**Description**

1. int main( int argc, char \*\*argv );

Main function

If main function get request of client, parent process make his child process.  
And make socket, bind port, send message to "parseMessage" function.  
"sendPage" function fine and send that file to client.  
If "sendPage" function can't find page, he will open open.html.

2. int parseMessage( char \*str, char \*\*request, char \*\*type, char \*\*httpver, char \*\*accepttype );

" parseMessage " is parsing client message, and find values.   
And return those values to " \*\*request ", " \*\*type ", " \*\*httpver ", " \*\*acceptype ".

3. int sendPage( int sockfd, char \*filename, char\* httpver, char\* accepttype );

" sendPage " is find file based on " \*filename ".  
If he can't find page, he will send default page( open.html ).  
He can find page, he will read that file and write page in line by line.

**Problem**

1. Originally, I think that if my server get message of client, he will send to request message, close socket and be died. So. I make my server to making his child process. But my original code died before the child process.

: So I take my server waiting child process by putting this code ( while(waitpid(-1,NULL,WNOHANG) > 0); )

2. If I open "localhost:< port number >" in browser, my " parseMessage " function return "Get", "/", "HTTP/1.1".   
So my " sendPage " function try to find "/" page.   
It return "error".

:So in this case, I make my server return "open,html".

3. I want to know "Context-length". So I use "lseek( fd, 0, SEEK\_END )", success to get file length. But "lseek" is moving start pointer to end.

: So I use "lseek( fd, 0, 0 )" one more time to move start pointer.

**Sample tutorial**

server on : ./server< port number >  
client on ( browser ) : localhost:< port number >  
default page : localhost< port number >  
index.html ( index.mp3, index.jpg, index.gif, index.pdf )   
: localhost< port number >/index.html ( index.mp3, index.jpg, index.gif, index.pdf )

ex)

server on : ./server 4000  
client on : localhost:4000  
-> localhost:4000/open.html  
localhost:4000/index.jpg  
localhost:4000/index.gif  
localhost:4000/index.mp3  
localhost:4000/index.pdf