hchatd - HTTP chat daemon

hchatd is a daemon, that acts as a simple HTTP server. Client can create new chat room, or connect to existing chat room. Any number of clients can connect to chat room. Once in a chat room, clients can on-line send and receive messages. Also after connection, client can receive a few recent messages.
Program incomporates client's side, i.e. HTML and JavaScript, that runs in user's HTTP browser. Author believes it's simple and standard compliant enough to work with any contemporary browser (Chromium, Firefox and IE tested).
If user creates a chat, he has to send the generated URL to another user, in order to communicate with him. Or he can mark the chat as "public", i.e. listed.

Program schema

hchatd is single-process, single-thread daemon, using `poll` to work with multiple sockets. It receives data from client, util `\r\n\r\n` sequence appears. Then it runs finite-state automaton to parse the HTTP header. It serves following requests:

- `GET /`
  Index page
  Returns: HTML page where user can either choose from list of existing (public) chat or create new chat
- `GET /chat_create`
  Creates new chat room. Parameters can specify subject of chat and flag if it should be enlisted on the / page. Every chat has it's unique 32 char identifier (further referenced as xxx)
  Returns: HTTP redirect to `/chat/xxx`
- `GET /chat/xxx`
  Chat page. Every time client calls this request, server generates new unique identifier of client and returns its identifier as HTTP-Cookie - that enables the server to identify authors of incomming message posts
  Returns: HTML page including JavaScript that requests new messages from server (using `GET /msg/xxx/yyy`) and sends new messages (using `POST /send/xxx`)
- `GET /msg/xxx/yyy`
  Returns message with id yyy (messages in a chat are numbered in continuous series) from chat xxx. If such message doesn't exist yet, the request hangs until it appears.
  Returns: Specified message (as plain text)
- `POST /send/xxx`
  Sends new message to chat xxx. Server stores the message and checks if any client is waiting for such a message.
  Returns: HTTP 200

Usage

hchatd takes these optional arguments:

-h show help and finish

-d work as a daemon

-p **portnum** to specify port number (default is `1234`)