

Taller de Sistemas Operativos

Taller 02

Profesor: Gabriel Astudillo Muñoz
Escuela de Ingeniería Civil Informática
Universidad de Valparaíso

1 Objetivo

Implementar scripts que permitan ejecutar repetidamente un programa externo, calcular tiempos de ejecución y graficarlos. Taller grupal.

2 Instrucciones

Su trabajo deberá ser entregado en el servidor, en `~/taller02`. Este directorio deberá tener la siguiente estructura:

```
+ ~/taller01
  - README
  - bvr.sh
  - graph.plt
  - heapsort.pl
```

El archivo **README** es un archivo en texto plano, donde se debe explicar el diseño de su solución, funciones utilizadas, etc. Además, debe especificar los autores y sus correos.

El código del script debe ser ordenado y utilizar estructuras de datos apropiadas.

3 Antecedentes

3.1 Programa heapsort.pl

Este programa ordena mediante heapsort cierta cantidad de datos, especificados mediante el parámetro `--size`. Al terminar, el programa informa sobre la cantidad de datos ordenados, el tiempo de ejecución en segundos y la cantidad de operaciones realizadas.

3.2 Software para generar gráficos: GnuPlot

GnuPlot¹ es un programa para realizar gráficos a partir de datos numéricos. En Internet puede buscar información y ejemplos de scripts. Para este taller, sus scripts deben basarse en los ejemplos estudiados en la **actividad 2**, recientemente realizada en clases.

¹ <http://gnuplot.sourceforge.net>

4 Trabajo a realizar

Su grupo deberá caracterizar el comportamiento temporal del programa **heapsort.pl**, a través de un script BASH denominado **bvr.sh**. Para esto, deberá medir el tiempo de ejecución para las siguientes cantidades de datos: $\{1 \cdot 10^6, 2 \cdot 10^6, \dots, 10 \cdot 10^6\}$. Para cada una de estas cantidades, se deben realizar como mínimo 10 mediciones y luego calcular el promedio, su máximo y mínimo. Todas las mediciones se deben almacenar en un archivo llamado **graph.txt**. Luego, debe graficar dichos datos a través del script **graph.plt**, el que generará un archivo PNG con nombre **graph.png**. Una propuesta de pseudo-código se muestra en continuación:

Script **bvr.sh**

Entradas:

Ninguna

Salida:

Gráfico PNG para visualizar el costo en tiempo del programa de ordenamiento **heapsort.pl**

Inicio

Para cada N en $\{1 \cdot 10^6, 2 \cdot 10^6, \dots, 10 \cdot 10^6\}$

Ejecutar 10 veces **heapsort --size N**.

Calcular tiempo de ejecución promedio, máximo y mínimo.

Agregar datos a **graph.txt**.

Fin Para Cada N

Ejecutar **graph.plt** (script que genera **graph.png** a partir del archivo **graph.txt**)

Fin

Como resultado esperado, el gráfico contenido en **graph.png** debe ser similar al mostrado en la Figura 1.

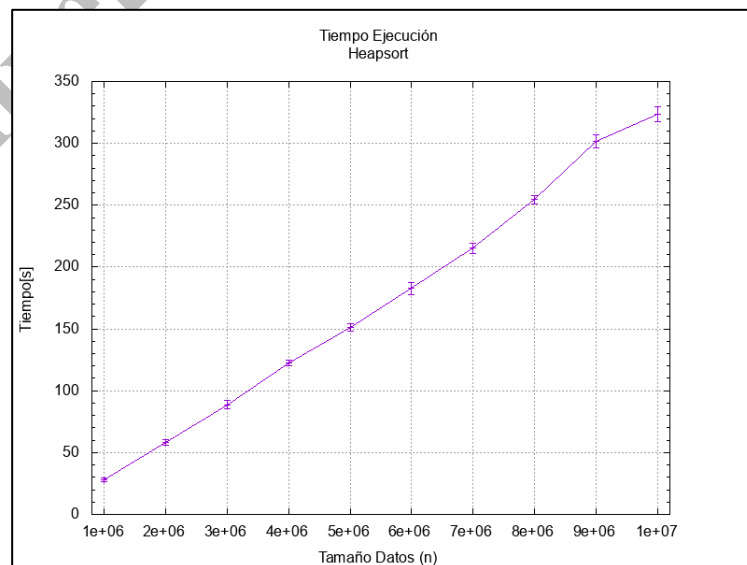


Figura 1