

Práctica 2.3. Perfilado con código fuente

Creación de una máquina virtual para pruebas

Arranca Linux y entra como “Usuario VMs”. Introduce tu usuario y contraseña.

Abre la carpeta `Disco VMs` desde el escritorio y ve al directorio `ECO`. Abre el fichero `ECO.ova` haciendo doble *click*. Pulsa en “Importar” en la ventana de VirtualBox que aparecerá.

Desde VirtualBox, selecciona la máquina virtual “ECO” y pulsa en “Iniciar” para arrancarla. Entra como usuario “usuario”, con contraseña “usuario”.

gcc (~30 min)

Compila el programa `edges.c` (disponible en el Campus Virtual) con diferentes niveles de optimización (`-O0`, `-O1`, `-O2` y `-O3`) y obtén los tiempos de ejecución con cada nivel al procesar la imagen `img.pgm` (también disponible en el Campus Virtual):

```
$ ./edges img.pgm out.pgm
```

Repite el ejercicio anterior activando las optimizaciones guiadas por perfil de ejecución. Para ello:

1. Compila el programa (sin optimización) activando la generación del perfil de ejecución (opción `-fprofile-generate`)
2. Ejecuta el programa con la imagen (se genera el fichero `edges.gcd`)
3. Vuelve a compilar el programa con cada nivel de optimización usando el perfil (opción `-fprofile-use`)

Entrega: Haz una tabla con los resultados y escribe un breve análisis de los mismos.

gprof (~30 min)

Consulta la página de manual de `gprof`.

Obtén el tiempo consumido por cada función de `edges.c` compilando sin optimización.

¿Qué función intentarías mejorar primero? ¿Cuánto tardaría en ejecutarse el programa si consiguieras mejorar esa función en un 15%? ¿Cuál sería la máxima mejora que podrías obtener mejorando solamente esa función?

Entrega: Copia los resultados y responde a las preguntas.

También se puede hacer un análisis por líneas de código usando la opción `-l` de `gprof`. Lógicamente, es necesario haber compilado con la opción `-g`.

google-pprof (~30 min)

Instala el paquete `google-perftools` con:

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

Consulta la página de manual de `google-pprof`.

Compila el programa `edges.c` enlazando con la biblioteca `profiler`:

```
$ gcc -o edges edges.c -lprofiler
```

Activa el muestreo durante la ejecución del programa definiendo la variable de entorno `CPUPROFILE` con el fichero de perfil:

```
$ CPUPROFILE=/tmp/edges.prof ./edges img.pgm out.pgm
```

Obtén un informe a partir del perfil de ejecución con el comando `google-pprof`:

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

El significado de cada columna es el siguiente:

1. Número de muestras de esta función
2. Porcentaje de muestras de esta función
3. Porcentaje acumulado de muestras de esta función y las anteriores
4. Número de muestras de esta función y de las funciones a las que llama
5. Porcentaje de muestras de esta función y de las funciones a las que llama
6. Nombre de la función

También se puede obtener un informe gráfico en formato PDF (opción `--pdf`), GIF (opción `--gif`) o SVG (opción `--svg`). Por ejemplo:

```
$ google-pprof --gif edges /tmp/edges.prof > edges.gif  
$ xdg-open edges.gif
```

Entrega: Copia el fichero `edges.gif`.

Se puede modificar la frecuencia de muestreo (por defecto, 100) definiendo la variable de entorno `CPUPROFILE_FREQUENCY`:

```
$ CPUPROFILE=/tmp/edges.prof CPUPROFILE_FREQUENCY=200 ./edges  
img.pgm out.pgm  
$ CPUPROFILE=/tmp/edges.prof CPUPROFILE_FREQUENCY=300 ./edges  
img.pgm out.pgm
```

Por defecto, se usa el temporizador de tiempo de CPU `ITIMER_PROF` (ver `man getitimer`) para muestrear. El temporizador de tiempo real `ITIMER_REAL` es más apropiado para programas que realicen E/S y permite mayor frecuencia de muestreo.

Para usarlo, se define la variable `CPUPROFILE_REALTIME`:

```
$ CPUPROFILE=/tmp/edges.prof CPUPROFILE_FREQUENCY=300  
CPUPROFILE_REALTIME=1 ./edges img.pgm out.pgm  
$ CPUPROFILE=/tmp/edges.prof CPUPROFILE_FREQUENCY=1000  
CPUPROFILE_REALTIME=1 ./edges img.pgm out.pgm
```

Obtén el tiempo de ejecución sin muestreo, muestreando con la frecuencia y el temporizador por defecto y muestreando 1000 veces por segundo con el temporizador de tiempo real. ¿Qué sobrecarga se produce en cada caso?

Entrega: Copia los resultados de este último ejercicio y responde a las preguntas.

Se puede obtener un informe por líneas de código en lugar de por funciones, usando la opción `--lines` (es necesario haber compilado con la opción `-g`):

```
$ google-pprof --text --lines edges /tmp/edges.prof
```

Para calcular la sobrecarga

1: Saber el período muestreo

pe \rightarrow 1s \rightarrow 1000 muestras

$$\text{Intervalo muestreo} = 0,001$$

2: Saber cada activación cuanto tiempo CPU consume

Para ello, cogemos el tiempo de ejecución

$$\begin{array}{lcl} T_{\text{sin muestreo}} = 4,375 & \xrightarrow{f=1000} & \text{si en 4370 muestras se consume } 0,47 \\ T_{\text{con muestreo}} = 4,84 & & \end{array}$$

$$\text{dif} = 0,47 \quad \text{Una muestra consume ?}$$

$$\begin{array}{lcl} 4370 & \rightarrow & 0,47 \\ 1 & \rightarrow & X \end{array}$$

$$X = 0,000107551$$

\downarrow
tiempo ejecución del monitor

$$\text{Sobrecarga} = \frac{\text{Tiempo ejecución monitor}}{\text{Intervalo}}$$