

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	2
1.4	Funcions addicionals	2
2	Servidor	3
2.1	Enregistrament de clients	3
2.2	Manteniment de comunicació amb equips enregistrats	3
2.3	Enviament/recepció de arxius de configuració	3
2.4	Espera de comandes per consola	3
3	Anexe	3
3.1	Problemes i solucions trobats en el desenvolupament	3
4	Referències	4

1 Client

Programat en: Python versió 2.7.10.

El client és un programa què es comunicarà 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 seràn utilitzades per al 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 de enviament d'aquests paquets en qualsevol altre fase donarà com a resultat una recepció d'un paquet d'error per part del servidor.

1.2 Manteniment de comunicació

1.3 Esperar comandes per l'enviament/recepció del seu arxiu de configuració

1.4 Funcions addicionals

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 què permet respondre a les peticions de 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 de 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

3 Annexes

3.1 Problemes i solucions trobats en el desenvolupament

Client

Servidor

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