

# Organización y Arquitectura de Computadoras

## Practica 1: Medidas de Desempeño

Profesor: José Galaviz Casas

Ayudante de Laboratorio: Ricardo Enrique Pérez Villanueva

21 de agosto de 2025

### 1. Objetivos

#### 1.1. Generales

El alumno aprenderá como medir el desempeño de sistemas de computo.

#### 1.2. Particulares

Al finalizar la practica el alumno tendrá la capacidad de:

- Ejecutar pruebas de desempeño con la batería de pruebas phoronix [Phoronix Test Suite].
- Realizar estudios comparativos del desempeño de sistemas de computo.
- Determinar la medida de tendencia central adecuada a la muestra tomada para sintetizar el desempeño.

### 2. Requerimientos

#### 2.1. Conocimientos Previos:

- De preferencia, el sistema operativo GNU/Linux y el manejo de comandos en una terminal.
- Conceptos básicos de medidas de desempeño: tiempo de respuesta, rendimiento y pruebas de desempeño.
- Las propiedades básicas de las siguientes medidas de tendencia central:
  - Media aritmética.
  - Media aritmética ponderada.
  - Media armónica.
  - Media geométrica.

#### 2.2. Tiempo de realización sugerido: 5 horas

#### 2.3. Numero de colaboradoras: Equipos de 4 personas

#### 2.4. Software a utilizar.

- El paquete phoronix-test-suite, el cual puede ser encontrado en los repositorios de las principales distribuciones de Linux o es posible descargarlo de forma gratuita directamente del sitio web [Phoronix].
- Mucho tiempo de espera.
- Opcionalmente, la plataforma de juegos Steam.

### 3. Planteamiento

El diseñador de arquitecturas de computo debe tomar varias decisiones que tienen impacto directo tanto en el costo como en el desempeño del sistema, por lo que requiere de mecanismos eficaces para medir y comparar el rendimiento de los sistemas de computo. La forma de llevar a cabo el análisis es a través de la ejecución de baterías de pruebas. Nuestra tarea sera ejecutar algunas pruebas representativas para conocer el rendimiento de nuestra computadora personal y compararlo con el de otros equipos de computo.

### 4. Desarrollo:

#### 4.1. Phoronix-test-suite

Phoronix-test-suite es una batería de pruebas de código abierto desarrollada por Phoronix.com bajo la licencia GNU-GPL. Algunas de sus características que la hacen útil para nuestro propósito son:

- Se encuentra disponible para distintos sistemas operativos.
- Arquitectura extendible.
- Mas de 450 perfiles de pruebas.
- Mas de 150 baterías de pruebas.
- La descarga e instalación de pruebas es automática.
- Detección detallada de software y hardware.

#### 4.2. Instalación

La plataforma se encuentra disponible en los repositorios de las principales distribuciones de Linux, instala el paquete a través del manejador de paquetes de tu distribución.

En el caso de ubuntu, basta con hacer:

```
$ sudo apt install phoronix-test-suite
```

En otro caso, es posible descargar el paquete de la pagina oficial e instalarlo de forma manual. En caso de requerir alguna dependencia el programa informara cual paquete es necesario al ejecutarse alguna prueba.

#### 4.3. Uso

El uso de la batería de pruebas es a través de la linea de comandos, los comandos básicos para ejecutar las pruebas son:

- Mostrar todas las pruebas disponibles:  

```
$ phoronix-test-suite list-tests
```
- Para conocer más información acerca de alguna prueba en particular:  

```
$ phoronix-test-suite info <nombre de la prueba>
```
- Para ejecutar una prueba:  

```
$ phoronix-test-suite benchmark <nombre de la prueba>
```

Las pruebas no se incluyen en la instalación del paquete, por lo que la primera vez que se ejecute alguna prueba, los archivos necesarios serán descargados y las dependencias faltantes serán instaladas a través del manejador de paquetes de la distribución.
- Para desinstalar una prueba:  

```
$ phoronix-test-suite remove-installed-test <nombre de la prueba>
```

## 4.4. Pruebas

A continuación, se encuentra una lista y una breve descripción de las pruebas que serán utilizadas en esta practica. En el sitio web de Open Benchmarking [OpenBenchmarking] se pueden encontrar las pruebas disponibles, la descripción de las pruebas y resultados enviados por otros usuarios.

- **GZIP Compression**

Esta prueba mide el tiempo necesario para comprimir un archivo usando GZIP. Es una prueba de Procesador.

`pts/compress-gzip`

- **Fhourstones**

Resuelve posiciones del juego Conecta 4, jugado en un tablero de 7x6. Usa un algoritmo alfa-beta para calcular las mejores posiciones. Es una prueba de Procesador.

`fhourstones`

- **CUDA Mini-Nbody**

La version CUDA de la prueba "Mini-nbody". Es una prueba de Gráficos.

`cuda-mini-body`

- **Team Fortress 2**

Monitoriza el rendimiento del juego Free-To-Play, Team Fortress 2. Esta prueba asume que ya tienes instalado tanto Steam como Team Fortress 2 en tu computadora. Esta prueba mide gráficos, y se puede sustituir por otra prueba que compile un videojuego, como Cyberpunk 2077 o Counter Strike Source. También se puede sustituir por otra prueba de gráficos si no se quiere descargar Steam y/o juegos.

`tf2`

- **Git**

Toma el tiempo en que toma llevar a cabo ciertas operaciones de Git a un repositorio estatico de ejemplo. Es una prueba de Sistema.

`pts/git`

- **REDIS**

Redis es un motor de bases de datos en memoria de código abierto, la prueba mide el numero de solicitudes que el servidor puede responder por segundo. Es una prueba de Sistema

`pts/redis` o `windows/redis`

- **BlogBench**

Prueba que replica la carga real en un servidor de archivos, estresando el sistema de archivos con hilos que escriben, leen y reescriben. Es una prueba de Disco.

`system/blogbench`

- **Unpacking The Linux Kernel**

Esta prueba mide el tiempo que toma extraer el archivo .tar.xz que contiene el kernel de Linux. Es una prueba de Disco.

`unpack-linux`

## 5. Entrada

No hay datos de entrada para esta practica.

## 6. Salida

El equipo deberá entregar un reporte con la información requerida en las secciones de ejercicios y preguntas.

## 7. Procedimiento

### 7.1. Tabla de especificaciones.

En equipo, elaboren una forma para reportar los resultados de las pruebas. La forma deberá solicitar la siguiente información:

- Nombre del Alumno.
- Datos de la Computadora.
  - Fabricante y modelo de la computadora.
  - Tipo de Mother Board
  - Fabricante, modelo, capacidad de la GPU, en caso de tenerla. Se puede sustituir por los gráficos integrados.
  - Fabricante, modelo, frecuencia, número de núcleos y arquitectura del procesador.
  - Capacidad y tipo de memoria RAM y de caches de los procesadores.
  - Capacidad, tipo y velocidad del disco duro.
  - Distribución del sistema operativo y versión del Kernel.
- Una tabla con dos columnas para reportar los resultados, la primera columna será el nombre de la prueba y la segunda el resultado indicando la unidad de medida.

### 7.2. Análisis de datos

Cada miembro del equipo deberá ejecutar las pruebas en una computadora y reportar los resultados en una forma. Las computadoras deberán diferir en al menos una característica (sistema operativo, procesador, capacidad de memoria RAM, etc.). Si alguna de las pruebas es imposible de ejecutar, el equipo deberá escoger otra prueba que corresponda al mismo tipo (tiempo de respuesta o rendimiento, además de que está probando (Disco, procesador, gráficos o sistema) y sustituir los resultados.

## 8. Ejercicios

- Identifica cuáles de las pruebas miden el tiempo de respuesta y cuáles miden el rendimiento.
- Usando la medida de tendencia central adecuada y tu reporte de resultados, calcula:
  - La medida de tiempo de respuesta.
  - La medida de rendimiento.

Agrega los resultados obtenidos a tu reporte y compártelo con tus compañeros de equipo.

- Una vez que tengas los reportes de tus compañeros; Cada alumno fijará su computadora como computadora de referencia, después calcula los tiempos normalizados y obtén la medida de tendencia central adecuada de cada una de las computadoras. Agrega cada tabla obtenida al reporte. Al final, el reporte debe tener 4 tablas donde se usa cada equipo como computadora de referencia.
- Dada una prueba de rendimiento y otra de tiempo de respuesta, cada alumno deberá realizar el siguiente análisis: ¿Si pudieras cambiar una pieza de tu computadora para que la prueba se pudiera mejorar, que cambiarías? ¿Cuánto costaría el cambio? ¿Qué tanta sería la mejora que este cambio da?

## 9. Preguntas

1. ¿Cual computadora tiene el mejor tiempo de ejecución? Comparada con la computadora con la peor medida de tiempo de ejecución ¿por que factor es mejor la computadora? Enuncia el resultado de la forma “El tiempo de ejecución de la computadora A es x veces \_\_\_\_\_que la computadora B”.
2. ¿Cual computadora tiene el mejor rendimiento? Comparada con la computadora con el peor rendimiento ¿por que factor es mejor la computadora? Enuncia el resultado de la forma “El rendimiento de la computadora A es x veces \_\_\_\_\_que la computadora B”.
3. Considera todas las computadoras usadas como referencia; Para cada computadora, ¿cual computadora tiene el mejor desempeño y cual computadora tiene el peor desempeño?
4. De entre los atributos de **cada** maquina comparada, ¿Que maquina tiene el mejor disco duro? ¿Cual tiene la mejor GPU?, ¿Cual tiene la mejor RAM? ¿Que los hace mejores que los otros? ¿Cuales resultan determinantes en la perdida o ganancia de desempeño en las pruebas realizadas?
5. Tomando en cuenta la computadora con el mejor rendimiento y la del peor rendimiento: ¿La computadora con el peor rendimiento sufre de **Bottleneck**? ¿Que tal la del mejor rendimiento, tiene Bottleneck? Si alguna de las respuestas es correcta, ¿Donde se presenta?
6. Supongamos que puedes cambiar solo un componente para cada una de las 4 computadoras, pero el resto de tu equipo sigue siendo el mismo. ¿Que cambiarías para tener una mejora en rendimiento del 50 % en **Compilación de código y procesamiento de videos**? Justifica tus respuestas. (Si tu computadora no tiene GPU, o mas de 1 stick de RAM, es valido añadir uno)