Non-blocking server.

Every socket must be non-blocking in this order, we are not going to wait until data is available, only through epoll we are notified of the change in a fd(we don’t know which fd).

Build a logic to find the end of the request, instead of waiting to the read return 0;

Remember, read may return zero and is not the end of the request.

Using a thread pool to deal with responses, avoid thread overhead;  
  
  
We only close the connection when, the client says so.  
We must change

TO DO:

Rafa:  
 Finalizar o parser; -> Parsing

Edgy case dos inputs; -> Parsing

Cookies e sessões -> Bônus

TimeOut -> Conexão

Lidar com vários servers na mesma porta mas com o nome diferente; -> Conexão

Tereza:  
 Loop infinito dentro da CGI; -> CGI

Criar diversos scripts para testar -> CGI

Crash dentro da CGI; -> CGI  
 Garantir que nunca trava; -> CGI  
 Funcionar com várias CGIs -> Bônus

(O que devemos fazer caso seja loop infinito?)

Arthur:  
 Lidar com vários ficheiros; -> Ficheiros

Lidar com ficheiros de extensões diferente; -> Parsing

Chunked Request -> Conexão

Lidar com request inválidos -> Parsing

Validar o request -> Lexer

Todos:

Status code;  
 Teste de estresse para travar;

[...]

Para não travar quando o a CGI encontra-se em loop infinito:  
Utilização do waitpid com a flag **WNOHANG**, para verificar o estado do child process.  
  
Teremos dois time out, um para a conexão e outro para CGI  
  
Exemplo:  
 CGI timeout: 4s

Connection timeout: 20s

Tempo de espera do Epoll() 10000 microsegundos,  
 Verificar se deu o timeout da conexão  
 Se fim fecha as conexões

Verifica se deu timeout da CGI

Se fim fecha a conexão

Se nao deu timeout da CGI

Escreve a resposta

//

The value of *options* is an OR of zero or more of the following constants:

**WNOHANG**

return immediately if no child has exited.

/\*\*\*\*\*\*\*\*/

Files to be uploaded:  
 No parsing caso o diretório não exista o cria;  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Limit the heap size:

ulimit -d <heap\_size\_in\_kilobytes>

./your\_program

TO DO:

Rafa:  
 Finalizar o parser; -> Parsing

Edgy case dos inputs; -> Parsing

Cookies e sessões -> Bônus

Lidar com vários servers na mesma porta mas com o nome diferente; -> Conexão

Tereza:

Criar diversos scripts para testar -> CGI

Crash dentro da CGI; -> CGI  
 Garantir que nunca trava; -> CGI  
 Funcionar com várias CGIs -> Bônus

(O que devemos fazer caso seja loop infinito?)

Arthur:  
 Lidar com vários ficheiros; -> Ficheiros => Feito

Lidar com ficheiros de extensões diferente; -> Parsing => Feito

Chunked Request -> Conexão

Lidar com request inválidos -> Parsing

TimeOut -> Conexão

TimeOut -> CGI

Errors Status code-> Request and response cycle

Todos:

Status code;  
 Teste de estresse para travar;

[...]