

## Práctica 3. Perfilado con código fuente

En esta práctica veremos cómo realizar el perfilado cuando disponemos del código fuente. Utilizaremos una **máquina virtual**.

### Contenido:

[gcc \(~40 min.\)](#)  
[gprof \(~20 min.\)](#)  
[google-pprof \(~30 min.\)](#)

### gcc (~40 min.)

Compila el programa `edges.c` (disponible con la práctica) con cada nivel de optimización (opciones `-O0`, `-O1`, `-O2` y `-O3`). Por ejemplo:

```
$ gcc -o edges0 -O0 edges.c
```

Se puede usar la opción `-fopt-info` para obtener un informe de optimización.

Obtén los tiempos de ejecución para cada nivel de optimización al procesar la imagen `img.pgm` (también disponible con la práctica). Por ejemplo:

```
$ time ./edges0 img.pgm out.pgm
```

Obtén los tiempos de ejecución compilando con optimización basada en perfil. Para ello:

1. Compila el programa con `-O1` añadiendo instrumentación para generar un perfil (opción `-fprofile-generate`).
2. Ejecuta el programa (se generará el fichero `edges.gcda`):  

```
$ ./edges1pg img.pgm out.pgm
```
3. Vuelve a compilar el programa para cada nivel de optimización (`-O1`, `-O2` y `-O3`) activando la optimización basada en perfil (opción `-fprofile-use`).
4. Mide el tiempo de ejecución con cada nivel de optimización (`-O1`, `-O2` y `-O3`). Por ejemplo:

```
$ time ./edges1pu img.pgm out.pgm
```

*Crea una tabla con los resultados y escribe un breve análisis de los mismos.*

### gprof (~20 min.)

Consulta la página de manual de `gprof`.

Obtén un perfil de ejecución del programa `edges.c` (sin optimización).

*Copia el perfil y escribe un breve análisis de los resultados.*

Obtén el tiempo de ejecución del programa sin instrumentación y con ella. Calcula la sobrecarga producida por la instrumentación como la diferencia entre los tiempos de ejecución con y sin instrumentación, dividida por el tiempo de ejecución con instrumentación y multiplicada por 100, en porcentaje.

*Escribe los tiempos de ejecución y la sobrecarga.*

Obtén un perfil de ejecución por líneas de código usando la opción `-l`. Es necesario haber compilado el programa con la opción `-g`, que añade información de depuración al ejecutable.

## google-pprof (~30 min.)

Instala Google Performance Tools con:

```
$ sudo apt-get update
$ sudo apt-get install google-perftools libgoogle-perftools-dev
```

Consulta la página de manual de `google-pprof`. Más información en <https://gperftools.github.io/gperftools/cpuprofile.html>.

Obtén un perfil de ejecución en formato de texto del programa `edges.c` (en versiones recientes de `gcc` es necesario usar la opción `-Wl,--no-as-needed` para forzar el enlace).

*Copia el perfil y compáralo con el obtenido con `gprof`.*

Genera un grafo de llamadas en formato SVG y visualízalo en un navegador *web* con:

```
$ google-pprof --web edges /tmp/edges.prof
```

También se puede generar una imagen en formato PDF (`--pdf`), GIF (`--gif`), SVG (`--svg`)...

*Copia una imagen o captura de pantalla con el grafo de llamadas.*

Con la variable `CPUPROFILE_FREQUENCY` se establece la frecuencia de muestreo (por defecto, 100). Definiendo la variable `CPUPROFILE_REALTIME` se muestrea en tiempo real, por lo que es más apropiado para programas que realicen E/S y, además, permite mayor frecuencia de muestreo. Por ejemplo:

```
$ CPUPROFILE=/tmp/edges.prof CPUPROFILE_FREQUENCY=1000
CPUPROFILE_REALTIME=1 ./edges img.pgm out.pgm
```

Obtén el tiempo de ejecución del programa sin instrumentación, con la instrumentación por defecto y con instrumentación en tiempo real con frecuencia 1000. Calcula la sobrecarga producida por la instrumentación en ambos casos.

*Escribe los tiempos de ejecución y la sobrecarga.*

Obtén un perfil de ejecución por líneas de código añadiendo la opción `--lines`. Es necesario haber compilado el programa con la opción `-g`.

*Compara `gprof` y `google-pprof` en cuanto a implementación, sobrecarga, precisión, exactitud, ámbito de medida, resolución temporal y facilidad de uso.*