Checklist de Tutorial de Profiling en Linux con `gprof` y `perf`

Requerimientos e instalación de herramientas
- [✅] Instalar gcc, make y las herramientas de desarrollo:

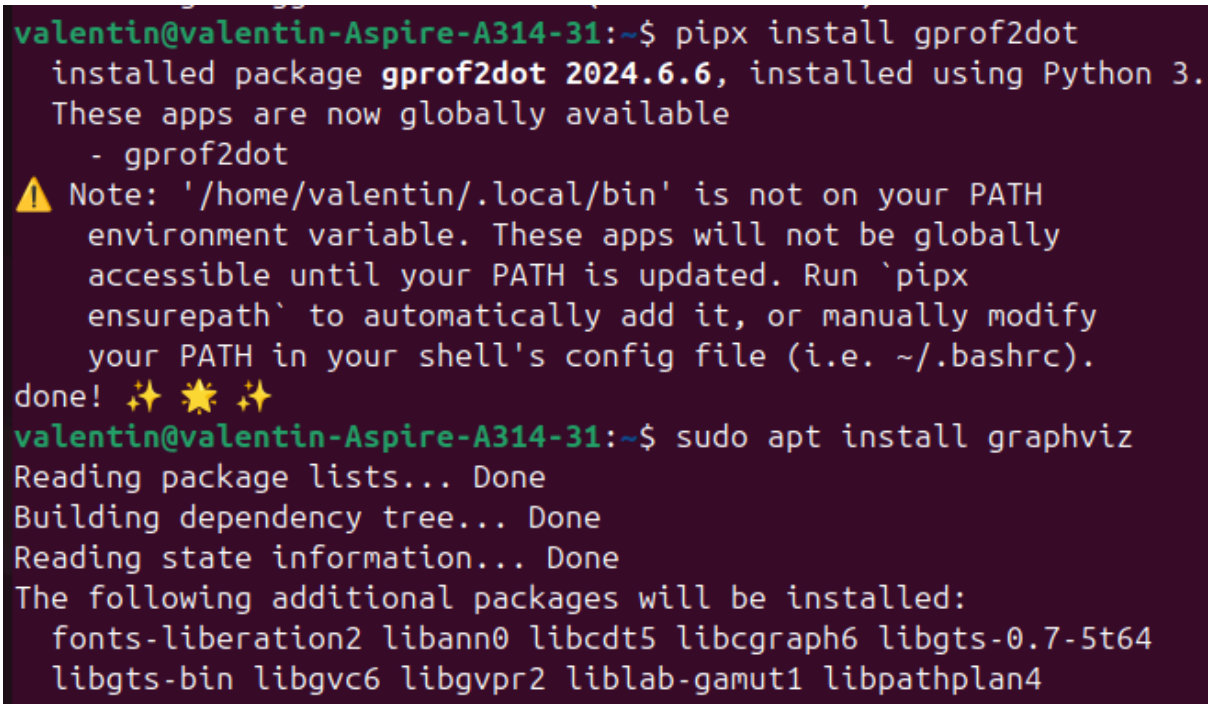
    sudo apt update
    sudo apt install build-essential

- [✅] Instalar linux-tools (para perf):

    sudo apt install linux-tools-common
    sudo apt install linux-tools-$(uname -r)

- [✅] Instalar gprof2dot y graphviz (para la visualización de gprof):

    pip install gprof2dot
    sudo apt install graphviz



Parte 1: Profiling con GPROF

Preparación del código
- [✅] Crear archivo `test_gprof.c` con el siguiente contenido:

```c
#include <stdio.h>

void new_func1(void);

void func1(void) {
```

```
        printf("\n Inside func1 \n");
        for (int i = 0; i < 0xffffffff; i++);
        new_func1();
}

static void func2(void) {
        printf("\n Inside func2 \n");
        for (int i = 0; i < 0xffffffaa; i++);
}

int main(void) {
        printf("\n Inside main()\n");
        for (int i = 0; i < 0xffffff; i++);
        func1();
        func2();
        return 0;
}
```

- [✅] Crear archivo `test_gprof_new.c` con la siguiente función:

```c
#include <stdio.h>

void new_func1(void) {
        printf("\n Inside new_func1()\n");
        for (int i = 0; i < 0xfffffffe; i++);
}
```

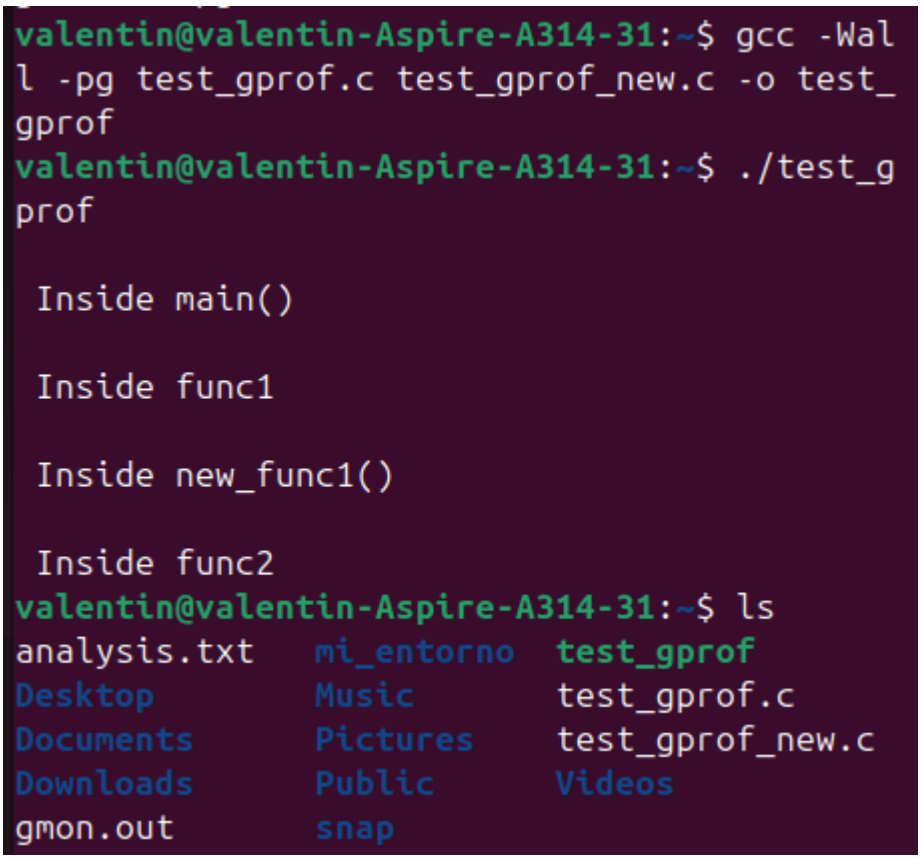
Compilación con soporte de profiling
- [✅] Compilar usando:

```
        gcc -Wall -pg test_gprof.c test_gprof_new.c -o test_gprof
```

Ejecución del programa
- [✅] Ejecutar:

```
        ./test_gprof
```

  Esto generará `gmon.out`.



Generar análisis de perfil
- [✅] Ejecutar:

```
gprof test_gprof gmon.out > analysis.txt
```

- ✅] Verificar que el archivo `analysis.txt` contiene:
  - [✅] Flat profile
  - [✅] Call graph

Personalización del análisis con flags (ejecutar cada uno y analizar)
- [✅] `gprof -a test_gprof gmon.out > analysis.txt`
  - Suprime funciones `static` como `func2`
  - [✅] Verificar que `func2` no aparece en el output

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 68.08     23.61    23.61        2    11.80    17.32  func1
 31.81     34.64    11.03        1    11.03    11.03  new_func1
  0.12     34.68     0.04                             main

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
           function is profiled, else blank.
```

- [ ] `gprof -b test_gprof gmon.out > analysis.txt`
  - Elimina explicaciones detalladas

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 35.39     11.95    11.95        1    11.95    22.84  func1
 32.25     22.84    10.89        1    10.89    10.89  func2
 32.25     33.73    10.89        1    10.89    10.89  new_func1
  0.12     33.77     0.04                             main
^L
                     Call graph
```

- [✅] `gprof -p -b test_gprof gmon.out > analysis.txt`
  - Solo muestra perfil plano

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
100.00     11.95    11.95        1    11.95    11.95  func1
analysis.txt (END)
```

Visualización con gprof2dot
- [✅] Instalar herramientas necesarias:

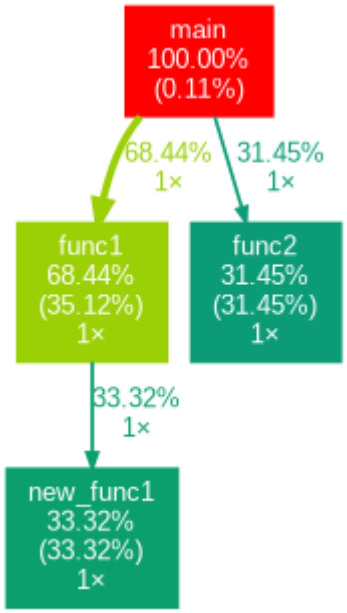
    pip install gprof2dot

    sudo apt install graphviz

- [✅] Generar gráfico:

    gprof test_gprof gmon.out | gprof2dot | dot -Tpng -o output.png

- [✅] Verificar que se generó `output.png`



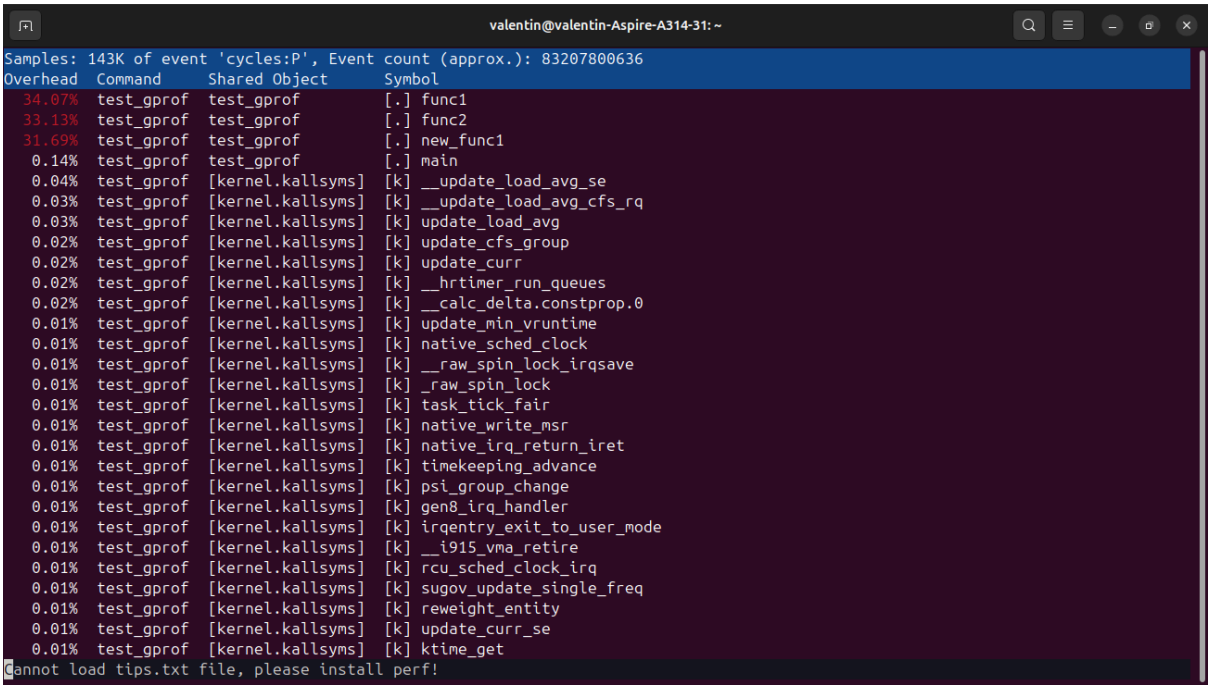
Parte 2: Profiling con `perf`

Ejecución de análisis con `perf`
- [✅] Ejecutar:

```
sudo perf record ./test_gprof
```

- [✅] Generar reporte interactivo:

```
sudo perf report
```



Limpieza opcional
- [✅] Eliminar archivos generados si se desea:

```
rm gmon.out test_gprof analysis.txt output.png
```

Fin del tutorial
- [✅] Confirmar comprensión de:
  - [✅] Perfil plano
  - [✅] Gráfico de llamadas
  - [✅] Flags para personalización
  - [✅] Visualización con `gprof2dot`
  - [✅] Análisis básico con `perf`