

UNIVERSITAT DE LLEIDA



Universitat de Lleida

GRAU EN ENGINYERIA INFORMÀTICA

XARXES

Pràctica 1, Programació d'aplicacions de xarxa

Autor:
Jordi Ricard Onrubia
Palacios

Professor:
Enric Guitart

17 d'abril de 2016

Resum

Índex

1	Client	1
1.1	Registrar-se al servidor	1
1.2	Manteniment de comunicació	2
1.3	Esperar comandes per l'enviament/recepció del seu arxiu de configuració	3
1.4	Funcions addicionals	3
2	Servidor	4
2.1	Enregistrament de clients	4
2.2	Manteniment de comunicació amb equips enregistrats	4
2.3	Enviament/recepció de arxius de configuració	4
2.4	Espera de comandes per consola	4
3	Anexe	5
3.1	Problemes i solucions trobats en el desenvolupament	5
4	Referències	6

1 Client

Programat en: Python versió 2.7.10.

El client és un programa què es comunica amb el servidor, inicialment, aquest demana permís per a connectar-se, un cop connectat mantindrà una comunicació temporitzada amb el servidor per tal d'informar de la seva presència i evitar la desconnexió amb el servidor.

A més a més, aquest podrà rebre una sèrie de comandes per tal d'enviar al servidor el seu arxiu de configuració o fer una petició per tal que el servidor li envii el seu arxiu de configuració.

A continuació s'expliquen les funcions que ha de realitzar i la seva resolució.

1.1 Registrar-se al servidor

Per a la realització del registre ens caldrà, primerament, crear un socket que utilitzarem per al transport de les PDUs (*Protocol Data Unit*), aquest socket utilitzarà el protocol UDP (*User Datagram Protocol*) per a la comunicació.

Les PDUs seran utilitzades per a l'enviament i recepció de paquets tant en el procés de registre com en el procés de manteniment de comunicació com es veurà més endavant. Per a la realització de la PDU utilitzarem la funció *struct* de Python amb el contingut següent:

tipus:	unsigned char	char	char	char	char
	Tipus Paquet	Nom Equip	Adreça MAC	Número Aleatori	Dades
bytes:	1	7	13	7	50

Taula 1: PDU Registre

Per a la fase de registre s'han predefinit uns paquets:

Valor	Mnemònic	Significat
0x00	REGISTER_REQ	Petició de registre
0x01	REGISTER_ACK	Acceptació de registre
0x02	REGISTER_NACK	Denegació de registre
0x03	REGISTER_REJ	Rebuig de registre
0x09	ERROR	Error de protocol

Taula 2: Paquets de Registre

Aquests paquets són únicament correctes en la fase de registre, qualsevol intent d'enviament d'aquests paquets en qualsevol altra fase donarà com a resultat una recepció d'un paquet d'error per part del servidor.

L'enviament de les PDUs de registre es farà temporitzat

1.2 Manteniment de comunicació

Valor	Mnemònic	Significat
0x10	ALIVE_REQ	Enviament d'informació ALIVE
0x11	ALIVE_ACK	Confirmació de recepció d'informació ALIVE
0x12	ALIVE_NACK	Denegació de recepció d'informació ALIVE
0x13	ALIVE_REJ	Rebuig de recepció d'informació ALIVE

Taula 3: Paquets de Informe ALIVE

1.3 Enviament/recepció del seu arxiu de configuració

Enviament

ENVIAMENT

tipus:	unsigned char	char	char	char	char
	Tipus Paquet	Nom Equip	Adreça MAC	Número Aleatori	Dades
bytes:	1	7	13	7	150

Taula 4: PDU Enviament i Recepció

Valor	Mnemònic	Significat
0x20	SEND_FILE	Petició d'enviament d'arxiu de configuració
0x21	SEND_ACK	Acceptació de la petició d'enviament d'arxiu de configuració
0x22	SEND_NACK	Denegació de la petició d'enviament d'arxiu de configuració
0x23	SEND_REJ	Rebuig de la petició d'enviament d'arxiu de configuració
0x24	SEND_DATA	Bloc de dades de l'arxiu de configuració
0x25	SEND_END	Fi de l'enviament de dades de l'arxiu de configuració

Taula 5: Paquets d'enviament d'arxiu de configuració

Recepció

RECEPCIO

1.4 Espera de comandes per consola

- quit:
- send-conf:
- get-conf:

Valor	Mnemònic	Significat
0x30	GET_FILE	Petició de recepció d'arxiu de configuració
0x31	GET_ACK	Acceptació de la petició de recepció d'arxiu de configuració
0x32	GET_NACK	Denegació de la petició de recepció d'arxiu de configuració
0x33	GET_REJ	Rebuig de la petició de recepció d'arxiu de configuració
0x34	GET_DATA	Bloc de dades de l'arxiu de configuració
0x35	GET_END	Fi de la recepció de dades de l'arxiu de configuració

Taula 6: Paquets de recepció d'arxiu de configuració

2 Servidor

Compilat en: Linux versió 4.2.0 gcc versió 5.2.1.

Comanda utilitzada: gcc -ansi -pedantic -Wall.

El servidor és un programa que permet respondre a les peticions d'un client i respondre en concordança.

En el nostre cas el servidor registrarà i mantindrà la comunicació amb els clients que estiguin autoritzats i que facin una petició de registre i que també mantinguin la comunicació per la seva part, a més a més, el servidor rebrà peticions d'enviament i recepció dels arxius de configuració dels clients i se'n carregarà de respondre a ells. Per part pròpia el servidor pot mostrar tots els equips autoritzats amb les seves dades i el seu estat si es demana.

2.1 Enregistrament de clients

2.2 Manteniment de comunicació amb equips enregistrats

2.3 Enviament/recepció de arxius de configuració

2.4 Espera de comandes per consola

- quit:
- list:

3 Anexe

3.1 Problemes i solucions trobats en el desenvolupament

Client

1. Bloqueig del programa per part del `recvfrom`, `recvfrom` és una funció què s'encarrega de rebre el missatge rebut pel socket, la solució aplicada va ser posar el timeout a 0 fent-la així no bloquejant.
2. Senyal `SIGTERM` ignorada, la solució aplicada va ser enviar-la continuament amb un bucle infinit.
3. Llançament de excepcions per part de Python a l'hora de detectar per teclear un `ctrl-c`, la solució aplicada va ser tractar la lectura del teclat amb un `try except` ignorant les interrupcions de teclat.

Servidor

1. Problema d'accedir i actualitzar les dades dels clients des d'altres processos. La solució ha estat fer un `mmap` utilitzant la llibreria `sys/mman.h`.
2. Si els dos primers "ALIVES" la temporització no arriba a agafar el tercer "ALIVE" en cas de què fos enviat, ja que aquest arriba just en el moment en què el servidor a desconnectat al client. La solució ha estat aplicar un temporitzador per tal de què el servidor només comprovi els estats dels clients cada segon en lloc de continuament, a més a més, es permet que el temps en què haguera d'arribar "l'ALIVE" sigui més gran que l'especificat, exactament 1 segon més.

4 Referències

Importar constants d'un altre fitxer en Python 2.

<http://zetcode.com/lang/python/packages/>

Structs en Python 2.

<https://docs.python.org/2/library/struct.html>

UDP sockets en Python 2.

<https://wiki.python.org/moin/UdpCommunication>

Funcions de la llibreria Time de Python 2.

<https://docs.python.org/2/library/time.html>

TCP sockets en Python 2.

<https://wiki.python.org/moin/TcpCommunication>

Signals en Python 2.

<https://docs.python.org/2/library/signal.html>

UDP sockets en C.

<https://www.cs.rutgers.edu/~pxk/417/notes/sockets/udp.html>

Compartir memòria en C.

<http://man7.org/linux/man-pages/man2/mmap.2.html>

Funcions de la llibreria Time de C.

<http://www.cplusplus.com/reference/ctime/>