

## Práctica 7

### Programando en Java

#### 1. Ficheros de trazas

Para la realización de esta práctica disponemos de varios ficheros de trazas para probar. Cada fichero contiene los datos de una trama en cada línea. Cada uno de los datos de la trama se separa mediante un único espacio.

Cada fichero tiene un formato distinto, pero todos están ordenados de forma creciente por el campo de tiempo.

##### captura\_1.txt

```
<tiempo> <tamaño de la trama> <jerarquía del protocolo> <nombre del protocolo> <IP origen> <IP destino>
```

##### captura\_2.txt

```
<tiempo> <tamaño de la trama> <nombre del protocolo> <jerarquía del protocolo> <IP origen> <IP destino>
```

##### captura\_3.txt

```
<tiempo> <tamaño de la trama> <MAC origen> <MAC destino> <jerarquía del protocolo> <nombre del protocolo>
```

##### captura\_4.txt

```
<tiempo> <tamaño de la trama> <nombre del protocolo> <jerarquía del protocolo> <MAC origen> <MAC destino>
```

#### 2. Tamaño de la captura (0,10 puntos)

Realizar un programa en Java que lea un fichero y muestre la cantidad total de bytes que se han transferido.

El programa deberá poder ejecutarse de esta forma:

```
$ java TotalSize <nombre del fichero>
```

donde:

- <nombre del fichero> es el nombre del fichero con trazas a usar

Ejemplos (puedes probar con cualquiera de los 4 ficheros de trazas):

```
$ java TotalSize captura_1.txt
captura_1.txt 210626
$ java TotalSize captura_2.txt
captura_2.txt 156092
$ java TotalSize captura_3.txt
captura_3.txt 175538
$ java TotalSize captura_4.txt
captura_4.txt 236062
```

### 3. Tamaño por intervalos (0,30 puntos)

Realizar un programa que lea la lista de tramas de un fichero dado y muestre la cantidad de bytes que se han transmitido en cada intervalo de tiempo según la duración indicada. Se recomienda hacer primero el punto 2.

El programa deberá poder ejecutarse de esta forma:

```
$ java IntervalSize <nombre del fichero> <duración del intervalo>
```

donde:

- <nombre del fichero> es el nombre del fichero con trazas a usar
- <duración del intervalo> número decimal con el intervalo en segundos

El programa leerá los datos de <nombre del fichero> y agrupará las tramas que aparezcan en cada intervalo de duración indicado. El resultado final a mostrar será una línea por intervalo con el siguiente formato:

```
<tiempo final del intervalo> <bytes del intervalo> <bytes totales hasta ahora>
```

**AYUDA:** como los ficheros están ordenados por tiempo para ir contando los bytes en cada intervalo, bastará con saber cuándo acaba el intervalo actual e ir sumando bytes hasta ese momento. Una vez superado el intervalo, podemos mostrar la línea, poner a cero el contador y buscar el siguiente intervalo.

### Ejemplos:

```
$ java IntervalSize captura_1.txt 0.02
0.020000 812 812
0.040000 104 916
0.080000 314 1230
(...)
6.860000 2951 210205
6.880000 66 210271
6.900000 355 210626
```

```
$ java IntervalSize captura_2.txt 0.02
0.020000 528 528
0.380000 1330 1858
0.560000 60 1918
(...)
5.380000 839 146243
5.440000 60 146303
5.800000 9789 156092
```

```
$ java IntervalSize captura_3.txt 0.02
0.020000 66 66
0.340000 66 132
0.600000 477 609
0.620000 1550 2159
(...)
8.460000 370 174921
8.480000 384 175305
8.520000 167 175472
8.640000 66 175538
```

```
$ java IntervalSize captura_4.txt 0.02
0.020000 893 893
0.040000 198 1091
0.220000 60 1151
(...)
11.760000 107 235846
11.780000 123 235969
12.100000 93 236062
```

## 4. Filtrando por MAC/IP en intervalos (0,10 puntos)

Realizar un programa que, basándonos en el punto anterior en el que hemos filtrado por intervalos, podamos indicar una MAC o una IP origen y un protocolo. Mostraremos la misma información que en el apartado anterior pero únicamente de aquellas trazas que cumpla con esa MAC/IP origen y protocolo. Únicamente es necesario realizar un programa, bien para MAC o bien para IP.

Si elegimos un filtrado por MAC, usaremos el fichero `captura_3.txt` y ejecutaremos de la siguiente forma:

```
$ java MacSize <nombre del fichero> <duración del intervalo> <dirección  
MAC> <protocolo>
```

donde:

- `<nombre del fichero>` es el nombre del fichero con trazas a usar
- `<duración del intervalo>` número decimal con el intervalo en segundos
- `<dirección MAC>` dirección MAC (columna 3 o columna 4)
- `<protocolo>` nombre del protocolo (columna 6)

Si elegimos un filtrado por IP, usaremos el fichero `captura_1.txt` y ejecutaremos de la siguiente forma:

```
$ java IpSize <nombre del fichero> <duración del intervalo> <dirección  
IP> <protocolo>
```

donde:

- `<nombre del fichero>` es el nombre del fichero con trazas a usar
- `<duración del intervalo>` número decimal con el intervalo en segundos
- `<dirección IP>` dirección IP (columna 5 o columna 6)
- `<protocolo>` nombre del protocolo (columna 4)

Ejemplos:

```
$ java MacSize captura_3.txt 0.02 bc:14:01:0b:5d:72 TCP  
0.020000 66 66  
0.340000 66 132  
0.600000 0 132  
0.620000 0 132  
(...)  
8.460000 132 29010  
8.480000 66 29076  
8.520000 66 29142  
8.640000 66 29208
```

```
$ java IpSize captura_1.txt 0.02 192.168.1.10 TCP  
0.020000 0 0  
0.040000 0 0  
0.080000 0 0  
0.100000 66 66  
(...)  
6.800000 108 52184  
6.860000 1616 53800  
6.880000 66 53866  
6.900000 132 53998
```

## 5. Algunas ayudas

Para comparar una cadena de texto (String) con otra, puedes usar este método de un objeto de tipo String:

```
public boolean equals(Object anObject)
```

más info: [http://docs.oracle.com/javase/6/docs/api/java/lang/String.html#equals\(java.lang.Object\)](http://docs.oracle.com/javase/6/docs/api/java/lang/String.html#equals(java.lang.Object))

Si tenemos una línea de texto con columnas de datos separados por espacios, una de las formas de obtener cada columna es usar la clase StringTokenizer:

```
StringTokenizer tokenizer = new StringTokenizer("this is a test");  
tokenizer.nextToken();
```

Operador y lógico; cuando se deben cumplir dos (o más) condiciones: &&

```
String ip = st.nextToken();  
String protocolo = st.nextToken();  
// Cuando la IP sea 10.1.1.1 y el protocolo TCP  
if(ip.equals("10.1.1.1") && protocolo.equals("TCP")) {  
    // Más código  
}
```

## 6. Cómo entregar la práctica

Para entregar la práctica es necesario subir el fichero .java que habéis creado en el apartado correspondiente en la web de la asignatura.

Verás que hay disponibles 4 posibilidades:

- TotalSize: para el apartado 2
- IntervalSize: para el apartado 3
- MacSize o IpSize: para el apartado 4