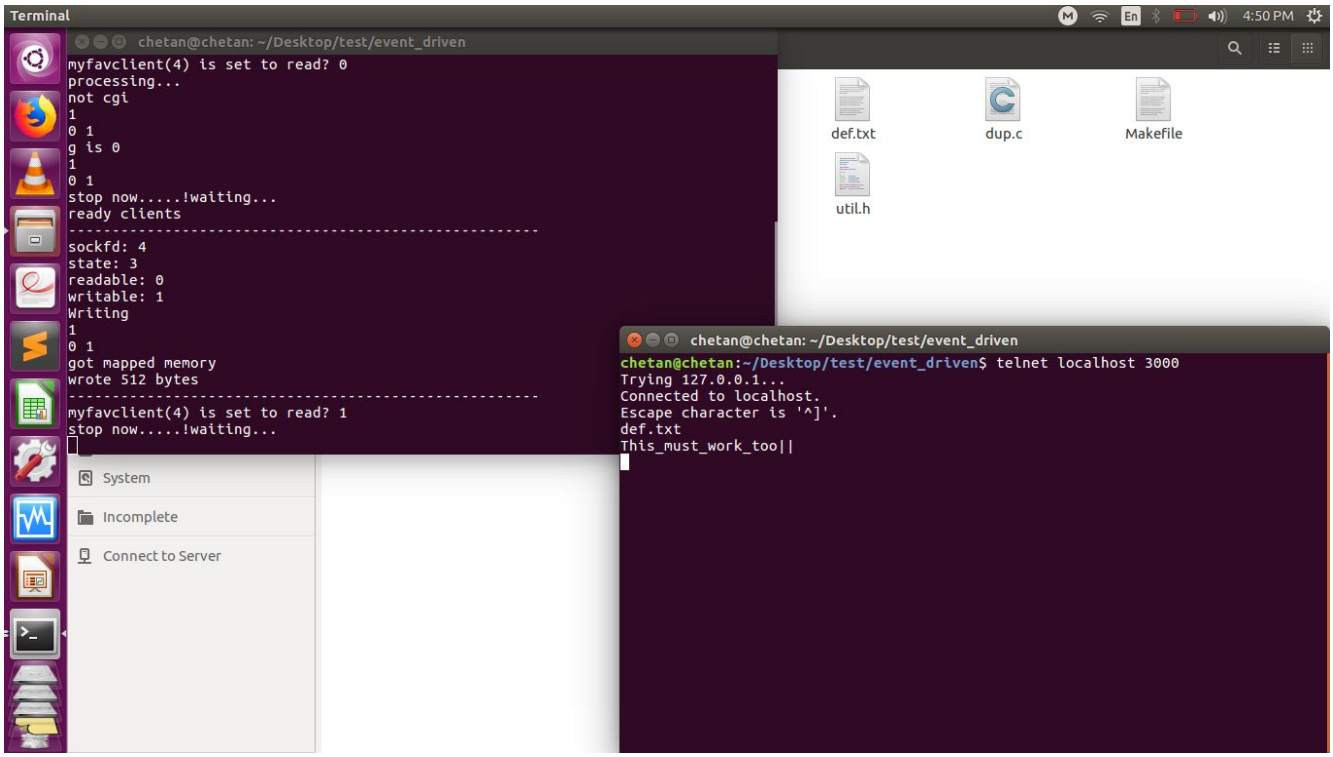
To start server
make
./server P
Where P is port number

Telnet can be used as a client.
telnet localhost P

Design

1. The server uses only one process and no threads.
2. For I/O tasks like fetching a file in main memory, it uses child processes.
3. The server uses non-blocking i/o calls on sockets.
4. For checking the states of socket descriptors it uses select method.
5. Every client connected gets stored in a client structure which contains its current state and client request and how much data already been written to it.
6. The client requests a file with file name e.g. abc.txt.
7. Fn displayMincore() checks if the file is in main memory, and if it not in main memory a function loadFile(helper child) loads it into main memory.
8. If the file is large, server sends the file in small chunks and stores the offset of data transferred in client structure.
9. With this, server never wastes too much time on one client.

Result


Terminal
4:50 PM
chetan@chetan: ~/Desktop/test/event_driven
myfavclient(4) is set to read? 0
processing...
not cgi
1
0 1
g is 0
1
0 1
stop now.....!waiting...
ready clients
sockfd: 4
state: 3
readable: 0
writable: 1
Writing
1
0 1
got mapped memory
wrote 512 bytes
myfavclient(4) is set to read? 1
stop now.....!waiting...
def.txt
dup.c
Makefile
util.h
System
Incomplete
Connect to Server
chetan@chetan: ~/Desktop/test/event_driven
chetan@chetan:~/Desktop/test/event_driven$ telnet localhost 3000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
def.txt
This_must_work_too||