LibCli 使用教學

先貼範例程式碼

```
#include <libcli.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <time.h>
#include <string.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include "libcli.h"


time_t presentTime;

int cmd_system(struct cli_def * cli, const char *Msg, char **argv, int argc){
    FILE *result = popen(argv[0], "r");
    char str[10000];
    while(!feof(result)){
        memset(str, 0, 10000);
        fgets(str, 10000, result);
        cli_print(cli, "%s", str);
    }
    return CLI_OK;
}

int cmd_authMsg(const char *usname, const char *passwd){
  time(&presentTime);
  char *timeMsg = ctime(&presentTime);
  printf("[Log]User %s try to login on %s ", usname, timeMsg);

}

int cmd_deleteHello(struct cli_def * cli, const char *Msg, char **argv, int argc){
  cli_unregister_command(cli, "hello");
  return CLI_OK;
}

int cmd_hello(struct cli_def * cli, const char *Msg, char **argv, int argc){
  cli_print(cli, "Hello Kitty.");
  return CLI_OK;
}

int cmd_sanjuan(struct cli_def * cli){
  cli_print(cli, "I don't like SanJuan");
  return CLI_OK;
}

void cli_setting(struct cli_def **cli){
  struct cli_command *sys;

  *cli = cli_init();

  cli_set_hostname(*cli, "AoK_Blast");

  cli_set_banner(*cli, "It is a test cli command line program.");

  cli_allow_user(*cli, "aokblast", "123");
  cli_set_auth_callback(*cli, cmd_authMsg);

  sys = cli_register_command(*cli, NULL, "system", cmd_system, PRIVILEGE_UNPRIVILEGED, MODE_EXEC, "Do normal linux operation on remote server.");

  cli_register_command(*cli, NULL, "hello", cmd_hello, PRIVILEGE_UNPRIVILEGED, MODE_EXEC, "Say hello");

  cli_register_command(*cli, NULL, "dhello", cmd_deleteHello, PRIVILEGE_UNPRIVILEGED, MODE_EXEC, "Delete hello command");

  cli_regular(*cli, cmd_sanjuan);

  cli_allow_enable(*cli, "Test");

}


int main(int argc, char *argv[]) {
  struct sockaddr_in servaddr;
  struct cli_def *cli;
  int socketFd, runningFd, on;

  cli_setting(&cli);

  socketFd = socket(AF_INET, SOCK_STREAM, 0);
  setsockopt(socketFd, SOL_SOCKET, SO_REUSEADDR, &on, sizeof(on));

  memset(&servaddr, 0, sizeof(servaddr));
  servaddr.sin_family = AF_INET;
  servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
  servaddr.sin_port = htons(12345);


  bind(socketFd, (struct sockaddr *)&servaddr, sizeof(servaddr));

  listen(socketFd, 50);

  printf("Server runs on port: 12345\n");

  while ((runningFd = accept(socketFd, NULL, 0))) {
    cli_loop(cli, runningFd);
    close(runningFd);
  }

  cli_done(cli);

  // Free data structures

  return 0;
}
```

從主程式開始說，這個城市會先建立必要的結構(cli_def)，然後進去我們自己的setting function(等等講)，之後會嘗試建立socket連線，等到client端連線進來的時候，整個program便交給libcli用cli_loop(接管)，cli_loop裡面的實做是用一個無限迴圈對client端的輸入做判斷，當然還有我們預先setting(cli_setting)好的東西。

現在來說最重要的cli_setting()，作為一個手動物件，當然要先經過library給我們的init function(cli_init())來做初始化，裡面包括對預設函數的function pointer指定還有一些變數的設定，跟cli_done()是成對的，一個負責malloc, 一個負責刪除。

接下來可以用cli_set_hostname()為主機設定名稱，會顯示在命令行的旁邊，還有用cli_set_banner()設定歡迎語句，在每次使用者登入成功的時候都會顯示。

然後用cli_allow_user()建立使用者帳戶，密碼預設是明文儲存，因此有些風險。

cli_set_auth_callback()會設置登入時的反應程式。

待續...