

**Pràctica 1:** *Aplicació client/servidor per a la implementació d'una eina per a la configuració d'un switch Gigabit Ethernet de sis ports.*

## **1 Objectiu**

En aquesta pràctica veureu les comunicacions entre dues aplicacions situades a màquines diferents utilitzant `sockets` `TCP`, orientats a connexió. Per això, desenvolupareu una parella d'aplicacions client/servidor en C.

## **2 Enunciat**

S'implementaran dues aplicacions, un client i un servidor per configurar, de forma remota, un *switch* `ethernet` a Gigabit de 6 ports.

Des de l'aplicació client podrem especificar l'adreça MAC de l'equip que volem connectar amb un cable `ethernet` cap a un dels ports del *switch*.

L'aplicació servidora centralitzarà les connexions que donem d'alta, i de baixa i generarà un fitxer de configuració amb les connexions que hem configurat en el *switch*.

Aquest fitxer de configuració el podrà utilitzar el tècnic que hagi de fer físicament les connexions dels diferents cables `ethernet` dels equips, al *switch* de l'armari de comunicacions.

A continuació es detalla la funcionalitat de l'aplicació.

## **3 Funcionalitat**

L'aplicació client mostra a l'administrador de sistemes un menú per poder gestionar la configuració dels sis ports del *switch* `ethernet`.

A través d'aquest menú l'usuari podrà fer les següents operacions:

**Hello** És una funcionalitat de test per veure si ens funciona bé la comunicació bidireccional entre les aplicacions client i servidor. El client envia un missatge de tipus `HELLO` i en rebre la resposta del servidor, mostra per pantalla un dels camps del missatge.

**Mostrar la configuració actual** Mostra l'estat actual de configuració del *switch* tot indicant, per a cada port del *switch*, si està ocupat, i, en aquest cas, amb quina MAC, o bé si està lliure.

Un possible resultat de l'execució d'aquesta opció de menú seria:

Configuració del switch 1000BaseT SW-GB-101-1:

```
IF0-> 00:00:00:00:00:11
IF1-> 00:00:00:00:11:22
IF2-> lliure
IF3-> 00:00:00:11:22:33
IF4-> lliure
IF5-> lliure
```

Noteu les IF, interfaces, és com s'anomenen cadascun dels ports del *switch*.

**Connectar-se a un port concret del *switch*** L'usuari especificarà quin port vol utilitzar del *switch* i quina MAC *address* hi vol associar.

En cas que el port estigui ocupat, el servidor enviarà un missatge d'error.

En cas que el port estigui lliure, el servidor enviarà un missatge de OK.

**Connectar-se a un port lliure del *switch*** L'usuari especificarà quina adreça MAC vol associar a qualsevol port lliure. El servidor informará al client de a quin port s'ha fet l'associació. En cas que no hi hagi cap port lliure el servidor enviarà un missatge d'error.

**Alliberar un port del *switch*** L'usuari especificarà quin port del *switch* vol alliberar.

El servidor informará a l'usuari si s'ha pogut fer l'acció. En cas que el port especificat ja estigui lliure, el servidor enviarà un missatge d'error.

**Alliberar tots els ports del *switch*** El servidor alliberarà tots els ports del *switch* que estiguin ocupats.

El servidor ha d'informar si s'ha pogut fer l'acció.

**Generar el fitxer de configuració del *switch*** Quan el client especifiqui aquesta opció de menú el servidor generarà un fitxer text, `nom_del_switch.conf`, amb la informació de

les adreces MAC associades a cada port. Aquest fitxer és el que utilitzarà el tècnic de comunicacions per configurar físicament el *switch*.

Aquest podria ser un exemple de fitxer de configuració generat, `SW-GB-101-1.conf`, associat a la configuració llistada de l'exemple anterior:

```
IF0 00:00:00:00:00:11
IF1 00:00:00:00:11:22
IF2 empty
IF3 00:00:00:11:22:33
IF4 empty
IF5 empty
```

Com veieu, per cada un dels ports es genera una fila amb l'identificador del port, IF, i l'adreça MAC associada.

Noteu que els ports que es poden configurar estan identificats com: IF0, IF1, IF2, IF3, IF4, i IF5.

El servidor informarà a l'usuari si s'ha pogut fer l'acció.

En les següents seccions us detallem el funcionament del programa client i del programa servidor.

## 4 Client de configuració del *switch*

L'administrador de sistemes executarà l'aplicació client per configurar el *switch*. Se li mostrarà un menú amb totes les operacions disponibles.

A través de la línia de comandes, l'administrador indicarà quina opció de menú ha triat.

Per llençar l'aplicació client la sintaxi serà la següent:

```
./swConfClient -h host [-p port]
```

on

**-h host** És el host on s'està executant el servidor, p.e: `deic-dc23.uab.es` o bé `158.109.70.227`.

**-p port** És el port a on s'està executant el servidor. Aquest paràmetre és opcional, si no s'especifica pren un valor per defecte.

Si executem el client amb la crida:

```
# ./swConfClient -h deic-dc15.uab.es -p 8311
```

Les operacions que hauria de fer el client serien:

1. Li envia un missatge al servidor per a que, aquest, li enviï les dades de les connexions que hi han en els ports del *switch*.
2. Espera la resposta del servidor.  
El servidor en rebre aquest missatge li envia al client un missatge amb les dades de les connexions d'aquest *switch*.
3. Li mostra a l'administrador les connexions del *switch*.
4. Li mostra a l'administrador el menú amb les diferents operacions que pot fer sobre el *switch*.

Us mostrem un exemple del que l'aplicació client mostraria a l'administrador:

Configuració del switch 1000BaseT SW-GB-101-1:

```
IF0-> 00:00:00:00:00:11
IF1-> 00:00:00:00:11:22
IF2-> lliure
IF3-> 00:00:00:11:22:33
IF4-> lliure
IF5-> lliure
```

Operacions de configuració:

```
0.Hello
1.Connectar a un port
2.Connectar a qualsevol port
3.Alliberar la connexió d'un port
4.Alliberar totes les connexions
5.Generar un fitxer amb la configuració actual.
6.Sortir
```

Opció? 1

MAC a connectar? 00:01:02:03:04:05

Port a on connectar-la? 1

Un cop l'usuari selecciona una opció de menú, se li envia un missatge al servidor amb l'opció de menú seleccionada, i si s'escau, els seus paràmetres, i s'espera una resposta del servidor.

Si cal, el client mostra per pantalla la resposta del servidor.

A continuació torna a mostrar el llistat de les connexions actuals i el menú per a que l'administrador pugui continuar configurant el switch.

En la següent secció indicarem com funcionarà el programa servidor.

## 5 Servidor de configuració del *switch*

Aquesta és la funcionalitat que ha d'implementar el vostre servidor:

1. En executar-se, el servidor crea un array en memòria de sis posicions, corresponents als sis ports del *switch*. En cada una d'aquestes posicions guardarà una cadena de 18 bytes amb l'adreça MAC de l'equip que es connecti al port associat.

La cadena de la MAC és de 18 bytes ja que també guardem el caracter de finalització de cadena '\0'.

Inicialment aquest array està buit.

El codi per muntar aquest array en memòria us el donarem fet.

2. A mesura que l'administrador va creant connexions, va omplint cada una de les posicions de l'array amb l'adreça MAC de l'aparell que connecta a un port concret.
3. El servidor ha de tenir un **socket** escoltant per un port les peticions que li arriben dels diferents clients. Un cop hi hagi una connexió amb un client, el servidor s'encarregarà d'atendre les comandes que li enviï el client, executar-les i tornar-li una resposta.
4. Un cop el client enviï el missatge de `sortir` de l'aplicació, el servidor desarà la configuració actual en un fitxer i tancarà la connexió establerta amb aquest client.

El servidor s'invocarà de la següent manera:

```
./swConfServer [-p port]
```

on

**-p port** Indica el port a on s'està executant el servidor. Aquest paràmetre és opcional. Si no s'especifica pren un valor per defecte seguint les indicacions descrites més a vall.

El servidor ha d'utilitzar un port conegut per a que el client el pugui localitzar. Per aquest motiu tindrem un port per defecte, que calcularem segons la següent fórmula:  $8000 + (100 * Grup) + Subgrup$  (Grup A  $\rightarrow$  1, B  $\rightarrow$  2, C  $\rightarrow$  3,...). Exemple: grup fx-c11, grup C és 3, port  $8000 + 100 * 3 + 11 = 8311$ .

D'altra banda, el port per defecte es pot canviar al executar el servidor, utilitzant el paràmetre `-p`. En aquest cas, el servidor farà servir el port especificat en comptes del port per defecte, i el client s'haurà d'invocar, per tant, amb el mateix port.

## 6 Com generar els executables: `swConfClient` i `swConfServer`

Per a generar els fitxers que es puguin executar: `swConfClient` i `swConfServer`, cal que compilem el codi i generem els executables.

Per compilar el codi utilitzarem l'eina `make`.

Aquesta eina utilitza un fitxer de configuració: `Makefile`, on hi han els paràmetres de configuració per a la compilació i generació dels executables, tant del client com del servidor.

Aquest fitxer ja el tindreu en el vostre compte de pràctiques, en el directori de la pràctica.

Per a generar els fitxers `swConfServer` i `swConfClient`, servidor i client respectivament, executarem la següent comanda:

```
$ make
```

## 7 Concurrència

Quan el servidor rep una connexió d'un client, passa a executar les comandes que el client li demana. Si mentrestant altres clients s'hi volen connectar, els encua fins que no hagi acabat amb la comanda que està executant actualment. Tenim doncs un servidor **iteratiu**.

Imagineu-vos el següent context. El vostre servidor administra tots els *switchs* dels armaris de comunicacions de la vostra empresa. En aquesta situació té sentit pensar en que poden haver diferents administradors de sistema que es vulguin connectar al servidor per gestionar diferents *switchs*.

En aquest context, doncs, ens caldria tenir un **servidor concurrent** per a fer la tasca el més eficient possible.

Caldrà, doncs, que modifiqueu el servidor per a que sigui **concurrent**. D'aquesta manera, el servidor podrà atendre a la vegada totes les diferents operacions que li demanin els diferents administradors que s'hi han connectat. Penseu, que en el nostre cas, seran diferents administradors que administraran un *switch*.

Per a fer-ho cal que:

1. El servidor tingui un **socket** permanentment escoltant per un port possibles connexions entrants.
2. En rebre una connexió, el servidor ha de crear un altre **socket**, associat a un altre port diferent per atendre les possibles operacions que ens enviï el client i enviar-li les respostes. Quan el client surti de l'aplicació el servidor tancarà aquest **socket**.

## 8 Format dels missatges

Entre l'aplicació client i l'aplicació servidora es manté un enviament bidireccional de missatges. És a dir, implementen un protocol de comunicació. A continuació us especificuem quin ha de ser el format dels missatges. **El fet que no seguiu aquest format implicarà que les vostres aplicacions client/servidor no seran compatibles entre els altres grups de pràctiques i per tant hi haurà una penalització en la nota final.**

Type	Opcode		
	2 bytes		
	-----		
HELLO	01		
	-----		
	2 bytes	12 bytes	
	-----		
HELLO_RP	02	Hello World	
	-----		
	2 bytes		
	-----		
LIST	03		
	-----		
	2 bytes	n bytes	
	-----		
		18 bytes	
		-----	
LIST_RP	04	[ MAC ] *6	
		-----	
	-----		
	2 bytes	2 bytes	18 bytes
	-----		
CONN_TO	05	port	MAC
	-----		
	2 bytes	18 bytes	
	-----		
CONN_ANY	06	MAC	



	2 bytes	2 bytes
CONN_ANY_RP	07	port
	2 bytes	2 bytes
FREE	08	port
	2 bytes	
FREE_ALL	09	
	2 bytes	
GEN_FILE	10	
	2 bytes	
FINISH	11	
	2 bytes	
OP_OK	12	
	2 bytes	2 bytes
OP_ERR	13	error_code

A continuació us mostrem una taula amb el significat de cada un dels missatges:

<b>codi</b>	<b>Tipus</b>	<b>Missatge</b>
1	HELLO	Hello Request: Petició de <i>Hello World</i> .
2	HELLO_RP	Hello Response: Resposta a la petició HELLO. <b>Camps:</b> * Missatge " <i>Hello World</i> \0"
3	LIST	List Request: Petició per llistar totes les connexions del switch.
4	LIST_RP	List Response: Resposta a la petició de llistat. <b>Camps:</b> * Cadena de 18 caracters amb la MAC associada a cada port. Aquest camps apareixen tants cops com ports tenim, 6. L'ordre d'aparició de cada MAC correspon al port al que està associada. En cas que en un port no hi hagi connectat cap equip, el camp de la MAC estarà ple amb 18 zeros. Exemple: 04 00:01:02:03:04:05 000000000000000000 A4:31:B4:AC:33:BB ... En aquest cas en el primer i tercer port hi ha connectat un equip, però en el segon port no hi cap equip connectat.
5	CONN_TO	Connect to: Petició per connectar un equip a un port concret. <b>Camps:</b> * Port a on volem connectar un equip. * Adreça MAC de l'equip que volem connectar a un port concret.
6	CONN_ANY	Connect to any port: Petició per connectar un equip a qualsevol port lliure. <b>Camps:</b> * Adreça MAC de l'equip que volem connectar a qualsevol port lliure.
6	CONN_ANY_RP	Connect to any response: Missatge de resposta del servidor a la petició de connectar un equip a qualsevol port lliure. <b>Camps:</b> * Port a on hem connectat l'equip.
8	FREE	Petició per disconnectar un equip d'un port. <b>Camps:</b> * Port que volem alliberar.
8	FREE_ALL	Petició per disconnectar tots els equips dels ports.
9	GEN_FILE	Petició per generar el fitxer de configuració amb les diferents connexions que hem configurat.
10	FINISH	Missatge per indicar que l'usuari ha sortit.
11	OP_OK	Missatge per indicar que l'operació s'ha dut a terme amb èxit.
12	OP_ERR	Missatge per indicar que s'ha produït un error. <b>Camps:</b> * El codi d'error del missatge.

Codis d'error (err\_code):

codi	Error
1	Port ocupat
2	No hi han ports lliures
3	Port lliure
4	Port fora de rang
5	Format incorrecte de l'adreça MAC

## 9 Observacions

- **Control:** La setmana del **26 d'octubre**, al principi de la sessió, es farà el control d'una part de la funcionalitat. Aquest control està valorat en **0.5 punts** i és opcional. En cas que no feu el control només podreu optar a un 9.5 de nota de pràctiques.

## 10 Proposta de planificació de la pràctica per setmanes

Setembre	27	Implementar els missatges <code>HELLO</code> i <code>HELLORP</code> . El client mostra per pantalla el camp "Hello world". Implementar els missatges <code>LIST</code> i <code>LIST_RP</code> .
Octubre	26	<b>Control</b> de la funcionalitat anterior. Modificar el codi per a que el servidor sigui concurrent. Implementar la funcionalitat restant.
Novembre	8	<b>Entrega de la pràctica <i>offline</i>.</b>

## 11 Condicions de lliurament

Llegiu-vos atentament l'enunciat abans del lliurament i verifiqueu que la vostra pràctica té implementat tot el que es demana.

El lliurament de la pràctica es farà en un únic dia per tots els grups, serà el **08-11-20** a les 12:00h es copiarà la vostra carpeta de codi:

`$HOME/practiques/sockets/skel`