

Reporte Técnico 1º Taller

Taller de Sistemas Operativos

Escuela de Ingeniería Informática

Felipe A. Rojas Gonzalez

felipe.rojasg@alumnos.uv.cl

Resumen---El objetivo principal del reporte técnico es explicar y desarrollar el diseño de un script en bash que procesa resultados de otros programas. En donde se desarrollará una breve introducción a cerca del trabajo a realizar para entrar en más detalles en la descripción del problema, donde se deben incluir los datos a utilizar y quedar explícito el significado de cada una de las variables utilizadas. Luego se diseñará la solución al problema donde tiene que incluir diagramas de alto nivel UML (4+1) que representara la arquitectura del software usando 5 vistas concurrentes de los diferentes stakeholders del sistema, pero solo mostrando la vista física de este en base al diseño y, además de los métodos utilizados para resolver el problema. Finalmente, para llegar al análisis de los resultados y una conclusión acorde a lo planteado en el reporte técnico.

1 Introducción

^[1] Bash es un lenguaje de comandos el cual está diseñado para comunicarse fácilmente entre diferentes programas, donde está diseñado para recibir entradas/comandos del usuario, como también puede leer y ejecutar comandos desde un archivo llamado ^[2] script de Shell el cual es una secuencia de comandos que se usan para designar a un programa relativamente simple, donde se van ejecutando comandos de forma secuencial de principio a fin. Los scripts regularmente no se compilan con anticipación a código máquina, sino que son ejecutados por un intérprete que lee el archivo de código fuente al momento.

^[3] El UML 4+1 consiste en un modelo de vistas diseñado por el profesor Philippe Kruchten y que se utiliza para describir la arquitectura de un sistema software intensivo basado en el uso de múltiples puntos de vista.

En este informe se realizará un script en bash que permitirá realizar estadísticas descriptivas de los datos entregados, los cuales se van a dividir en 3 grandes problemas para llevar a cabo una serie de operaciones como por ejemplo determinar la cantidad máxima, mínima y el promedio de una serie de datos los cuales se mencionaran detalladamente en el procedimiento del problema. Para el ingreso de parámetros, se exigirá el ^[4] estándar Unix. Los cuales se centran en la definición del API del sistema utilizando el lenguaje “C”. El objetivo es que un mismo código fuente pueda ser compilado en dos sistemas POSIX distintos sin necesidad de modificar el código. Los ^[5] parámetros son una entidad que almacena valores en donde puede ser un name, un numero o uno de los caracteres especiales. Una variable es un parámetro denotado por un name y se establece un parámetro si se le ha asignado un valor.

^[6] Awk es una excelente herramienta para crear scripts de Shell UNIX/Linux el cual es un lenguaje de programación que está diseñado para procesar datos basados en texto, ya sea en archivos o secuencias de datos o utilizando canalizaciones de Shell. Como también se usa ampliamente para listas indexadas por cadenas claves y expresiones regulares.

^[7] El comando find es muy útil ya que va a recorrer una jerarquía de archivos donde se puede utilizar para buscar archivos o directores y realizar operaciones posteriores en ellos. También admite búsquedas por archivos, carpetas, nombre, fecha de creación, etc.

La estructura del informe está compuesta por una sección que introduzca sobre los conceptos primordiales del objetivo del informe con su respectivo marco teórico, para luego presentar el trabajo realizado indicando como se abordó el problema explicando el procedimiento detalladamente y la descripción de los datos a utilizar como también el significado de las variables que llevaron a cabo. Para luego representar el diseño con diagramas de alto nivel como es el diagrama UML 4+1 enfocándose en la vista física del sistema (diagrama de despliegue), para pasar al análisis de los resultados obtenidos, cerrando con una conclusión acorde a lo presentado en el informe.

2 Descripción del Problema

En esta sección se detallará los 3 grandes problemas a desarrollar en este informe técnico. El trabajo a realizar consiste primero que nada en diseñar un script en Bash denominado stats.sh, que permita realizar estadísticas descriptivas de los datos entregados. Junto con ello para el ingreso de parámetros, se exigirá el estándar Unix y por otro lado el script debe realizar las siguientes tareas:

El primer problema es determinar la cantidad máxima, mínima y promedio para las siguientes métricas de desempeño computacional del simulador. Como también el tiempo de simulación total y la memoria utilizada por el simulador. Los resultados se deben entregar en un archivo metrics.txt con la siguiente estructura que se muestra en la (**Figura 1**):

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

Figura 1: Estructura para estadísticas de desempeño simulador

El segundo problema para realizar tiene que ver con determinar el tiempo promedio de evacuación, además del mínimo, y el máximo para los siguientes grupos de personas. (Todas las personas simuladas, Solo Residentes, Solo Visitantes Tipo1, Solo Residentes separados por grupo etario y Solo Visitantes tipo1 separados por grupo etario). Todos los resultados se deben entregar en un archivo evacuation.txt. con la siguiente estructura que se puede presentar en la (**Figura 2**):

```
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 2: Estructura para estadísticas personas.

El tercer problema consiste en determinar el promedio de uso de teléfonos móviles, además del mínimo y el máximo para los instantes de tiempo especificados en los archivos usePhone-NNN.txt. Los resultados se deben entregar en un archivo usePhone-stats.txt con la siguiente estructura que se muestra en la (Figura3):

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

Figura 3: Estructura para estadísticas de uso de teléfono móvil.

En el caso de que si se utilizan archivos intermedios para llevar acabo el problema, estos deben ser eliminados antes de la finalización del script

2.1 Descripción de los datos a utilizar y significado de las variables

Primero que nada, la organización de los datos consiste en que cada simulación entrega los resultados ordenados en una estructura de directorios donde NNN es el identificador de la simulación que se está analizando como se muestra en la (Figura 4).

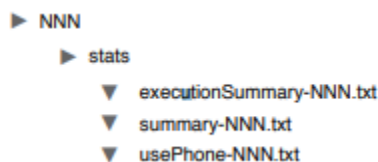


Figura 4: Estructura de directorios de los resultados de cada simulación.

El primer archivo para analizar es el executionSummary-NNN.txt que contiene datos sobre el desempeño de la simulación NNN. Posee un header (cabecera) y una segunda línea con los datos la cual tiene 10 campos, separados por el símbolo ‘:’ como se muestra en la (Figura5).

Campo	Descripción
numExperiment	Número del experimento
Tsim	Tiempo de simulación
CalibrationTime	Tiempo de calibración del simulador
Residents	Cantidad de Residentes simulados
Visitors	Cantidad de Visitantes simulados

timeExeMakeAgents	Tiempo real que demora en crear en memoria las personas simuladas
timeExecCal	Tiempo real que demora la calibración de las personas
timeExecSim	Tiempo real que toma la simulación
maxMemory	Costo espacial del simulador.
agentsMem	Memoria utilizada por las estructuras de datos relacionados con las personas.

Figura 5: Descripción de los campos del archivo executionSummary

Todos estos campos mencionados anteriormente contienen ciertas variables que significan algo específico cada una de ellas, pero en verdad solo 4 variables se necesitarán para llevar a cabo el primer problema donde se necesita hallar la cantidad máxima, mínima y el promedio del tiempo de simulación total y la memoria utilizada por el simulador. Para ello (“timeExecMakeAgents”+ “timeExecCal”+ “timeExecSim”), la suma de las 3 variables representan el tiempo de simulación total que se necesita, en cambio la variable “agentsMem” permitirá obtener la cantidad total de memoria utilizada por cada simulación que tienen como unidad KBytes.

El archivo summary-NNN.txt contiene datos sobre el comportamiento de las personas en la simulación NNN. Posee un header (cabecera), seguida de tantas líneas como la cantidad de personas que se simularon. Estas líneas tienen ocho campos, separados por el símbolo ‘:’ como se muestra en la (Figura 6).

Campo	Descripción
numExperiment	Número del experimento
id	Identificador de la persona simulada
model	Identificador del modelo de la persona 0: Residente 1: Visitante tipo I 2: Visitante tipo II
groupAge	Identificador del grupo etario de la persona 0: G0, 1: G1, 2: G2, 3: G3

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.
distanceToTargetPos	Distancia a la que quedo la persona de su objetivo inicial.
responseTime	Numero aleatorio que representa cuanto 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.
evacTime	Tiempo que la persona se demoró en llegar a la zona de evacuación. Si es 0, entonces la persona nunca llego a la zona de evacuación.

Figura 6: Descripción de los campos del archivo summary.txt

Los variables importante en este caso para dar pie al problema 2 serían el “responseTime” que ayudara a determinar el tiempo promedio, mínimo y máximo de todas las personas simuladas ya que representa el tiempo que se demoró la persona en tomar la decisión de evacuar. También se tiene “model” permite identificar qué tipo de persona es dependiendo de donde vive para llevar acabo las mismas operaciones, pero solo de los residentes y visitantes tipo1 y por último se tiene el “groupAge” que contiene un identificador para agrupar a las personas por su grupo etario e ir comparando con la variable “model” para hallar el tiempo promedio, mínimo y máximo de los diferentes grupos de personas junto con el grupo etario al que pertenecen.

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 un header (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 ‘:’ como se muestra en la (Figura 7).

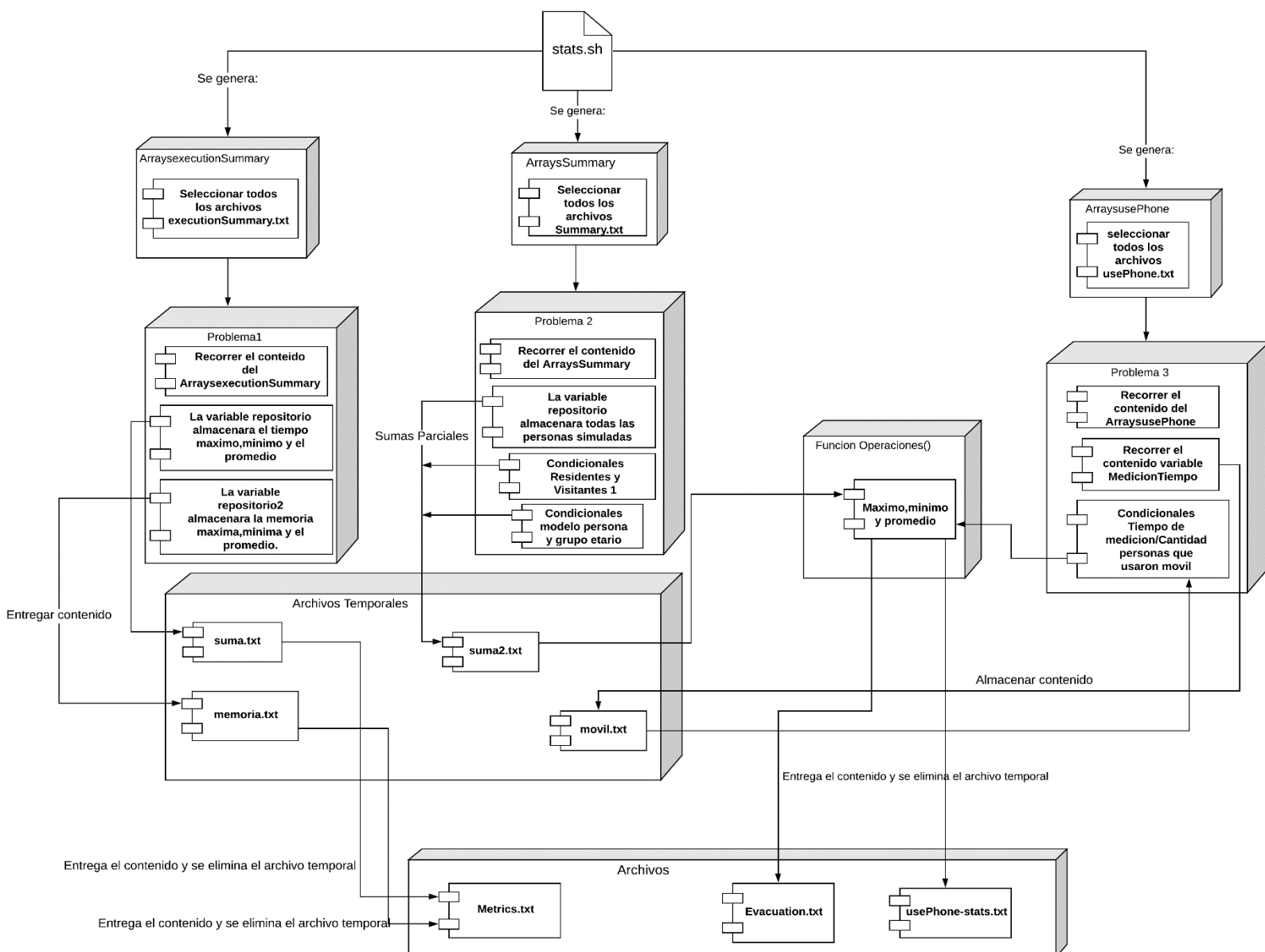
Campo	Descripción
numExperiment	Número del experimento
timeStamp	Tiempo de la medición
usePhone	Cantidad de personas que utilizaron el teléfono móvil en el tiempo especificado.

Figura 7: Descripción de los campos del archivo usePhone.txt.

En este caso las variables claves para dar pie al problema 3 son el “timeStamp” que permitirá llevar el tiempo de la medición que va a ir aumentando cada 10 segundos hasta llegar a los 3600 segundos (1 hora) por cada simulación, en cambio el “usePhone” representa la cantidad de personas que utilizaron el teléfono móvil en un tiempo especificado dentro de las simulaciones correspondientes. Junto con estas 2 variables se podrá desarrollar el promedio, mínimo y máximo de uso de los teléfonos móviles para cada instante de tiempo dentro las simulaciones.

3-Diseño

3.1 Diagrama de alto nivel



3.2 métodos para resolver el problema

En el problema 1 se abarco como un arreglo el cual con un comando find seleccionaría todos los archivos executionSummary.txt que existieran independientemente de donde uno este ubicado. En donde se recorre el contenido del arreglo uno por uno para ir analizando las 3 columnas que interesan de los archivos (timeExecMakeAgents”, “timeExecCal”, “timeExecSim”) los cuales se irán sumando por cada iteración y llevado a un archivo temporal en el cual se almacenaran las sumas parciales de cada simulación. Posteriormente la variable (repositorio) almacenará el contenido del archivo temporal suma.txt dado un awk que cumplirá la función de sacar el tiempo máximo, mínimo y promedio de cada simulación. Realizando el mismo procedimiento para hallar la memoria utilizada por el simulador, donde todo esto se guardará en dos variables repositorio y repositorio2 para llevar su contenido a un archivo metrics.txt con la estructura mencionada anteriormente. Para cumplir con las observaciones detalladas en el informe se eliminarán los dos txt temporales que se utilizaron en el proceso.

En el problema 2 se creará un segundo arreglo el cual cumplirá la función del almacenar todos los archivos Summary.txt que existan gracias al comando find. Luego se va a recorrer el arreglo por cada archivo Summary que corresponde a cada simulación determinada y se analizaran los campos importantes para resolver el problema como son el “responseTime” para determinar el tiempo de todas las personas simuladas gracias al awk. Junto con esto se invoca a una función llamada Operaciones que su objetivo es sacar el promedio, máximo y mínimo de todas las personas simuladas y llevar esos datos a un archivo temporal llamado evacuation.txt. Como siempre eliminando los archivos intermedios como es el suma2.txt, el cual se ocupará para las siguientes métricas de desempeño a analizar en los distintos grupos de personas cómo son solo los residentes, solo visitantes tipo1, solo residentes separados por grupo etario y solo visitantes tipo 1 separados por grupo etario. En todos estos grupos de personas se realizará el mismo procedimiento mencionado para todas las personas simuladas solo con una leve diferencia que al momento de recorrer el arreglo se tendrá que someter el contenido del archivo a un condicional con respecto al campo (model), el cual si es igual a cero serian de los solo residentes y se realizaría todo el proceso normal, pero si el campo es igual a 1 seria de solo los visitantes 1. De esta manera para diferenciar el tipo de modelo de persona y el grupo etario al que pertenece se realiza un condicional con respecto al contenido del archivo en el campo (model) que sea igual a cero (solo residentes) y el campo (groupAge) con todas las posibilidades, lo cual abarcara a todos los residentes con respecto al grupo etario. Lo mismo se debe hacer para los visitantes tipo1 en relación con el grupo etario, la única diferencia es que el campo (model) será igual a 1 lo que indica que es visitante tipo1, de este mismo modo realizando todo el procedimiento y las operaciones ya mencionadas anteriormente.

El problema 3 igualmente se piensa como un arreglo en donde el comando find selecciona todos los archivos llamados usePhone.txt que se encuentran en el servidor dejándolos ordenados por nombre en un arreglo. Cumpliendo el mismo procedimiento que en los otros 2 problemas donde se recorre el arreglo mostrando el contenido de cada archivo mediante ocurre la iteración cortando los campos importantes para resolver el problema como son (timeStamp, usePhone) y guardándolos en otro arreglo llamado MedicionTiempo. Donde se realizará otro ciclo para recorrer el contenido del arreglo MedicionTiempo y almacenarlo en un archivo temporal llamado movil.txt donde las columnas queden puestas como filas es decir realizar la transpuesta de esa matriz.

Al tener las columnas puestas como filas en una matriz se podrá recorrer más fácilmente ya que se tiene que analizar el tiempo promedio de uso de teléfonos para los instantes de tiempos especificados. Donde se ira comparando cada tiempo de medición (timeStamp) con respecto a la cantidad de personas que usaron el móvil (usePhone) y se podrá llamar a la función Operaciones que realiza el promedio, máximo y mínimo de ese tiempo en específico. Para finalmente guardar todo el contenido en usePhone-stats.txt y eliminar los archivos temporales.

IV- Resultados esperados

Primero que nada, tener una claridad de la descripción del problema a realizar que con lleva a incluir una descripción de los datos a utilizar acorde a lo planteado, como también el significado de cada una de las variables utilizadas. Otra cosa importante es tener bien en cuenta el procedimiento del problema que se tiene que llevar a cabo por medio de diversas normas establecidas en el informe, representándolo en un diseño de la solución bien elaborado que represente la implementación a realizar en sus rasgos generales donde debe incluir un diagrama de alto nivel como es el diagrama de despliegue de la vista física del sistema y además de tener conciencia de los métodos utilizados para resolver el problema.

V- Conclusiones

Recomendaciones importantes es tener en cuenta que siempre se debe tener claro el problema que se intenta resolver antes de programar. Ya que se tiene que establecer un diseño antes de una implementación del script, porque con ella se pueden aclarar muchas cosas antes de comenzar a programar el código. Donde el diseño y la implementación tienen que ser coherentes entre sí, sin tener ninguna anomalía que los distinga. Muchas veces un código limpio, bien comentado es el mejor diseño.

VI- Referencias

[1]"Bash", *Es.wikipedia.org*, 2020. [Online]. Available: <https://es.wikipedia.org/wiki/Bash>. [Accessed: 07-Jun- 2020].

[2]"Como hacer un shell script en Linux", *Profesional Review*, 2020. [Online]. Available: <https://www.profesionalreview.com/2017/03/12/shell-script-linux/>. [Accessed: 07- Jun- 2020].

[3]"Modelo "4+1" vistas de Kruchten (para Dummies)", *Jarroba*, 2020. [Online]. Available: <https://jarroba.com/modelo-41-vistas-de-kruchten-para-dummies/>. [Accessed: 29- Apr- 2020].

[4]"Estandares UNIX (II)", *Gii.upv.es*, 2020. [Online]. Available: <http://www.gii.upv.es/rportal/tutorial/01-intro/tr9.html>. [Accessed: 07- Jun- 2020].

[5]"Shell Parameters (Bash Reference Manual)", *Gnu.org*, 2020. [Online]. Available: https://www.gnu.org/software/bash/manual/html_node/Shell-Parameters.html. [Accessed: 08- Jun- 2020].

[6]"How To Use awk In Bash Scripting - nixCraft", *nixCraft*, 2020. [Online]. Available:
<https://www.cyberciti.biz/faq/bash-scripting-using-awk/#:~:text=AWK%20is%20a%20programming%20language,in%20your%20bash%20shell%20scripts.>
[Accessed: 09- Jun- 2020].

[7]"find command in Linux with examples - GeeksforGeeks", *GeeksforGeeks*, 2020. [Online]. Available:
<https://www.geeksforgeeks.org/find-command-in-linux-with-examples/>. [Accessed: 09- Jun- 2020].