

LAB 2 Programming and Optimizing Tools Activities

Accounting Tools

1.1.2

a) Genero el ejecutable con optimización -O0

```
1 $ gcc -O0 pi.c -o pi.O0
```

b) Ejecuto el ejecutable pi.O0 y guardo el resultado en pi.0_time

```
1 $ /usr/bin/time -o pi.0_time ./pi.O0
2 $ cat pi.0_time
3
4 55.69user 0.18system 0:55.89elapsed 99%CPU (0avgtext+0avgdata
5 852maxresident)k 0inputs+0outputs (0major+64minor)pagefaults 0swaps
```

c) El %CPU es el tanto por ciento de CPU usada para ejecutar el programa

1.1.3

a) Genero los ejecutables correspondientes

```
1 $ gcc -O1 pi.c -o pi.O1
2 $ gcc -O2 pi.c -o pi.O2
3 $ gcc -O3 pi.c -o pi.O3
```

Ejecuto el ejecutable pi.O3 y guardo el resultado en pi.3_time

```
1 $ /usr/bin/time -o pi.3_time ./pi.O3
2 $ cat pi.3_time
3
4 17.80user 0.16system 0:17.99elapsed 99%CPU
5 0avgtext+0avgdata84maxresident)k 0inputs+0outputs (0major+66minor)pagefaults 0swaps
```

Vemos que el resultado es diferente en el tiempo usado por el usuario y el sistema, y por lo tanto, en el elapsed time. Esto es debido a que el compilador hace unas optimizaciones en el código y ahora se ejecuta más rápido.

b) i c) He añadido un tipo de formato diferente para ver el resultado como quiero para evaluar las diferencias mejor

```
1 $ /usr/bin/time -o pi.0_time -f "Elapsed time: %e User: %U System: %S " ./pi.O0
2 $ /usr/bin/time -o pi.3_time -f "Elapsed time: %e User: %U System: %S " ./pi.O3
```

Profiling Tools

1.2.4

a) i b) Vemos que la rutina más invocada por el programa es "SUBTRACT" con 2011 calls. Y el mayor tiempo de CPU consumido por una rutina del programa es 26.32% (supongo que `_aeabi_uidiv` no es una rutina del programa)

```
1 $ gcc -pg -g -O0 pi.c -o pi.O0.pg #Compilamos con -g y -pg para poder utilizar gprof
2 $ ./pi.O0.pg 1000 #Y ejecutamos antes de gprof siempre
3 $ gprof -b ./pi.O0.pg > pi.O0.pg.gprof.txt
4 $ cat pi.O0.pg.gprof.txt
5
6 Flat profile:
7 Each sample counts as 0.01 seconds.
8 % cumulative self self total
9 time seconds seconds calls ms/call ms/call name
10 47.37 0.27 0.27 __aeabi_uidiv
11 26.32 0.42 0.15 2011 0.07 0.07 SUBTRACT
12 24.56 0.56 0.14 3014 0.05 0.05 DIVIDE
13 1.75 0.57 0.01 1004 0.01 0.01 LONGDIV
14 0.00 0.57 0.00 1007 0.00 0.00 SET
15 0.00 0.57 0.00 1005 0.00 0.00 progress
16 0.00 0.57 0.00 2 0.00 0.00 MULTIPLY
17 0.00 0.57 0.00 1 0.00 300.00 calculate
18 0.00 0.57 0.00 1 0.00 0.00 epilog
```

c) Ejecutamos gprof con el flag `-l` para ver la información por línea de código. Por línea de código observamos que DIVIDE (línea 17 del programa) consume el mayor tiempo de CPU con 7.89 %

```
1 $ gprof -b -l ./pi.O0.pg > pi.O0.pg.gprof.txt
2
3 Each sample counts as 0.01 seconds.
4 % cumulative self self total
5 time seconds seconds calls Ts/call Ts/call name
6 47.37 0.27 0.27 __aeabi_uidiv
7 7.89 0.32 0.04 DIVIDE (pi.c:17 @ 10792)
8 7.89 0.36 0.04 SUBTRACT (pi.c:91 @ 109e8)
9 7.02 0.40 0.04 DIVIDE (pi.c:19 @ 107b6)
10 7.02 0.44 0.04 SUBTRACT (pi.c:94 @ 10a30)
11 6.14 0.47 0.04 SUBTRACT (pi.c:93 @ 10a16)
12 5.26 0.51 0.03 SUBTRACT (pi.c:89 @ 10a46)
13 3.51 0.53 0.02 DIVIDE (pi.c:18 @ 107a8)
14 3.51 0.55 0.02 DIVIDE (pi.c:15 @ 107d0)
15 1.75 0.56 0.01 DIVIDE (pi.c:20 @ 107c4)
16 1.75 0.56 0.01 LONGDIV (pi.c:37 @ 10838)
17 0.88 0.57 0.01 DIVIDE (pi.c:15 @ 1078c)
18 0.00 0.57 0.00 3014 0.00 0.00 DIVIDE (pi.c:9 @ 10778)
19 0.00 0.57 0.00 2011 0.00 0.00 SUBTRACT (pi.c:85 @ 109c8)
20 0.00 0.57 0.00 1007 0.00 0.00 SET (pi.c:78 @ 10990)
21 0.00 0.57 0.00 1005 0.00 0.00 progress (pi.c:181 @ 10c8c)
22 0.00 0.57 0.00 1004 0.00 0.00 LONGDIV (pi.c:25 @ 107f0)
23 0.00 0.57 0.00 2 0.00 0.00 MULTIPLY (pi.c:64 @ 1090c)
24 0.00 0.57 0.00 1 0.00 0.00 calculate (pi.c:126 @ 10b08)
25 0.00 0.57 0.00 1 0.00 0.00 epilog (pi.c:186 @ 10ca0)
```

d) No, en gprof no muestra nada del sistema igual que en Valgrind

1.2.5

a) i b)

```
1 $ gcc -pg -g -O3 pi.c -o pi.O3.pg
2 $ ./pi.O3.pg 1000
3 $ gprof -b ./pi.O3.pg > pi.O3.pg.gprof.txt
4 $ cat pi.O3.pg.gprof.txt
```

Solo aparece la funcion calculate, debido seguramente a que copia el código de todas las funciones a calculate que es donde se llaman, mejorando el tiempo de ejecución pues se hacen menos calls (inline). Ello conlleva tambien a que aumente el peso del ejecutable. Pues si hay dos llamadas de una función, el código de esta se ve duplicado.

time seconds seconds calls Ts/call Ts/call name

70.66 0.12 0.12 calculate

29.44 0.17 0.05 __aeabi_uidiv

He mirado por línea de código a ver que pasaba, y como se aprecia, no tienen ninguna llamada, por lo que reafirmo lo que dije anteriormente (inline de las funciones)

```
1 $ gprof -b -l ./pi.O3.pg > pi.O3.pg.gprof.txt
2
3 time seconds seconds calls Ts/call Ts/call name
4 29.44 0.05 0.05 __aeabi_uidiv
5 11.78 0.07 0.02 SUBTRACT (pi.c:91 @ 10ada)
6 8.83 0.09 0.02 SUBTRACT (pi.c:94 @ 10b80)
7 5.89 0.10 0.01 DIVIDE (pi.c:18 @ 10bd6)
8 5.89 0.11 0.01 DIVIDE (pi.c:20 @ 10bdc)
9 5.89 0.12 0.01 SUBTRACT (pi.c:93 @ 10aee)
10 5.89 0.13 0.01 SUBTRACT (pi.c:89 @ 10afa)
11 5.89 0.14 0.01 SUBTRACT (pi.c:91 @ 10b68)
12 2.94 0.14 0.01 DIVIDE (pi.c:18 @ 10b46)
13 2.94 0.15 0.01 DIVIDE (pi.c:20 @ 10b4c)
14 2.94 0.15 0.01 LONGDIV (pi.c:33 @ 10c1c)
15 2.94 0.16 0.01 LONGDIV (pi.c:35 @ 10c34)
16 2.94 0.16 0.01 LONGDIV (pi.c:37 @ 10c4c)
17 2.94 0.17 0.01 LONGDIV (pi.c:33 @ 10c50)
18 2.94 0.17 0.01 SUBTRACT (pi.c:89 @ 10b88)
```

1.2.6

a) Las diferencias en la columna de samples es debido a que oprofile utiliza el evento predeterminado para Linux llamado CYCLES, que cuenta los ciclos del procesador. La diferencia de este contador determina la frecuencia con la que ocurre un evento y por lo tanto, esto afecta al recuento de muestras de cada evento. De ahí las diferencias de recuento de las muestras (samples)

```
1 $ gcc -g -O0 pi.c -o pi.O0.g
2 $ perf --event=CPU_CYCLES:100000 ./pi.O0.g
3 $ opreport -l
4
5 CPU: ARM Cortex-A9, speed 999 MHz (estimated)
6 Counted CPU_CYCLES events (CPU cycle) with a unit mask of 0x00 (No unit mask) count
```

```

7 100000
8 samples % image name symbol name
9 130235 33.5958 pi.O0.g __acabi_uidiv
10 109775 28.3179 pi.O0.g SUBTRACT
11 94519 24.3824 pi.O0.g DIVIDE
12 34861 8.9928 pi.O0.g LONGDIV
13 13832 3.5681 pi.O0.g .divsi3_skip_div0_test
14 3594 0.9271 no-vmlinux /no-vmlinux
15 289 0.0746 pi.O0.g __divsi3
16 199 0.0513 libc-2.23.so memset
17 67 0.0173 libc-2.23.so vfprintf
18 39 0.0101 libc-2.23.so _IO_file_write@@GLIBC_2.4
19 34 0.0088 libc-2.23.so new_do_write
20 31 0.0080 libc-2.23.so putchar
21 22 0.0057 libc-2.23.so write
22 22 0.0057 pi.O0.g calculate
23 16 0.0041 libc-2.23.so _IO_file_overflow@@GLIBC_2.4
24 16 0.0041 libc-2.23.so _IO_file_xsputn@@GLIBC_2.4
25 16 0.0041 pi.O0.g epilog
26 14 0.0036 libc-2.23.so buffered_vfprintf
27 11 0.0028 libc-2.23.so strchrnul
28 10 0.0026 libc-2.23.so _IO_do_write@@GLIBC_2.4
29 10 0.0026 pi.O0.g MULTIPLY
30 8 0.0021 libc-2.23.so _IO_default_xsputn
31 7 0.0018 pi.O0.g SET
32 6 0.0015 libc-2.23.so __overflow
33 6 0.0015 libc-2.23.so _itoa_word
34 5 0.0013 pi.O0.g progress
35 4 0.0010 libc-2.23.so fputc
36 1 2.6e-04 ld-2.23.so _dl_relocate_object
37 1 2.6e-04 ld-2.23.so check_match
38 1 2.6e-04 ld-2.23.so open
39 1 2.6e-04 libc-2.23.so _IO_flush_all_lockp
40 1 2.6e-04 libc-2.23.so fwrite

```

```

1 $ perf --event=CPU_CYCLES:750000 ./pi.O0.g
2 $ opreport -l
3
4 CPU: ARM Cortex-A9, speed 999 MHz (estimated)
5 Counted CPU_CYCLES events (CPU cycle) with a unit mask of 0x00 (No unit mask) count
6 750000
7 samples % image name symbol name
8 16731 33.2955 pi.O0.g __acabi_uidiv
9 14284 28.4259 pi.O0.g SUBTRACT
10 12407 24.6905 pi.O0.g DIVIDE
11 4470 8.8955 pi.O0.g LONGDIV
12 1775 3.5323 pi.O0.g .divsi3_skip_div0_test
13 467 0.9294 no-vmlinux /no-vmlinux
14 41 0.0816 pi.O0.g __divsi3
15 26 0.0517 libc-2.23.so memset
16 11 0.0219 libc-2.23.so write
17 8 0.0159 libc-2.23.so vfprintf
18 4 0.0080 libc-2.23.so buffered_vfprintf
19 3 0.0060 libc-2.23.so _IO_file_write@@GLIBC_2.4
20 3 0.0060 libc-2.23.so new_do_write
21 3 0.0060 pi.O0.g calculate
22 2 0.0040 libc-2.23.so _IO_file_overflow@@GLIBC_2.4
23 2 0.0040 libc-2.23.so __overflow
24 2 0.0040 libc-2.23.so putchar
25 2 0.0040 pi.O0.g MULTIPLY
26 2 0.0040 pi.O0.g SET
27 2 0.0040 pi.O0.g epilog
28 1 0.0020 libc-2.23.so _IO_default_xsputn
29 1 0.0020 libc-2.23.so _IO_file_xsputn@@GLIBC_2.4
30 1 0.0020 libc-2.23.so fprintf
31 1 0.0020 libc-2.23.so strchrnul
32 1 0.0020 pi.O0.g progress

```

1.2.7

a) Las principales diferencias están en la columna samples y en las funciones de las que se han tomado muestras (samples). Con `opreport` y `-O3` vemos que las funciones `DIVIDE`, `SUBTRACT`, etc, no son mostradas. Esto es porque no hay llamadas hacia estas funciones, puesto que al haber hecho inline se ha copiado el código de las funciones donde se llamaban anteriormente. Por ello, con `opannotate` vemos que para `-O0` si vemos el recuento de muestras justo donde empieza cada una de las funciones nombradas anteriormente, pero no para `-O3`. En cambio si hay recuento del código de cada una de ellas. Porque el código igualmente se ejecuta. Por último, hay diferencias de recuentos en los "for" por una optimización del `-O3` sobre los bucles. Esto provoca una disminución de recuento sobre ellos.

```
1 $ opannotate --source pi.O0.g
2
3 Using /home/ubuntu/lab2_session/1.2/oprofile_data/samples/ for session-dir
4 /*
5  * Command line: opannotate --source pi.O0.g
6  *
7  * Interpretation of command line:
8  * Output annotated source file with samples
9  * Output all files
10 *
11 * CPU: ARM Cortex-A9, speed 999 MHz (estimated)
12 * Counted CPU_CYCLES events (CPU cycle) with a unit mask of 0x00 (No unit mask)
13 count 100000
14 */
15 /*
16 * Total samples for file : "/home/ubuntu/lab2_session/1.2/pi.c"
17 *
18 * 239215 61.7085
19 */
20 :#include <memory.h>
21 :#include <stdio.h>
22 :#include <stdlib.h>
23 :
24 :int N, N4;
25 :signed char a[25480], b[25480], c[25480];
26 :
27 :void DIVIDE( signed char *x, int n )
28 8 0.0021 : { /* DIVIDE total: 94519 24.3824 */
29 : int j, k;
30 : unsigned q, r, u;
31 : long v;
32 :
33 : r = 0;
34 23800 6.1395 : for( k = 0; k <= N4; k++ )
35 : {
36 21718 5.6024 : u = r * 10 + x[k];
37 20615 5.3179 : q = u/n;
38 14237 3.6726 : r = u - q * n;
39 14137 3.6468 : x[k] = q;
40 : }
41 4 0.0010 :}
42 :
43 :void LONGDIV( signed char *x, int n )
44 2 5.2e-04 : { /* LONGDIV total: 34861 8.9928 */
45 : int j, k;
46 : unsigned q, r, u;
47 : long v;
48 :
49 : if( n < 6553 )
50 : {
51 : r = 0;
```

```

52 2663 0.6870 : for( k = 0; k <= N4; k++ )
53 : {
54 2640 0.6810 : u = r * 10 + x[k];
55 2262 0.5835 : q = u / n;
56 1499 0.3867 : r = u - q * n;
57 1471 0.3795 : x[k] = q;
58 : }
59 : }
60 : else
61 : {
62 : r = 0;
63 6151 1.5867 : for( k = 0; k <= N4; k++ )
64 : {
65 2581 0.6658 : if( r < 6553 )
66 : {
67 4328 1.1165 : u = r * 10 + x[k];
68 1846 0.4762 : q = u / n;
69 1500 0.3869 : r = u - q * n;
70 : }
71 : else
72 : {
73 3320 0.8564 : v = (long) r * 10 + x[k];
74 1037 0.2675 : q = v / n;
75 940 0.2425 : r = v - q * n;
76 : }
77 2619 0.6756 : x[k] = q;
78 : }
79 : }
80 2 5.2e-04 : }
81 :
82 : void MULTIPLY( signed char *x, int n )
83 : { /* MULTIPLY total: 10 0.0026 */
84 : int j, k;
85 : unsigned q, r, u;
86 : long v;
87 : r = 0;
88 : for( k = N4; k >= 0; k-- )
89 : {
90 1 2.6e-04 : q = n * x[k] + r;
91 2 5.2e-04 : r = q / 10;
92 7 0.0018 : x[k] = q - r * 10;
93 : }
94 : }
95 :
96 : void SET( signed char *x, int n )
97 5 0.0013 : { /* SET total: 7 0.0018 */
98 : memset( x, 0, N4 + 1 );
99 1 2.6e-04 : x[0] = n;
100 1 2.6e-04 : }
101 :
102 :
103 : void SUBTRACT( signed char *x, signed char *y, signed char *z )
104 3 7.7e-04 : { /* SUBTRACT total: 109775 28.3179 */
105 : int j, k;
106 : unsigned q, r, u;
107 : long v;
108 24639 6.3559 : for( k = N4; k >= 1; k-- )
109 : {
110 46029 11.8738 : if( (x[k] = y[k] - z[k]) < 0 )
111 : {
112 25189 6.4978 : x[k] += 10;
113 13903 3.5865 : z[k-1]++;
114 : }
115 : }
116 8 0.0021 : if( (x[k] = y[k] - z[k]) < 0 )
117 : {
118 : x[k] += 10;
119 : }
120 4 0.0010 : }

```

```

121 :
122 :
123 :void calculate( void );
124 :void progress( void );
125 :void epilog( void );
126 :
127 :
128 :int main( int argc, char *argv[] )
129 :{
130 : N = 10000;
131 :
132 : if( argc > 1 )
133 : N = atoi(argv[1]);
134 :
135 : setbuf(stdout, NULL);
136 :
137 : calculate();
138 :
139 : epilog();
140 :
141 : return 0;
142 :}
143 :
144 :void calculate( void )
145 :{ /* calculate total: 22 0.0057 */
146 : int j;
147 :
148 : N4 = N + 4;
149 :
150 : SET( a, 0 );
151 : SET( b, 0 );
152 :
153 : 2 5.2e-04 : for( j = 2 * N4 + 1; j >= 3; j -= 2 )
154 : {
155 : 2 5.2e-04 : SET( c, 1 );
156 : 3 7.7e-04 : LONGDIV( c, j );
157 :
158 : 2 5.2e-04 : SUBTRACT( a, c, a );
159 : 3 7.7e-04 : DIVIDE( a, 25 );
160 :
161 : 2 5.2e-04 : SUBTRACT( b, c, b );
162 : 4 0.0010 : DIVIDE( b, 239 );
163 : 1 2.6e-04 : DIVIDE( b, 239 );
164 :
165 : 3 7.7e-04 : progress();
166 : }
167 :
168 : SET( c, 1 );
169 :
170 : SUBTRACT( a, c, a );
171 : DIVIDE( a, 5 );
172 :
173 : SUBTRACT( b, c, b );
174 : DIVIDE( b, 239 );
175 :
176 : MULTIPLY( a, 4 );
177 : SUBTRACT( a, a, b );
178 : MULTIPLY( a, 4 );
179 :
180 : progress();
181 :}
182 :
183 :/*
184 :
185 : N = 10000
186 : A = 0
187 : B = 0
188 : J = 2 * (N + 4) + 1
189 : FOR J = J TO 3 STEP -2

```

```

190 : A = (1 / J - A) / 5 ^ 2
191 : B = (1 / J - B) / 239 ^ 2
192 : NEXT J
193 : A = (1 - A) / 5
194 : B = (1 - B) / 239
195 : PI = (A * 4 - B) * 4
196 :
197 :*/
198 :
199 :void progress( void )
200 3 7.7e-04 : { /* progress total: 5 0.0013 */
201 : printf(".");
202 2 5.2e-04 :}
203 :
204 :void epilog( void )
205 : { /* epilog total: 16 0.0041 */
206 : int j;
207 :
208 : {
209 : fprintf( stdout, "\n3.");
210 4 0.0010 : for( j = 1; j <= N; j++ )
211 : {
212 5 0.0013 : fprintf( stdout, "%d", a[j]);
213 7 0.0018 : if( j % 5 == 0 )
214 : if( j % 50 == 0 )
215 : if( j % 250 == 0 )
216 : fprintf( stdout, "<%d>\n\n", j );
217 : else
218 : fprintf( stdout, "\n " );
219 : else
220 : fprintf( stdout, " " );
221 : }
222 : }
223 :}
224 :
225 opannotate (warning): unable to open for reading:
226 /build/glibc-XzEjT5/glibc-2.23/string/./sysdeps/arm/memset.S
227 /*
228 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/string/./sysdeps/arm/memset.S"
229 *
230 * 199 0.0513
231 */
232 /* memset total: 199 0.0513 */
233 opannotate (warning): unable to open for reading:
234 /build/glibc-XzEjT5/glibc-2.23/libio/fileops.c
235 /*
236 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/libio/fileops.c"
237 *
238 * 115 0.0297
239 */
240 /* _IO_do_write@@GLIBC_2.4 total: 10 0.0026 */
241 /* new_do_write total: 34 0.0088 */
242 /* _IO_file_overflow@@GLIBC_2.4 total: 16 0.0041 */
243 /* _IO_file_write@@GLIBC_2.4 total: 39 0.0101 */
244 /* _IO_file_xsputn@@GLIBC_2.4 total: 16 0.0041 */
245 opannotate (warning): unable to open for reading: /build/glibc-XzEjT5/glibc-2.23/stdiocommon/
246 vfprintf.c
247 /*
248 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/stdio-common/vfprintf.c"
249 *
250 * 79 0.0204
251 */
252 /* vfprintf total: 67 0.0173 */
253 /* buffered_vfprintf total: 14 0.0036 */
254 opannotate (warning): unable to open for reading:
255 /build/glibc-XzEjT5/glibc-2.23/libio/putchar.c
256 /*
257 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/libio/putchar.c"
258 *

```



```

259 * 24 0.0062
260 */
261 /* putchar total: 31 0.0080 */
262 opannotate (warning): unable to open for reading:
263 /build/glibc-XzEjT5/glibc-2.23/io/./sysdeps/unix/syscall-template.S
264 /*
265 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/io/./sysdeps/unix/syscalltemplate.
266 S"
267 *
268 * 23 0.0059
269 */
270 /* open total: 1 2.6e-04 */
271 /* write total: 22 0.0057 */
272 opannotate (warning): unable to open for reading:
273 /build/glibc-XzEjT5/glibc-2.23/libio/genops.c
274 /*
275 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/libio/genops.c"
276 *
277 * 15 0.0039
278 */
279 /* __overflow total: 6 0.0015 */
280 /* _IO_default_xsputn total: 8 0.0021 */
281 /* _IO_flush_all_lockp total: 1 2.6e-04 */
282 opannotate (warning): unable to open for reading:
283 /build/glibc-XzEjT5/glibc-2.23/string/strchrnul.c
284 /*
285 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/string/strchrnul.c"
286 *
287 * 11 0.0028
288 */
289 /* strchrnul total: 11 0.0028 */
290 opannotate (warning): unable to open for reading:
291 /build/glibc-XzEjT5/glibc-2.23/libio/libioP.h
292 /*
293 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/libio/libioP.h"
294 *
295 * 9 0.0023
296 */
297 opannotate (warning): unable to open for reading: /build/glibc-XzEjT5/glibc-2.23/stdiocommon/_
298 itoa.c
299 /*
300 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/stdio-common/_itoa.c"
301 *
302 * 6 0.0015
303 */
304 /* _itoa_word total: 6 0.0015 */
305 opannotate (warning): unable to open for reading:
306 /build/glibc-XzEjT5/glibc-2.23/libio/fputc.c
307 /*
308 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/libio/fputc.c"
309 *
310 * 3 7.7e-04
311 */
312 /* fputc total: 4 0.0010 */
313 opannotate (warning): unable to open for reading: /build/glibc-XzEjT5/glibc-2.23/stdiocommon/
314 printf-parse.h
315 /*
316 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/stdio-common/printf-parse.h"
317 *
318 * 2 5.2e-04
319 */
320 opannotate (warning): unable to open for reading: /build/glibc-XzEjT5/glibc-2.23/elf/dorel.
321 h
322 /*
323 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/elf/do-rel.h"
324 *
325 * 1 2.6e-04
326 */
327 opannotate (warning): unable to open for reading: /build/glibc-XzEjT5/glibc-2.23/elf/dllookup.

```

```

328 c
329 /*
330  * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/elf/dl-lookup.c"
331  *
332  * 1 2.6e-04
333  */
334 /* check_match total: 1 2.6e-04 */
335 %%%%%%%%%%
336 %%%%%%%%%%
337 $ gcc -g -O3 pi.c -o pi.O3.g
338 $ operf --event=CPU_CYCLES:100000 ./pi.O3.g
339 $ opreport -l
340 CPU: ARM Cortex-A9, speed 999 MHz (estimated)
341 Counted CPU_CYCLES events (CPU cycle) with a unit mask of 0x00 (No unit mask) count
342 100000
343 samples % image name symbol name
344 90344 69.4500 pi.O3.g calculate
345 20858 16.0341 pi.O3.g __acabi_uidiv
346 13123 10.0880 pi.O3.g .divsi3_skip_div0_test
347 4551 3.4985 no-vmlinux /no-vmlinux
348 481 0.3698 pi.O3.g __divsi3
349 367 0.2821 libc-2.23.so memset
350 75 0.0577 libc-2.23.so vfprintf
351 60 0.0461 libc-2.23.so putchar
352 37 0.0284 libc-2.23.so _IO_file_write@@GLIBC_2.4
353 28 0.0215 libc-2.23.so write
354 26 0.0200 libc-2.23.so new_do_write
355 23 0.0177 libc-2.23.so _IO_file_overflow@@GLIBC_2.4
356 14 0.0108 libc-2.23.so __memset_chk
357 12 0.0092 libc-2.23.so __fprintf_chk
358 12 0.0092 libc-2.23.so buffered_vfprintf
359 12 0.0092 libc-2.23.so strchrnul
360 10 0.0077 libc-2.23.so _IO_default_xsputn
361 10 0.0077 libc-2.23.so _itoa_word
362 9 0.0069 libc-2.23.so _IO_do_write@@GLIBC_2.4
363 9 0.0069 libc-2.23.so _IO_file_xsputn@@GLIBC_2.4
364 8 0.0061 libc-2.23.so __overflow
365 6 0.0046 libc-2.23.so ____GI_memset_from_thumb
366 4 0.0031 libc-2.23.so fputc
367 4 0.0031 pi.O3.g epilog
368 1 7.7e-04 ld-2.23.so _dl_relocate_object
369 1 7.7e-04 libc-2.23.so fwrite

```

```

1 $ opannotate --source pi.O3.g
2
3 Using /home/ubuntu/lab2_session/1.2/oprofile_data/samples/ for session-dir
4 /*
5  * Command line: opannotate --source pi.O3.g
6  *
7  * Interpretation of command line:
8  * Output annotated source file with samples
9  * Output all files
10  *
11  * CPU: ARM Cortex-A9, speed 999 MHz (estimated)
12  * Counted CPU_CYCLES events (CPU cycle) with a unit mask of 0x00 (No unit mask)
13  count 100000
14  */
15  /*
16  * Total samples for file : "/home/ubuntu/lab2_session/1.2/pi.c"
17  *
18  * 90339 69.4461
19  */
20 :#include <memory.h>
21 :#include <stdio.h>
22 :#include <stdlib.h>
23 :
24 :int N, N4;
25 :signed char a[25480], b[25480], c[25480];

```

```

26 :
27 :void DIVIDE( signed char *x, int n )
28 :{
29 : int j, k;
30 : unsigned q, r, u;
31 : long v;
32 :
33 : r = 0;
34 9401 7.2268 : for( k = 0; k <= N4; k++ )
35 : {
36 6593 5.0682 : u = r * 10 + x[k];
37 6196 4.7630 : q = u/n;
38 6221 4.7823 : r = u - q * n;
39 18625 14.3176 : x[k] = q;
40 : }
41 :}
42 :
43 :void LONGDIV( signed char *x, int n )
44 :{
45 : int j, k;
46 : unsigned q, r, u;
47 : long v;
48 :
49 2 0.0015 : if( n < 6553 )
50 : {
51 : r = 0;
52 805 0.6188 : for( k = 0; k <= N4; k++ )
53 : {
54 858 0.6596 : u = r * 10 + x[k];
55 1021 0.7849 : q = u / n;
56 45 0.0346 : r = u - q * n;
57 108 0.0830 : x[k] = q;
58 : }
59 : }
60 : else
61 : {
62 : r = 0;
63 1267 0.9740 : for( k = 0; k <= N4; k++ )
64 : {
65 2097 1.6120 : if( r < 6553 )
66 : {
67 2741 2.1071 : u = r * 10 + x[k];
68 888 0.6826 : q = u / n;
69 165 0.1268 : r = u - q * n;
70 : }
71 : else
72 : {
73 2 0.0015 : v = (long) r * 10 + x[k];
74 1185 0.9109 : q = v / n;
75 : r = v - q * n;
76 : }
77 204 0.1568 : x[k] = q;
78 : }
79 : }
80 : }
81 :
82 :void MULTIPLY( signed char *x, int n )
83 :{
84 : int j, k;
85 : unsigned q, r, u;
86 : long v;
87 : r = 0;
88 2 0.0015 : for( k = N4; k >= 0; k-- )
89 : {
90 1 7.7e-04 : q = n * x[k] + r;
91 : r = q / 10;
92 1 7.7e-04 : x[k] = q - r * 10;
93 : }
94 :}

```

```

95 :
96 :void SET( signed char *x, int n )
97 :{
98 : memset( x, 0, N4 + 1 );
99 1 7.7e-04 : x[0] = n;
100 :}
101 :
102 :
103 :void SUBTRACT( signed char *x, signed char *y, signed char *z )
104 :{
105 : int j, k;
106 : unsigned q, r, u;
107 : long v;
108 4409 3.3893 : for( k = N4; k >= 1; k-- )
109 : {
110 11613 8.9272 : if( (x[k] = y[k] - z[k]) < 0 )
111 : {
112 2630 2.0218 : x[k] += 10;
113 13241 10.1787 : z[k-1]++;
114 :}
115 :}
116 : if( (x[k] = y[k] - z[k]) < 0 )
117 : {
118 4 0.0031 : x[k] += 10;
119 :}
120 :}
121 :
122 :
123 :void calculate( void );
124 :void progress( void );
125 :void epilog( void );
126 :
127 :
128 :int main( int argc, char *argv[] )
129 :{
130 : N = 10000;
131 :
132 : if( argc > 1 )
133 : N = atoi(argv[1]);
134 :
135 : setbuf(stdout, NULL);
136 :
137 : calculate();
138 :
139 : epilog();
140 :
141 : return 0;
142 :}
143 :
144 :void calculate( void )
145 :{
146 : int j;
147 :
148 : N4 = N + 4; /* calculate total: 90344 69.4500 */
149 :
150 : SET( a, 0 );
151 : SET( b, 0 );
152 :
153 9 0.0069 : for( j = 2 * N4 + 1; j >= 3; j -= 2 )
154 : {
155 : SET( c, 1 );
156 : LONGDIV( c, j );
157 :
158 : SUBTRACT( a, c, a );
159 : DIVIDE( a, 25 );
160 :
161 : SUBTRACT( b, c, b );
162 : DIVIDE( b, 239 );
163 : DIVIDE( b, 239 );

```

```

164 :
165 : progress();
166 : }
167 :
168 : SET( c, 1 );
169 :
170 : SUBTRACT( a, c, a );
171 : DIVIDE( a, 5 );
172 :
173 : SUBTRACT( b, c, b );
174 : DIVIDE( b, 239 );
175 :
176 : MULTIPLY( a, 4 );
177 : SUBTRACT( a, a, b );
178 : MULTIPLY( a, 4 );
179 :
180 : progress();
181 : }
182 :
183 :/*
184 :
185 : N = 10000
186 : A = 0
187 : B = 0
188 : J = 2 * (N + 4) + 1
189 : FOR J = J TO 3 STEP -2
190 : A = (1 / J - A) / 5 ^ 2
191 : B = (1 / J - B) / 239 ^ 2
192 : NEXT J
193 : A = (1 - A) / 5
194 : B = (1 - B) / 239
195 : PI = (A * 4 - B) * 4
196 :
197 :*/
198 :
199 :void progress( void )
200 :{
201 : printf( "." );
202 : }
203 :
204 :void epilog( void )
205 :{ /* epilog total: 4 0.0031 */
206 : int j;
207 :
208 : {
209 : fprintf( stdout, "\n3." );
210 : for( j = 1; j <= N; j++ )
211 : {
212 : fprintf( stdout, "%d", a[j] );
213 : 2 0.0015 : if( j % 5 == 0 )
214 : 2 0.0015 : if( j % 50 == 0 )
215 : if( j % 250 == 0 )
216 : fprintf( stdout, " <%d>\n\n ", j );
217 : else
218 : fprintf( stdout, "\n " );
219 : else
220 : fprintf( stdout, " " );
221 : }
222 : }
223 : }
224 :
225 :opannotate (warning): unable to open for reading:
226 :/build/glibc-XzEjT5/glibc-2.23/string/../../sysdeps/arm/memset.S
227 :/*
228 : * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/string/../../sysdeps/arm/memset.S"
229 : *
230 : * 367 0.2821
231 : */
232 :/* memset total: 367 0.2821 */

```

```

233 opannotate (warning): unable to open for reading:
234 /build/glibc-XzEjT5/glibc-2.23/libio/fileops.c
235 /*
236 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/libio/fileops.c"
237 *
238 * 104 0.0799
239 */
240 /* _IO_do_write@@GLIBC_2.4 total: 9 0.0069 */
241 /* new_do_write total: 26 0.0200 */
242 /* _IO_file_overflow@@GLIBC_2.4 total: 23 0.0177 */
243 /* _IO_file_write@@GLIBC_2.4 total: 37 0.0284 */
244 /* _IO_file_xsputn@@GLIBC_2.4 total: 9 0.0069 */
245 opannotate (warning): unable to open for reading: /build/glibc-XzEjT5/glibc-2.23/stdiocommon/
246 vfprintf.c
247 /*
248 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/stdio-common/vfprintf.c"
249 *
250 * 86 0.0661
251 */
252 /* vfprintf total: 75 0.0577 */
253 /* buffered_vfprintf total: 12 0.0092 */
254 opannotate (warning): unable to open for reading:
255 /build/glibc-XzEjT5/glibc-2.23/libio/putchar.c
256 /*
257 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/libio/putchar.c"
258 *
259 * 51 0.0392
260 */
261 /* putchar total: 60 0.0461 */
262 opannotate (warning): unable to open for reading:
263 /build/glibc-XzEjT5/glibc-2.23/io/./sysdeps/unix/syscall-template.S
264 /*
265 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/io/./sysdeps/unix/syscalltemplate.
266 S"
267 *
268 * 28 0.0215
269 */
270 /* write total: 28 0.0215 */
271 opannotate (warning): unable to open for reading: /
272 build/glibc-XzEjT5/glibc-2.23/libio/genops.c
273 /*
274 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/libio/genops.c"
275 *
276 * 18 0.0138
277 */
278 /* __overflow total: 8 0.0061 */
279 /* _IO_default_xsputn total: 10 0.0077 */
280 opannotate (warning): unable to open for reading:
281 /build/glibc-XzEjT5/glibc-2.23/debug/memset_chk.c
282 /*
283 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/debug/memset_chk.c"
284 *
285 * 14 0.0108
286 */
287 /* __memset_chk total: 14 0.0108 */
288 opannotate (warning): unable to open for reading:
289 /build/glibc-XzEjT5/glibc-2.23/string/strchrnul.c
290 /*
291 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/string/strchrnul.c"
292 *
293 * 12 0.0092
294 */
295 /* strchrnul total: 12 0.0092 */
296 opannotate (warning): unable to open for reading: /build/glibc-XzEjT5/glibc-2.23/stdiocommon/_
297 itoa.c
298 /*
299 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/stdio-common/_itoa.c"
300 *
301 * 10 0.0077

```

```

302 */
303 /* _itoa_word total: 10 0.0077 */
304 opannotate (warning): unable to open for reading: /
305 /build/glibc-XzEjT5/glibc-2.23/libio/libioP.h
306 */
307 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/libio/libioP.h"
308 *
309 * 10 0.0077
310 */
311 /*
312 * Total samples for file : "/usr/include/arm-linux-gnueabi/bits/string3.h"
313 *
314 * 9 0.0069
315 */
316 :/* Copyright (C) 2004-2016 Free Software Foundation, Inc.
317 : This file is part of the GNU C Library.
318 :
319 : The GNU C Library is free software; you can redistribute it and/or
320 : modify it under the terms of the GNU Lesser General Public
321 : License as published by the Free Software Foundation; either
322 : version 2.1 of the License, or (at your option) any later version.
323 :
324 : The GNU C Library is distributed in the hope that it will be useful,
325 : but WITHOUT ANY WARRANTY; without even the implied warranty of
326 : MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
327 : the GNU
328 : Lesser General Public License for more details.
329 :
330 : You should have received a copy of the GNU Lesser General Public
331 : License along with the GNU C Library; if not, see
332 : <http://www.gnu.org/licenses/>. */
333 :
334 :#ifndef _STRING_H
335 :# error "Never use <bits/string3.h> directly; include <string.h> instead."
336 :#endif
337 :
338 :#if !_GNUC_PREREQ (5,0)
339 : __warndecl (__warn_memset_zero_len,
340 : "memset used with constant zero length parameter; this could be due to
341 : transposed parameters");
342 :#endif
343 :
344 :#ifndef __cplusplus
345 : /* XXX This is temporarily. We should not redefine any of the symbols
346 : : and instead integrate the error checking into the original
347 : : definitions. */
348 :# undef memcpy
349 :# undef memmove
350 :# undef memset
351 :# undef strcat
352 :# undef strcpy
353 :# undef strncat
354 :# undef strncpy
355 :# ifdef __USE_GNU
356 :# undef memcpy
357 :# undef stpcpy
358 :# endif
359 :# ifdef __USE_MISC
360 :# undef bcopy
361 :# undef bzero
362 :# endif
363 :#endif
364 :
365 :
366 : __fortify_function void *
367 : __NTH (memcpy (void * __restrict __dest, const void * __restrict __src,
368 : size_t __len))
369 : {
370 : return __builtin_memcpy_chk (__dest, __src, __len, __bos0 (__dest));

```

```

371 :}
372 :
373 :__fortify_function void *
374 :__NTH (memmove (void * __dest, const void * __src, size_t __len))
375 :{
376 : return __builtin__memmove_chk (__dest, __src, __len, __bos0 (__dest));
377 :}
378 :
379 :#ifdef __USE_GNU
380 :__fortify_function void *
381 :__NTH (memcpy (void * __restrict __dest, const void * __restrict __src,
382 : size_t __len))
383 :{
384 : return __builtin__memcpy_chk (__dest, __src, __len, __bos0 (__dest));
385 :}
386 :#endif
387 :
388 :
389 :/* The first two tests here help to catch a somewhat common problem
390 : where the second and third parameter are transposed. This is
391 : especially problematic if the intended fill value is zero. In this
392 : case no work is done at all. We detect these problems by referring
393 : non-existing functions. */
394 :__fortify_function void *
395 :__NTH (memset (void * __dest, int __ch, size_t __len))
396 :{
397 : /* GCC-5.0 and newer implements these checks in the compiler, so we don't
398 : need them here. */
399 :#if !__GNUC_PREREQ (5,0)
400 : if (__builtin_constant_p (__len) && __len == 0
401 : && (!__builtin_constant_p (__ch) || __ch != 0))
402 : {
403 : __warn_memset_zero_len ();
404 : return __dest;
405 : }
406 :#endif
407 : 9.0069 : return __builtin__memset_chk (__dest, __ch, __len, __bos0 (__dest));
408 :}
409 :
410 :#ifdef __USE_MISC
411 :__fortify_function void
412 :__NTH (bcopy (const void * __src, void * __dest, size_t __len))
413 :{
414 : (void) __builtin__memmove_chk (__dest, __src, __len, __bos0 (__dest));
415 :}
416 :
417 :__fortify_function void
418 :__NTH (bzero (void * __dest, size_t __len))
419 :{
420 : (void) __builtin__memset_chk (__dest, '\0', __len, __bos0 (__dest));
421 :}
422 :#endif
423 :
424 :__fortify_function char *
425 :__NTH (strcpy (char * __restrict __dest, const char * __restrict __src))
426 :{
427 : return __builtin__strcpy_chk (__dest, __src, __bos (__dest));
428 :}
429 :
430 :#ifdef __USE_GNU
431 :__fortify_function char *
432 :__NTH (stpcpy (char * __restrict __dest, const char * __restrict __src))
433 :{
434 : return __builtin__stpcpy_chk (__dest, __src, __bos (__dest));
435 :}
436 :#endif
437 :
438 :
439 :__fortify_function char *

```



```

440 : __NTH (strncpy (char *__restrict __dest, const char *__restrict __src,
441 : size_t __len))
442 : {
443 : return __builtin___strncpy_chk (__dest, __src, __len, __bos (__dest));
444 : }
445 :
446 :// XXX We have no corresponding builtin yet.
447 :extern char *__stpncpy_chk (char *__dest, const char *__src, size_t __n,
448 : size_t __destlen) __THROW;
449 :extern char *__REDIRECT_NTH (__stpncpy_alias, (char *__dest, const char
450 : *__src,
451 : size_t __n), stpncpy);
452 :
453 :__fortify_function char *
454 :__NTH (stpncpy (char *__dest, const char *__src, size_t __n))
455 : {
456 : if (__bos (__dest) != (size_t) -1
457 : && (!__builtin_constant_p (__n) || __n > __bos (__dest)))
458 : return __stpncpy_chk (__dest, __src, __n, __bos (__dest));
459 : return __stpncpy_alias (__dest, __src, __n);
460 : }
461 :
462 :
463 :__fortify_function char *
464 :__NTH (strcat (char *__restrict __dest, const char *__restrict __src))
465 : {
466 : return __builtin___strcat_chk (__dest, __src, __bos (__dest));
467 : }
468 :
469 :
470 :__fortify_function char *
471 :__NTH (strncat (char *__restrict __dest, const char *__restrict __src,
472 : size_t __len))
473 : {
474 : return __builtin___strncat_chk (__dest, __src, __len, __bos (__dest));
475 : }
476 :opannotate (warning): unable to open for reading:
477 :/build/glibc-XzEjT5/glibc-2.23/debug/fprintf_chk.c
478 :/*
479 : * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/debug/fprintf_chk.c"
480 : *
481 : * 9 0.0069
482 : */
483 :/* __fprintf_chk total: 12 0.0092 */
484 :opannotate (warning): unable to open for reading:
485 :/build/glibc-XzEjT5/glibc-2.23/libio/fputc.c
486 :/*
487 : * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/libio/fputc.c"
488 : *
489 : * 4 0.0031
490 : */
491 :/* fputc total: 4 0.0031 */
492 :opannotate (warning): unable to open for reading:
493 :/build/glibc-XzEjT5/glibc-2.23/debug/../libio/libioP.h
494 :/*
495 : * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/debug/../libio/libioP.h"
496 : *
497 : * 3 0.0023
498 : */
499 :opannotate (warning): unable to open for reading: /build/glibc-XzEjT5/glibc-2.23/stdiocommon/
500 :printf-parse.h
501 :/*
502 : * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/stdio-common/printf-parse.h"
503 : *
504 : * 1 7.7e-04
505 : */
506 :opannotate (warning): unable to open for reading:
507 :/build/glibc-XzEjT5/glibc-2.23/elf/../sysdeps/arm/dl-machine.h
508 :/*

```

```

509 * Total samples for file : "/build/glibc-XzEjT5/glibc-2.23/elf/./sysdeps/arm/dl-machine.h"
510 *
511 * 1 7.7e-04
512 */

```

1.2.8

a) Observamos que, dependiendo de la frecuencia que le pongamos, se ejecutan más samples, aun así el cuello de botella que genera cada rutina más o menos es el mismo. PD: El máximo que nos dejaba poner como frecuencia eran 32200.

```

1 $ sudo perf record -e cycles -F 32200 ./pi.O0.g 1000
2 $ sudo perf report --stdio -n
3
4 # To display the perf.data header info, please use --header/--hea
5 ##
6 # Total Lost Samples: 0
7 #
8 # Samples: 25K of event 'cycles'
9 # Event count (approx.): 514641219
10 #
11 # Overhead Samples Command Shared Object Symbol
12 # .....
13 #
14 33.81% 8499 pi.O0.g pi.O0.g [.] __udivs
15 25.59% 6429 pi.O0.g pi.O0.g [.] SUBTRAC
16 22.01% 5532 pi.O0.g pi.O0.g [.] DIVIDE
17 8.11% 2041 pi.O0.g pi.O0.g [.] LONGDIV
18 2.29% 580 pi.O0.g [kernel.kallsyms] [k] queue_w
19 0.96% 242 pi.O0.g [kernel.kallsyms] [k] __do_so
20 0.74% 186 pi.O0.g [kernel.kallsyms] [k] v7_dma_
21 0.33% 84 pi.O0.g [kernel.kallsyms] [k] l2c210_
22 0.28% 71 pi.O0.g [kernel.kallsyms] [k] tcp_ack
23 0.25% 64 pi.O0.g [kernel.kallsyms] [k] _raw_sp
24 0.25% 63 pi.O0.g libc-2.23.so [.] vfprintf
25 0.25% 62 pi.O0.g [kernel.kallsyms] [k] n_tty_w
26 0.19% 48 pi.O0.g [kernel.kallsyms] [k] tty_wri
27 0.15% 38 pi.O0.g [kernel.kallsyms] [k] ip_rcv
28 0.13% 32 pi.O0.g [kernel.kallsyms] [k] tcp_v4_
29 0.12% 30 pi.O0.g [kernel.kallsyms] [k] __netde
30 0.11% 29 pi.O0.g [kernel.kallsyms] [k] vector_
31 0.11% 28 pi.O0.g [kernel.kallsyms] [k] __netif
32 0.09% 23 pi.O0.g [kernel.kallsyms] [k] dma_cac
33 0.09% 22 pi.O0.g [kernel.kallsyms] [k] gem_rx_
34 0.09% 22 pi.O0.g [kernel.kallsyms] [k] vfs_wri
35 0.08% 20 pi.O0.g libc-2.23.so [.] __GI__
36 0.08% 20 pi.O0.g libc-2.23.so [.] new_do_
37 0.08% 19 pi.O0.g [kernel.kallsyms] [k] skb_rel
38 0.07% 19 pi.O0.g [kernel.kallsyms] [k] tty_par
39 0.07% 17 pi.O0.g libc-2.23.so [.] putchar
40 0.07% 17 pi.O0.g pi.O0.g [.] calcula
41 0.07% 17 pi.O0.g [kernel.kallsyms] [k] __l2c21
42 0.07% 17 pi.O0.g [kernel.kallsyms] [k] macb_po
43 0.07% 17 pi.O0.g [kernel.kallsyms] [k] __do_di
44 0.07% 17 pi.O0.g libc-2.23.so [.] buffere
45 0.06% 16 pi.O0.g [kernel.kallsyms] [k] eth_typ
46 0.06% 16 pi.O0.g [kernel.kallsyms] [k] tty_ins
47 0.06% 16 pi.O0.g [kernel.kallsyms] [k] __fdget
48 0.06% 16 pi.O0.g libc-2.23.so [.] _IO_fil
49 0.06% 15 pi.O0.g [kernel.kallsyms] [k] tcp_rcv
50 0.06% 15 pi.O0.g libc-2.23.so [.] _IO_fil
51 0.06% 15 pi.O0.g [kernel.kallsyms] [k] mmioset

```

```

52 0.06% 14 pi.O0.g [kernel.kallsyms] [k] skb_put
53 0.06% 14 pi.O0.g [kernel.kallsyms] [k] kfree
54 :
55 #
56 # (Cannot load tips.txt file, please install perf!)
57 #

```

```

1 $ sudo perf record -e cycles -F 5000 ./pi.O0.g 1000
2 $ sudo perf report --stdio -n
3 # To display the perf.data header info, please use --header/--header-only options.
4 ##
5 # Total Lost Samples: 0
6 #
7 # Samples: 3K of event 'cycles'
8 # Event count (approx.): 434938242
9 #
10 # Overhead Samples Command Shared Object Symbol
11 # .....
12 #
13 34.17% 1117 pi.O0.g pi.O0.g [.] __udivsi3
14 26.11% 854 pi.O0.g pi.O0.g [.] SUBTRACT
15 21.44% 701 pi.O0.g pi.O0.g [.] DIVIDE
16 7.09% 232 pi.O0.g pi.O0.g [.] LONGDIV
17 1.70% 65 pi.O0.g [kernel.kallsyms] [k] queue_work_on
18 1.48% 52 pi.O0.g [kernel.kallsyms] [k] __do_softirq
19 0.88% 31 pi.O0.g [kernel.kallsyms] [k] v7_dma_inv_range
20 0.43% 16 pi.O0.g [kernel.kallsyms] [k] n_tty_write
21 0.31% 11 pi.O0.g [kernel.kallsyms] [k] tcp_ack
22 0.30% 11 pi.O0.g [kernel.kallsyms] [k] l2c210_inv_range
23 0.22% 9 pi.O0.g [kernel.kallsyms] [k] _raw_spin_unlock_irqrestore
24 0.22% 8 pi.O0.g [kernel.kallsyms] [k] tcp_v4_rev
25 0.22% 8 pi.O0.g [kernel.kallsyms] [k] ip_rcv
26 0.21% 9 pi.O0.g libc-2.23.so [.] vfprintf
27 0.19% 7 pi.O0.g [kernel.kallsyms] [k] dma_cache_maint_page
28 0.15% 5 pi.O0.g libc-2.23.so [.] memset
29 0.14% 5 pi.O0.g [kernel.kallsyms] [k] ip_local_deliver
30 0.14% 5 pi.O0.g [kernel.kallsyms] [k] gem_rx_refill
31 0.14% 5 pi.O0.g [kernel.kallsyms] [k] tty_write
32 0.12% 4 pi.O0.g [kernel.kallsyms] [k] __kfree_skb
33 0.12% 4 pi.O0.g [kernel.kallsyms] [k] __netdev_alloc_skb
34 0.11% 3 pi.O0.g [kernel.kallsyms] [k] filemap_map_pages
35 0.11% 4 pi.O0.g [kernel.kallsyms] [k] gem_rx
36 0.11% 4 pi.O0.g [kernel.kallsyms] [k] __l2c210_cache_sync
37 0.09% 3 pi.O0.g [kernel.kallsyms] [k] vfs_write
38 0.09% 3 pi.O0.g libc-2.23.so [.] putchar
39 0.09% 3 pi.O0.g [kernel.kallsyms] [k] __rcu_read_unlock
40 0.09% 3 pi.O0.g [kernel.kallsyms] [k] vector_swi
41 0.09% 3 pi.O0.g [kernel.kallsyms] [k] tcp_v4_early_demux
42 0.09% 3 pi.O0.g [kernel.kallsyms] [k] __dma_page_dev_to_cpu
43 0.09% 3 pi.O0.g libc-2.23.so [.] new_do_write
44 0.09% 3 pi.O0.g [kernel.kallsyms] [k] mod_timer
45 0.08% 3 pi.O0.g [kernel.kallsyms] [k] __netif_receive_skb_core
46 0.08% 3 pi.O0.g [kernel.kallsyms] [k] tcp_xmit_recovery
47 0.08% 3 pi.O0.g [kernel.kallsyms] [k] __fdget_pos
48 0.08% 3 pi.O0.g [kernel.kallsyms] [k] __memzero
49 0.08% 3 pi.O0.g [kernel.kallsyms] [k] netif_receive_skb_internal
50 0.08% 3 pi.O0.g [kernel.kallsyms] [k] __vfs_write
51 0.07% 3 pi.O0.g [kernel.kallsyms] [k] __tty_buffer_request_room
52 0.06% 2 pi.O0.g pi.O0.g [.] progress

```

1.2.9

a) Vemos como para la misma frecuencia se ejecutan muchos menos samples, por lo que comentamos en ejercicios anteriores. Nos ahorramos las llamadas a las funciones que han le han hecho inline. Por lo que nos ahorramos ese recuento de samples. Tambien por las optimizaciones de los bucles. EL cuello de botella, sin embargo, es mayor en calculate que antes, por el mismo motivo (inline) al ejecutar el código de las funciones, que se les ha hecho inline, dentro de la rutina calculate, aumenta el código a ejecutar y por lo tanto genera más cuello de botella.

```
1 $ sudo perf record -e cycles -F 32200 ./pi.O3.g 1000
2 $ sudo perf report --stdio -n
3
4 # To display the perf.data header info, please use --header/--header-only options.
5 ##
6 # Total Lost Samples: 0
7 #
8 # Samples: 8K of event 'cycles'
9 # Event count (approx.): 166145838
10 #
11 # Overhead Samples Command Shared Object Symbol
12 # .....
13 #
14 61.88% 5061 pi.O3.g pi.O3.g [.] calculate
15 23.55% 1925 pi.O3.g pi.O3.g [.] __udivsi3
16 5.40% 461 pi.O3.g [kernel.kallsyms] [k] queue_work_on
17 0.72% 62 pi.O3.g [kernel.kallsyms] [k] n_tty_write
18 0.61% 53 pi.O3.g [kernel.kallsyms] [k] _raw_spin_unlock_irqrestore
19 0.45% 42 pi.O3.g libc-2.23.so [.] vfprintf
20 0.38% 33 pi.O3.g [kernel.kallsyms] [k] tty_write
21 0.33% 27 pi.O3.g libc-2.23.so [.] memset
22 0.29% 24 pi.O3.g libc-2.23.so [.] __GI___libc_write
23 0.28% 25 pi.O3.g libc-2.23.so [.] new_do_write
24 0.22% 18 pi.O3.g libc-2.23.so [.] putchar
25 0.21% 18 pi.O3.g [kernel.kallsyms] [k] vfs_write
26 0.21% 18 pi.O3.g [kernel.kallsyms] [k] vector_swi
27 0.19% 18 pi.O3.g [kernel.kallsyms] [k] v7_dma_inv_range
28 0.18% 17 pi.O3.g libc-2.23.so [.] _IO_file_overflow@@GLIBC_2.4
29 0.18% 15 pi.O3.g [kernel.kallsyms] [k] mutex_lock
30 0.18% 15 pi.O3.g libc-2.23.so [.] _IO_file_write@@GLIBC_2.4
31 0.16% 15 pi.O3.g [kernel.kallsyms] [k] __do_softirq
32 0.14% 14 pi.O3.g [kernel.kallsyms] [k] tcp_ack
33 0.13% 11 pi.O3.g [kernel.kallsyms] [k] process_echoes
34 0.11% 9 pi.O3.g [kernel.kallsyms] [k] sys_write
35 0.10% 9 pi.O3.g [kernel.kallsyms] [k] __tty_buffer_request_room
36 0.10% 10 pi.O3.g [kernel.kallsyms] [k] fsnotify
37 0.10% 11 pi.O3.g libc-2.23.so [.] buffered_vfprintf
38 0.07% 7 pi.O3.g [kernel.kallsyms] [k] tty_insert_flip_string_fixed_flag
39 0.07% 6 pi.O3.g [kernel.kallsyms] [k] v7_flush_icache_all
40 0.07% 6 pi.O3.g [kernel.kallsyms] [k] rw_verify_area
41 0.07% 6 pi.O3.g [kernel.kallsyms] [k] __fget_light
42 0.07% 6 pi.O3.g [kernel.kallsyms] [k] mmiocpy
43 0.07% 6 pi.O3.g [kernel.kallsyms] [k] add_wait_queue
44 0.07% 6 pi.O3.g [kernel.kallsyms] [k] tty_paranoia_check
45 0.07% 6 pi.O3.g [kernel.kallsyms] [k] tty_write_lock
46 0.07% 7 pi.O3.g [kernel.kallsyms] [k] get_seconds
47 0.07% 6 pi.O3.g [kernel.kallsyms] [k] tty_ldisc_ref_wait
48 0.06% 26 pi.O3.g [kernel.kallsyms] [k] _raw_spin_unlock_irq
49 0.06% 5 pi.O3.g [kernel.kallsyms] [k] pty_write
50 0.06% 5 pi.O3.g [kernel.kallsyms] [k] mutex_trylock
51 0.06% 5 pi.O3.g [kernel.kallsyms] [k] mmioset
52 0.06% 5 pi.O3.g [kernel.kallsyms] [k] ldsem_up_read
53 0.06% 5 pi.O3.g [kernel.kallsyms] [k] filemap_map_pages
```

1.2.10

a)

```
1 $ gcc -O0 pi_times.c -o pi_times
2 $ ./pi_times
3 user 0.560000 segons, system: 0.000000 segons
```

```
1 $ /usr/bin/time -o pi_time_10 -f "Elapsed time: %e User: %U System: %S " ./pitime.O0 1000
2 Elapsed time: 0.62 User: 0.59 System: 0.02
3
4 $ /usr/bin/time -o pi_O0 -f "Elapsed time: %e User: %U System: %S " ./pi.O0 1000
5 Elapsed time: 0.59 User: 0.55 System: 0.03
```

```
1 $ perf stat ./pi_times 1000 > out_pitimes_perfstat
2
3 Performance counter stats for './pi_times 1000':
4 586.056771 task-clock (msec) # 0.998 CPUs utilized
5 1 context-switches # 0.002 K/sec
6 0 cpu-migrations # 0.000 K/sec
7 36 page-faults # 0.061 K/sec
8 390,693,570 cycles # 0.667 GHz
9 397,726,420 instructions # 1.02 insn per cycle
10 20,442,204 branches # 34.881 M/sec
11 4,377,279 branch-misses # 21.41% of all branches
12 0.587480439 seconds time elapsed
```

Automatization and Data Managament tools

exec_program.py

```
1 import subprocess
2 import time
3 import sys
4 import re
5
6 if __name__ == "__main__":
7
8     program = input("Write the name of your program: ")
9     base_command = ["/usr/bin/time", "-o", "output.txt", "-a", "/" + program]
10    times_execute = input("Write the times of your program need to be execute: ")
11    arguments = input("Insert here your aguments of the program: ")
12    original_program = input("Insert your original program por compare: ")
13    base_command_o = ["/usr/bin/time", "-o", "output_o.txt", "-a", "/" + original_program]
14    arguments = arguments.split()
15    if len(arguments) != 0 :
16        base_command.extend(arguments)
17        base_command_o.extend(arguments)
18
19    for i in range(0, int(times_execute)):
20        subprocess.call(base_command, shell=False)
21
22    with open("output.txt", 'r') as output_file:
23        file = output_file.read()
24        list_of_file = file.split()
25        output_file.close()
26
27
```

```

28     cpu_sum = 0
29     time_sum = 0
30     count = 0
31     cpu_min = 1000
32     time_min_user = 1000
33     time_min_system = 1000
34     cpu_max = 0
35     time_max_user = 0
36     time_max_system = 0
37
38
39
40     for i in list_of_file:
41
42         if i.find("user") != -1 :
43             val = float(i[0:4])
44             time_sum += val
45             if val < time_min_user :
46                 time_min_user = val
47
48             if val > time_max_user:
49                 time_max_user = val
50             count += 1
51
52
53         elif i.find("system") != -1 :
54             val = float(i[0:4])
55             time_sum += val
56             if val < time_min_system :
57                 time_min_system = val
58
59             if val > time_max_system:
60                 time_max_system = val
61
62         elif i.find("CPU") != -1 :
63             if i[0] != '?':
64                 if i[3] == '0':
65                     val = float(i[0:3])
66                 else :
67                     val = float(i[0:2])
68                 cpu_sum += val
69                 if val < cpu_min :
70                     cpu_min = val
71
72                 if val > cpu_max:
73                     cpu_max = val
74
75     cpu_sum_o = 0
76     time_sum_o = 0
77     count_o = 0
78     cpu_min_o = 1000
79     time_min_user_o = 1000
80     time_min_system_o = 1000
81     cpu_max_o = 0
82     time_max_user_o = 0
83     time_max_system_o = 0
84
85     for i in range(0 , int(times_execute)):
86         subprocess.call(base_command_o , shell=False)
87
88     with open("output_o.txt" , 'r') as output_file:
89         file = output_file.read()
90         list_of_file = file.split()
91
92
93     for i in list_of_file:
94
95         if i.find("user") != -1 :
96             val = float(i[0:4])

```

```

97     time_sum_o += val
98     if val < time_min_user_o :
99         time_min_user_o = val
100
101     if val > time_max_user_o:
102         time_max_user_o = val
103     count_o += 1
104
105
106     elif i.find("system") != -1 :
107         val = float(i[0:4])
108         time_sum_o += val
109         if val < time_min_system_o :
110             time_min_system_o = val
111
112         if val > time_max_system_o:
113             time_max_system_o = val
114
115     elif i.find("CPU") != -1 :
116         if i[0] != '?':
117             if i[3] == '0':
118                 val = float(i[0:3])
119             else :
120                 val = float(i[0:2])
121             cpu_sum_o += val
122             if val < cpu_min_o :
123                 cpu_min_o = val
124
125             if val > cpu_max_o:
126                 cpu_max_o = val
127
128
129 print("The minimum of user time was : " + str(time_min_user))
130 print("The max of user time was : " + str(time_max_user))
131 print("The average of time was : " + str(time_sum / count))
132 print("The minimum of system time was : " + str(time_min_system))
133 print("The max of user time was : " + str(time_max_system))
134 print("The minimum of cpu was : " + str(cpu_min))
135 print("The max of cpu was : " + str(cpu_max))
136 print("Average of cpu was : " + str(cpu_sum / count))
137 print("-----")
138 print("Statistics for original program")
139 print("The minimum of user time was : " + str(time_min_user_o))
140 print("The max of user time was : " + str(time_max_user_o))
141 print("The average of time was : " + str(time_sum_o / count_o))
142 print("The minimum of system time was : " + str(time_min_system_o))
143 print("The max of user time was : " + str(time_max_system_o))
144 print("The minimum of cpu was : " + str(cpu_min_o))
145 print("The max of cpu was : " + str(cpu_max_o))
146 print("Average of cpu was : " + str(cpu_sum_o / count_o))
147 print("-----")
148 print(str(time_sum_o) + " " + str(time_sum) + " " + str(cpu_sum_o) + " " + str(cpu_sum))
149 print("Speed up :")
150 print("Speed up for time : " + str(time_sum_o/time_sum))
151 print("Speed up for cpu : " + str(cpu_sum_o / cpu_sum))

```