

1.- Desarrolle un programa Shell que permita remover todos los archivos de extensión .o del directorio donde se ejecuta su programa Shell script y de todos sus subdirectorios.

Mi primera solución:

Asumo que a este script es llamado p1.sh <dir a examinar>

```
#!/bin/bash
recursivo.sh `pwd` $1
```

En archivo recursivo.sh tenemos:

```
#!/bin/bash
cd $2
for i in *.o
do
    if test -f $i
    then rm $i
    fi
done
for i in *
do
    if test -d $i
    then $1/recursivo.sh $1 $i
    fi
done
```

Otras soluciones inspiradas en sus soluciones:

```
#!/bin/bash
rm -f `find ./ -type f -name |*.o`
```

Otra solución:

```
#!/bin/bash
function todo() {
rm $1/*.o
for i in $1/*
do
if test -d $i
then
todo $i
fi
done
}
todo `pwd`
```

2.- Qué cambio le haría al servidor eco TCP simple que se lista para que haga un eco de todo lo que le llegue desde el cliente siempre y cuando el cliente provenga de la IP 200.1.17.73. En los otros casos la petición no es procesada.

```
int main(int argc, char * argv[])
{
char buf[20];
```

```

int s, n, ns, len;
struct sockaddr_in name;
s = socket(AF_INET, SOCK_STREAM, 0);
name.sin_family = AF_INET;
name.sin_port = htons(atoi(argv[1]));
name.sin_addr.s_addr = htonl(INADDR_ANY); /* Use the wildcard address.*/
len = sizeof(struct sockaddr_in);
bind(s, (struct sockaddr *) &name, len);
listen(s, 4);
ns = accept(s, (struct sockaddr *) &name, &len);

if (name.sin_addr.s_addr == inet_addr("200.1.17.73"))
    while ((n = recv(ns, buf, sizeof(buf), 0)) > 0)
        send(ns, buf, n, 0);

close(ns);
close(s);
exit(0);
}

```

3.- En una red no conmutada (con un hub) hay dos máquinas (A y B) generando y recibiendo tráfico multicast con TTL=2. A envía audio y B video. Ambas tiene *loopback* activado.

En cada caso indique si la máquina A recibe audio, video o ambos en su interfaz de red (capa física), su capa internet, su capa transporte, su capa aplicación.

Caso a) La máquina A recibe y envía al puerto 2345, IP 235.5.5.5

La máquina B recibe y envía al puerto 2345, IP 236.5.5.5

A: recibe tráfico de audio en todas sus capas, además recibe video en su capa Internet donde es filtrada. El video no llega a la aplicación.

Caso b) La máquina A recibe y envía al puerto 2345, IP 235.5.5.5

La máquina B recibe y envía al puerto 2346, IP 235.5.5.5

A recibe audio como en caso a), además recibe video hasta su capa transporte. En esta capa el video es filtrado.

Caso c) La máquina A recibe y envía al puerto 2345, IP 235.5.5.5

La máquina B recibe y envía al puerto 2345, IP 235.5.5.6

La máquina recibe audio como es los casos anteriores, el video es filtrado por su capa MAC. No llega a capa Internet o Transporte.

4.- a) Indique una ventaja y una desventaja de usar memoria compartida frente a usar FIFOs.

Ventaja: La velocidad porque evitamos copias múltiples.

Desventaja: Se requiere algún mecanismo de sincronización para acceder a los datos compartidos.

b) ¿Qué significa el permiso s en las siguientes líneas?

```
agustin@agv2002:/bin$ ls -l ping
```

```
-rwsr-xr-x 1 root root 28520 Apr 22 01:29 ping
```

Significa que los permisos efectivos durante la ejecución del programa serán los de root. Es decir, mientras se ejecuta el programa el usuario tiene los mismos accesos que el superusuario.

c) ¿Qué significa el permiso x en las siguientes líneas?

```
agustin@agv2002:~$ ls -l |grep tmp
```

```
drwx----- 8 agustin users 408 Sep 24 18:11 tmp
```

Significa que el usuario dueño del directorio puede acceder a este vía comando cd por ejemplo.

5.- Normalmente cuando usted presiona las teclas control y luego C, (Control-C), generalmente los programas terminan su ejecución. ¿Qué código debe ser incorporados a un programa si no queremos que esta condición sea ignorada y el programa no termine?

Lo que debemos hacer es incorporar una rutina que maneje la llegada de la señal correspondiente al Control-C.

Para ello definimos una nueva función:

```
void muda(int i)
```

```
{  
}
```

Luego tan pronto se inicia el main, ponemos:

```
signal(SIGINT, muda);
```

Alternativamente podemos ignorar la señal poniendo en el main y como primera cosa:

```
signal(SIGINT, SIG_IGN);
```