



**USAC**  
**TRICENTENARIA**  
Universidad de San Carlos de Guatemala

# CLASE 9

# **PROFILING**


Análisis y Diseño de Sistemas 2

Aux. Kenny Miguel Eguizabal

# PROFILING

- En ingeniería de software Profiling o llamado en español *Análisis de Rendimiento o Perfilaje*, es la investigación del comportamiento de un programa de computadora usando información reunida desde el análisis dinámico del mismo.
- El Profiling se puede llevar a cabo en el código fuente o sobre un binario ejecutable mediante una herramienta llamada *profiler*.

# Objetivo

- Averiguar el tiempo dedicado a la ejecución de diferentes partes del programa para detectar los puntos problemáticos y las áreas donde sea posible llevar a cabo una optimización del rendimiento (ya sea en velocidad o consumo de recursos)
- 

# Clasificación de los Profilers

- Los profilers pueden clasificarse según la forma de recopilación de datos que utilicen, pudiendo destacar:
  - basados en eventos
  - Estadísticos
  - con instrumentación de código
  - como simulación.

- Los profilers, que también son propios programas específicos mediante la recopilación de información sobre su ejecución. Basado en su granularidad de datos, la forma en que los profilers recopilan información, se clasifican en profilers *basados en eventos* o *estadísticos*. Ya que los profilers interrumpen la ejecución del programa para recopilar información, tiene una resolución finita en las mediciones de tiempo, los cuales se deben de tomar como un subconjunto del total de la información.

# Profilers basados en eventos

Los lenguajes de programación que se listan a continuación poseen un profiler basado en eventos:



PROGRAMMING  
Language



python



TM

Microsoft®  
.NET




# Profilers Estadísticos

- Algunos profilers se calculan por muestreo. Un profiler por muestreo prueba el “Programa Counter” del programa objetivo a intervalos regulares usando interrupciones del sistema operativo.
- Los profilers de muestreo son típicamente menos exactos numéricamente y específicos, pero permite que el programa de destino funcione cerca de la velocidad máxima.

# Profilers Instrumentadores

- Algunos programas “Instrumentan” el programa objetivo con instrucciones adicionales para recopilar la información necesaria.
- Instrumentar el programa puede causar cambios en el rendimiento del programa, que puede causar resultados inexactos y *heisenbugs*. Instrumentar siempre tendrá algún impacto en la ejecución del programa, por lo general siempre es más lento. Sin embargo, la instrumentación puede ser muy específica y ser controlada cuidadosamente para tener un impacto mínimo.



- El impacto de la instrumentación a menudo se puede deducir (es decir, eliminada por sustracción) a partir de los resultados.
  - **gprof** es un ejemplo de un profiler que utiliza tanto la instrumentación y el muestreo. La instrumentación se utiliza para recopilar información de las llamadas y los valores de tiempo real se obtiene mediante muestreo estadístico.
- 

# PRACTICA #2

- Realizar un manual sobre GPROF ,
- -Una breve explicación de ¿Qué es? y ¿Para qué sirve?
- -instalar si es que se instala
- -Cómo se utiliza?
- tomar screenshot (sobre todo que aparezca su nombre o Carne, en la consola) de como utilizarlo, y explicando los resultados de profiling
- hacer un ejemplo de código para el análisis de rendimiento utilizando la herramienta gprof para el análisis y colocar los resultados.
- -Conclusiones
- -Recomendaciones
- **Copias parciales o totales tendrán una nota de Cero.**
- Enviar el día Viernes 22 de abril antes de Media Noche
- al correo 200818973@ingenieria.usac.edu.gt
- Asunto: [AyD2]Practica2\_#carné
- Archivo: [AyD2]Practica2\_#carné.pdf

# Referencias para la Práctica

- <http://docs.freebsd.org/44doc/psd/18.gprof/paper.pdf>
- <http://www.chuidiang.com/clinix/herramientas/profiler.php>
- <https://sourceware.org/binutils/docs/gprof/Compiling.html#Compiling>
- <https://sourceware.org/binutils/docs/gprof/>
- <http://www.thegeekstuff.com/2012/08/gprof-tutorial/>

**¡GRACIAS!**