

Reporte técnico scripts

Taller de Sistemas Operativos
Escuela de Ingeniería Informática

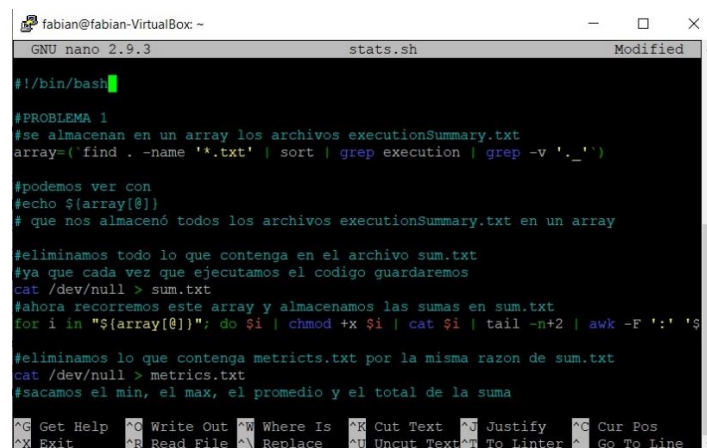
Fabián Rozas Alfaro

fabian.rozas@alumnos.uv.cl

Resumen. Como objetivo principal del trabajo es realizar una implementación de scripts para procesar resultados de programas basados en lo que pide el Taller 01. En este primer informe se dará a conocer un reporte técnico. Este reporte se basa en la descripción del problema y el diseño de la solución en el cual nos basaremos para la posterior implementación.

1. Introducción

Un scrip es un archivo que contiene un conjunto de órdenes para realizar una acción. Para generar un scrip se utilizan lenguajes interpretados, en el cual se puede escribir el código que compondrá el script que no será más que un fichero de órdenes o procesamiento por lotes [1]. En este caso ocuparemos como leguaje interpretado el propio BASH (Bourne-again shell), que básicamente es un lenguaje de comandos y shell de Unix. En otras palabras es un procesador de comandos que generalmente se ejecuta en una ventana de texto como lo muestra la Figura 1, donde el usuario escribe comandos que causan acciones. Dicho anteriormente este también puede leer y ejecutar comandos desde un archivo, llamado script de shell [2].



```
fabian@fabian-VirtualBox: ~
GNU nano 2.9.3 stats.sh Modified
#!/bin/bash

#PROBLEMA 1
#se almacenan en un array los archivos executionSummary.txt
array=( `find . -name '*.txt' | sort | grep execution | grep -v '._'` )

#podemos ver con
#echo ${array[@]}
# que nos almacenó todos los archivos executionSummary.txt en un array

#eliminamos todo lo que contenga en el archivo sum.txt
#ya que cada vez que ejecutamos el codigo guardaremos
cat /dev/null > sum.txt
#ahora recorremos este array y almacenamos las sumas en sum.txt
for i in "${array[@]"; do $i | chmod +x $i | cat $i | tail -n+2 | awk -F ':' '{s
#eliminamos lo que contenga metrics.txt por la misma razon de sum.txt
cat /dev/null > metrics.txt
#sacamos el min, el max, el promedio y el total de la suma

^G Get Help ^C Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^N Replace ^U Uncut Text ^L To Linter ^_ Go To Line
```

Figura 1. Ventana de texto con comandos

Los script tienen encabezamientos conocidos como shebang según el intérprete usado, que le indica al sistema cual ocuparemos. En este caso estamos trabajando con Linux, y ocuparemos el intérprete de Bash. El shebang que se utiliza para este intérprete es `#!/bin/bash` (Figura2), pero en cualquier otro caso será el binario que apunta hacia el intérprete o shell. Además, el fichero de script suele tener la extensión “.sh”. [3]

```
#!/bin/bash
#PROBLEMA 1
#se almacenan en un array los archivos executionSummary.txt
array=(`find . -name '*.txt' | sort | grep execution | grep -v '._'`)
```

Figura 2

2. Descripción del problema

2.1 Contexto de los datos a utilizar

Estos datos corresponden a estadísticas que provienen de cierta cantidad de experimentos de simulación de un sistema de evacuación de personas. Cada experimento es una simulación de una evacuación costera de la ciudad de Iquique ante un eventual tsunami, que considera 75000 personas. Existen tres modelos de personas: Residentes, Visitantes Tipo 1 y Visitantes Tipo II (Tabla 1). Todos los residentes conocen su zona de seguridad. Los visitantes, al inicio del proceso de evacuación, no. Eventualmente, un visitante puede pedir ayuda y conocer su zona de seguridad (Visitante tipo I). Si un visitante no logra determinar su zona de seguridad, queda caminando por la ciudad y no llega nunca a una zona de seguridad (Visitante tipo II).

<i>Tipo de Persona</i>	<i>Descripción</i>
Residente	Persona que vive en la ciudad y conoce su zona segura
Visitante Tipo I	Visitante que durante la evacuación logra determinar su zona segura
Visitante Tipo II	Visitante que durante la evacuación logra no determinar su zona segura

Tabla 1. Tipos de personas simuladas

Además de la clasificación mencionada, cada persona pertenece a un grupo etario, los que se resumen en la Tabla 2.

<i>Grupo Etario</i>	<i>Intervalo de edad</i>
G0	0-14
G1	15-29
G2	30-64
G3	65 o más

Tabla 2. Grupos etarios

2.2 Organización de los datos

Cada simulación entrega los resultados ordenados en una estructura de directorios, según el esquema que se visualiza en la Figura 3, donde NNN es el identificador de la simulación, representado por un código numérico de tres dígitos.

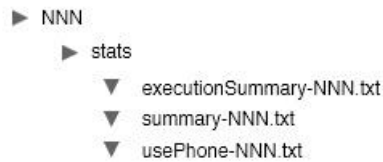


Figura 3. Estructura de directorios de los resultados de cada simulación

2.3 Descripción de los datos

2.3.1 Archivo executionSummary.txt

El archivo executionSummary-NNN.txt contiene datos sobre el desempeño de la simulación NNN. Posee una cabecera con campos del nombre que representa cada dato, y una segunda línea con los datos. Se compone por 10 campos, separados por el símbolo ':' como se muestra en la Figura 4. También en la Tabla 3 se detallan los campos y su descripción.

```
numExperiment:tsim:calibrationTime:Residents:Visitors:timeExecMakeAgents:timeExecCal:timeExecSim:maxMemory:agentsMem  
0:3600:100:69000:6000:28252:33157:182800:288236:33648
```

Figura 4. Estructura del archivo executionSummary.txt

<i>Campo</i>	<i>Descripción</i>	<i>Ejemplo</i>
numExperiment	Número del experimento.	0 (Corresponde al código 000)
Tsim	Tiempo de simulacion	3600: segundos de tiempo real que se simuló.
CalibrationTime	Tiempo de calibración del simulador	100: segundo de tiempo de real que se destina a la calibración inicial de las personas.
Residents	Cantidad de Residentes simulados	69000
Visitors	Cantidad de Visitantes simulados	6000

timeExecMakeAgents	Tiempo real que demora en crear en memoria las personas simuladas	28252: milisegundos
timeExecCal	Tiempo real que demora la calibracion de las personas	33157: milisegundos
timeExecSim	Tiempo real que toma la simulación	182800: milisegundos
maxMemory	Costo espacial del simulador	288236: Kbytes
agentsMem	Memoria utilizada por las estructuras de datos relacionados con las personas	33648: KBytes

<i>Campo</i>	<i>Descripción</i>	<i>Ejemplo</i>
numExperiment	Número del experimento.	0 (Corresponde al código 000)
id	Identificador de la persona simulada	40
model	Identificar del modelo de la persona 0: Residente 1: Visitante tipo I 2: Visitante tipo II	0
gruoupAge	Identificador del grupo etario de la persona 0: G1, 1: G2, 2: G3, 3: G4	3
safeZone	Identificador de la zona segura de la persona Z1, Z2, Z3, Z4, Z5 : Identificadores de zonas seguras NA: la persona no tiene zona segura asignada	Z1
distanceToTargetPos	Distancia a la que quedó la persona de su objetivo inicial	197.678312 metros
responseTime	Número aleatorio que representa cuánto tiempo se demoró la persona en tomar la decisión de evacuar, desde que dieron la orden de evacuar. Si es -1, entonces la persona nunca pudo evacuar.	323.091362 segundos
evacTime	Tiempo que la persona se demoró en llegar a la zona de evacuación. Si es 0, entonces la persona nunca llegó a la zona de evacuación.	3350 segundos

Tabla 4 Descripción de los campos del archivo summary.txt

2.3.3 Archivo usePhone.txt

El archivo usePhone-NNN.txt contiene datos sobre el comportamiento de las personas en la simulación NNN, con respecto al uso de un teléfono móvil. Posee una cabecera y cada línea posterior, representa la cantidad de personas que utilizaron el teléfono en cierto instante de tiempo. Cada una tiene tres campos, separados por el símbolo ‘:’. En la Figura 6 se muestra el contenido del archivo que corresponde a la simulación 000. En la Tabla 4 se detallan los campos y su respectiva descripción.

```
|usePhone
0:0:0
0:10:4333
0:20:4383
0:30:4261
0:40:4410
0:50:4388
0:60:4381
0:70:4440
0:80:4367
0:90:4506
0:100:4436
0:110:4364
0:120:4542
0:130:4368
0:140:4394
0:150:4426
0:160:4405
0:170:4504
0:180:4459
0:190:4410
0:200:4423
0:210:4422
0:220:4273
0:230:4242
0:240:4266
0:250:4114
0:260:4018
0:270:4143
0:280:3949
0:290:3951
0:300:3791
0:310:3686
0:320:3611
0:330:3431
0:340:3427
0:350:3365
0:360:3292
0:370:3142
0:380:3020
0:390:2825
0:400:2844
```

Figura 6 Estructura archivo usePhone.txt

<i>Campo</i>	<i>Descripción</i>	<i>Ejemplo</i>
numExperiment	Número del experimento	0 (Corresponde al código 000)
timeStamp	Tiempo de la medición	30
usePhone	Cantidad de persona que utilizaron el teléfono móvil en el tiempo especificado	4261

Tabla 5 Descripción de los campos del archivo usePhone.txt.

3. Diseño de la solución

Se utilizó el mismo diseño para el problema 1, 2 y 3. Debido que nos piden buscar el total, promedio, max y el min en cada uno de ellos.

Primero que todo, se deben de buscar los archivos descargados de cada simulación y almacenarlos en tres array distintos (uno para executionSummary-NNN.txt, otro para summary-NNN.txt, y otro para usePhone-NNN.txt). Después se recorren los array, buscando y realizando la sumas, o extracción de datos respectivos.

En el primer problema se realizan las sumas de las columnas o campos de timeExecMakeAgents + timeExecCal + timeExecSim en el array de executionSummary y se almacenan en un nuevo archivo. Estas sumas representan el tiempo de simulación total para cada executionSummary-NNN.txt. Luego para sacar el min, el max y el promedio visualizamos el tiempo de simulación total de cada una de las simulaciones almacenadas en el nuevo archivo. Finalmente al obtener el min, max, promedio y la suma total del tiempo de todas las simulaciones, se almacenan en un archivo llamado metrics.txt y luego se elimina el archivo intermedio que almacenó el tiempo de simulación.

Ahora para sacar el min, max, promedio y suma total de la memoria utilizada por las simulaciones, se utiliza el mismo diseño anterior, buscando en el array la columna o campo que representa la memoria utilizada y realizando las sumas que correspondan a dicha memoria para almacenarlas en un nuevo archivo. Luego calcular el promedio, max, min en dicho archivo y almacenar los resultados en el archivo metrics.txt. Sin olvidar eliminar los archivos intermedios. Finalmente, quedando en el archivo metrics.txt, el tiempo total de todas las simulaciones, con el promedio, min y max de dichos tiempos. También junto con esto en el archivo queda, la suma de la memoria total usada por las simulaciones, seguido del promedio, min y max de dichas memorias como se muestra en la Figura 7.

```
tsimTotal:promedio:min:max  
memUsed:promedio:min:max
```

Figura 7

En el segundo problema se debe calcular las sumas del tiempo de evacuación de todas las personas simuladas. Dicho campo se encuentra en el archivo de summary, que como dicho anteriormente se buscaron y almacenaron todos los archivos summary-NNN.txt en un array. Se extraen el campo necesario para realizar las sumas del tiempo de evacuación y se almacenan en un nuevo archivo. Luego para calcular el total, promedio, min y max se debe entrar en el nuevo archivo, realizar las operaciones correspondientes y almacenar los resultados en evacuation.txt. Luego se eliminan los archivos intermedios. Así mismo se deben de encontrar el total, promedio, min y max del tiempo de evacuación pero de los siguientes grupos o tipos de personas simuladas. También almacenando el resultado en el archivo evacuation.txt. Finalmente quedando compuesto dicho archivo por el total de personas simuladas, seguido de su tiempo de evacuación promedio, min y max. Luego almacenado lo mismo pero para residentes, visitantes tipo I, residentes separados por grupos etarios (G0,G1,G2,G3) y visitantes tipo I separados por grupos etarios. En la Figura 8 se muestra un ejemplo del archivo resultante.

```
alls : promedio: min:max  
residents :promedio: min:max  
visitorsI :promedio: min:max  
residents-G0 :promedio: min:max  
residents-G1 :promedio: min:max  
residents-G2 :promedio: min:max  
residents-G3 :promedio: min:max  
visitorsI-G0 :promedio: min:max  
visitorsI-G1 :promedio: min:max  
visitorsI-G2 :promedio: min:max  
visitorsI-G3 :promedio: min:max
```

Figura 8

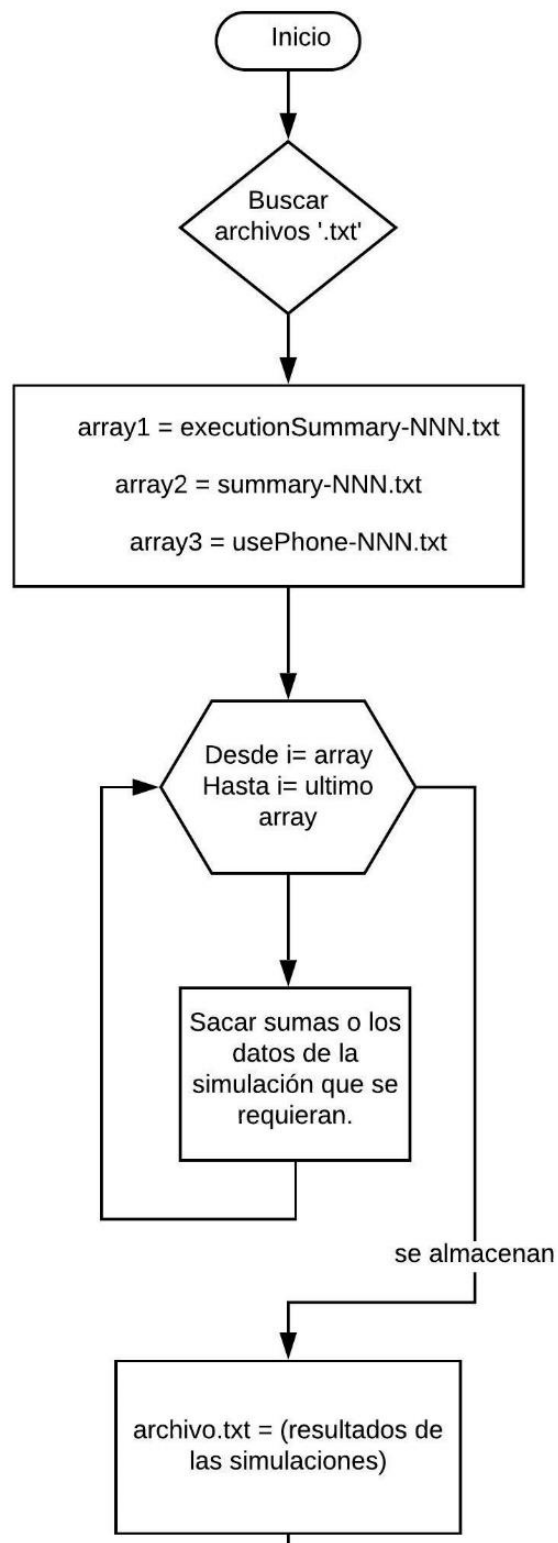
Para el último problema, al igual que los dos anteriores, se deben de encontrar el promedio, min y max pero en este caso del uso de teléfonos móviles para cada uno de los instantes de tiempo especificados de cada simulación. Es decir, que se deben encontrar la cantidad de personas que utilizaron el teléfono en un determinado tiempo de cada simulación, y almacenarlos en un archivo. Luego para calcular el promedio, el max y el min en un tiempo determinado, se debe de buscar en dicho archivo todos los resultados en un determinado tiempo, calculando el promedio de estos, el min y el max. Para finalmente almacenarlos en un archivo llamado usePhone-stats-txt. Así se debe hacer con todos los instantes de tiempo de las simulaciones, quedando el archivo compuesto por los 3600 instantes, junto a su promedios, max y min relacionado al uso de teléfonos de todas las simulaciones. Sin olvidar eliminar los archivos intermedios. En la Figura 9 se muestra un ejemplo de dicho archivo.

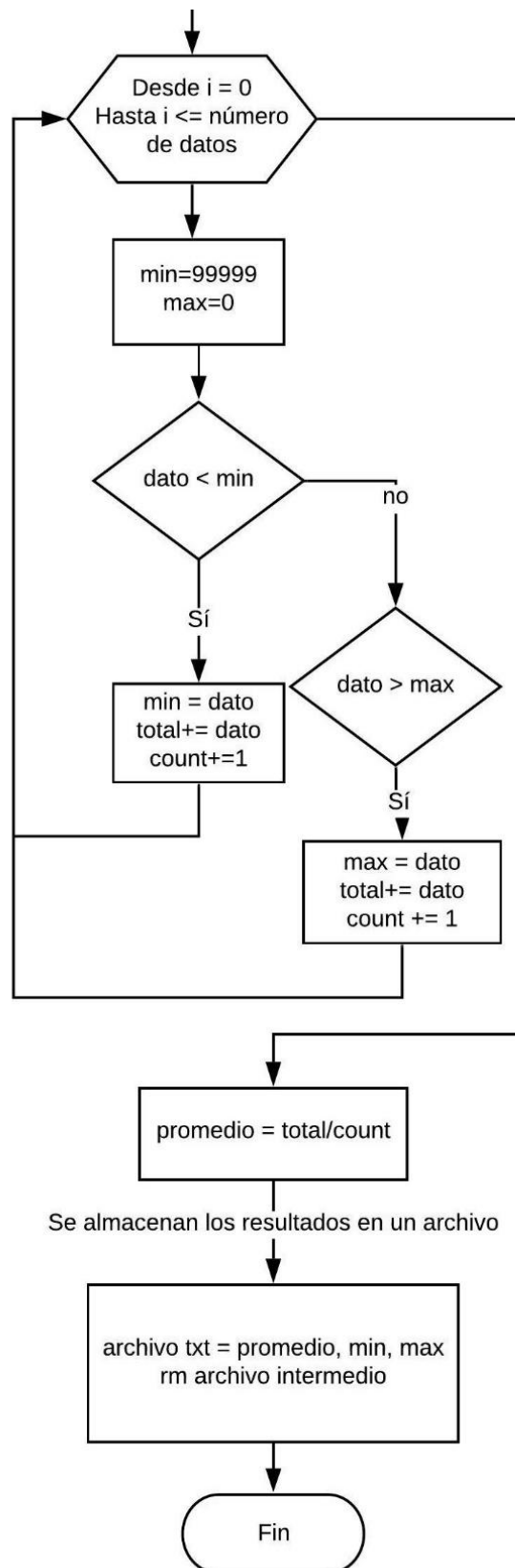
```
timestamp:promedio:min:max
```

Figura 9

3.2 Diagrama de alto nivel

Diagrama relacionado al diseño de la solución que se ocupará en los tres problemas.





4. Referencias

- [1] ¿Qué es un script?, <https://www.linuxadictos.com/que-es-script.html>
- [2] Bash, 21 de Mayo 2020, <https://es.wikipedia.org/wiki/Bash>
- [3] <https://www.linuxadictos.com/que-es-script.html>