```
=====================================
// Simple/Basic Server

server_sd = socket();
bind(server_sd);
listen(server_sd);

for (;;) {
   // wait for new connection request
   sd = accept(server_sd);

   // get request from client
   read(sd);
   parse_request();

   // get or compute response

   // send response to client
   write(sd);
}
-------------------------------------
```

```
------------------------------------
// Forking Server
// (Create Process / Thread to Handle Request)

for (;;) {
   // wait for connection
   // get request
   // fork/create child to handle request
   // pass request to child
}

forked_child(request) {
   // get request from parent
   // decode request
   // handle request / return result
}
------------------------------------
```

```
-----------------------------------
// Pre-Forking (Process or Thread Pool)

for (i=0; i<P; i++) {
    // fork/create a worker
}

for (;;) {
   // wait for connection
   // get request
   // pick child/worker and send request
}

forked_child(request) {
    // get request from parent
    // decode request
    // handle request / return result
}
-----------------------------------
```

```
-----------------------------------
// Event Driven
// (used in userver, Zeus, and
// original Harvest/Squid proxy caches)

server_sd = socket();
bind(server_sd);
listen(server_sd);
```

```
for (;;) {
   readfdset = rdfdset; writefdset = wrfdset;
   n = select(max_sd, &readfdset, &writefdset,
              exceptfds, &timeout);

   for (i=0; i<max_sd; i++) {
       if (FD_ISSET(i, &readfdset)) {
           if (i == server_sd) {
               // get new connection
               new_sd = accept(server_sd);
               FD_SET(new_sd, rdfdset);
               FD_SET(new_sd, wrfdset);
           } else {
               read_request(sd);
               parse_request();
           }
       }
       if (FD_ISSET(i, &writefdset)) {
           // get response
           write_response(sd);
       }
   }
}
```

==========