lscpu:	CPU(s): 4 Nombre del modelo: Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz Virtualización: VT-x Caché L3: 6144K

POPCOUNT:

```
for i in 0 g 1 2; do
  printf "__OPTIM%1c__%48s\n" $i "" | tr " " "="
  rm  popcount
  gcc popcount.c -o popcount -O$i -D TEST=0
 for j in $(seq 0 10); do
  echo $j; ./popcount
 done | pr -11 -l 22 -w 80
done
```

ignorar medición 0, repetir columna si alguna medición se sale demasiado de la media

Prácticas de Estructura de Computadores
por Javier Fernández y Mancia Anguita

Zona para reprod
recordar que se ig

Optimización -O0	0	1	2	3	4	5	6	7	8	9	10	media
popcount1 (lenguaje C - for):	88383	84862	85236	87818	85713	86265	85389	85008	84887	87404	87601	86018
popcount2 (lenguaje C - while):	44753	50380	44931	49696	44392	44287	44384	44278	45396	45815	44281	45784
popcount3 (leng.ASM-body while 4i):	14332	14221	14148	14197	14068	14088	14045	16249	14122	14089	14078	14331
popcount4 (leng.ASM-body while 3i):	16136	14097	14052	14039	14067	14105	14090	14440	14156	14068	14056	14117
popcount5 (CS:APP2e 3.49-group 8b):	22166	22142	22046	22055	22116	22133	23467	22127	22073	22071	22038	22227
popcount6 (Wikipedia- naive - 32b):	10087	10496	10046	10075	10014	10078	10116	10060	10095	10283	10024	10129
popcount7 (Wikipedia- naive -128b):	5246	5250	5279	5273	5253	6360	5312	5267	5314	5249	5254	5381
popcount8 (asm SSE3 - pshufb 128b):	897	894	892	921	896	1401	897	898	889	891	892	947
popcount9 (asm SSE4- popcount 32b):	3148	3131	3201	3121	3167	4024	3134	3099	3106	3150	3107	3224
popcount10(asm SSE4- popcount128b):	956	973	961	959	957	1035	960	982	954	959	954	969

media	0	1
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		

Optimización -Og	0	1	2	3	4	5	6	7	8	9	10	media
popcount1 (lenguaje C - for):	24500	28126	26378	24496	24439	24481	25156	25035	25145	24409	24425	25209
popcount2 (lenguaje C - while):	9452	10063	9425	9426	9459	9434	9428	9798	9764	9406	9432	9564
popcount3 (leng.ASM-body while 4i):	11725	12022	11689	11966	11656	11768	11674	12033	12302	11671	11681	11846
popcount4 (leng.ASM-body while 3i):	11409	10850	10833	11122	10897	10810	10805	11170	10935	12265	10809	11050
popcount5 (CS:APP2e 3.49-group 8b):	6178	6204	6188	6179	6162	6155	6183	6350	6189	6422	6160	6219
popcount6 (Wikipedia- naive - 32b):	3255	3298	3289	3341	3280	3274	3277	3392	3267	3287	3259	3296
popcount7 (Wikipedia- naive -128b):	1764	1771	1788	1832	1867	1764	1772	1865	1778	1770	1765	1797
popcount8 (asm SSE3 - pshufb 128b):	455	455	454	464	456	478	489	471	472	455	465	466
popcount9 (asm SSE4- popcount 32b):	525	520	528	538	533	533	526	540	531	571	525	535
popcount10(asm SSE4- popcount128b):	370	402	358	369	361	365	361	369	363	364	373	369

media	0	1
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		

Optimización -O1	0	1	2	3	4	5	6	7	8	9	10	media
popcount1 (lenguaje C - for):	23085	23039	23069	22980	23063	23113	25328	22866	22896	22963	22975	23229
popcount2 (lenguaje C - while):	12654	12415	11779	11741	12317	12314	12227	12486	13988	11479	11130	12188
popcount3 (leng.ASM-body while 4i):	11786	12544	11828	11810	11857	11791	11816	11823	12398	11832	11783	11948
popcount4 (leng.ASM-body while 3i):	13779	13824	13720	17466	13821	14739	13730	13715	13851	13700	13712	14228
popcount5 (CS:APP2e 3.49-group 8b):	6406	6380	6369	7627	6367	6334	6394	6364	6377	6336	6751	6530
popcount6 (Wikipedia- naive - 32b):	3235	3282	3243	3265	3263	3281	3275	3263	3275	3257	3415	3282
popcount7 (Wikipedia- naive -128b):	1717	1695	1712	1711	1689	1700	1714	1689	1704	1693	1691	1700
popcount8 (asm SSE3 - pshufb 128b):	478	457	454	461	451	488	460	453	457	455	456	459
popcount9 (asm SSE4- popcount 32b):	526	522	519	525	520	529	528	521	527	535	522	525
popcount10(asm SSE4- popcount128b):	362	363	360	363	358	365	362	357	364	363	360	362

media	0	1
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		

Optimización -O2	0	1	2	3	4	5	6	7	8	9	10	media
popcount1 (lenguaje C - for):	14757	14671	14714	14667	14668	16538	14719	14630	14669	14676	14666	14862
popcount2 (lenguaje C - while):	8931	10844	8918	8917	11151	8907	8951	10113	8906	8914	9208	9483
popcount3 (leng.ASM-body while 4i):	11503	11541	11459	11408	11489	11404	11443	11439	11414	11425	11936	11496
popcount4 (leng.ASM-body while 3i):	10812	10922	10760	10797	10782	10756	10818	10835	10824	10771	10908	10817
popcount5 (CS:APP2e 3.49-group 8b):	4928	5270	4922	4926	4940	4931	4966	4923	4885	4945	4924	4963
popcount6 (Wikipedia- naive - 32b):	2699	2722	2678	2729	2676	2687	2678	2680	2701	2711	2685	2695
popcount7 (Wikipedia- naive -128b):	1589	1538	1559	1561	1573	1539	1583	1561	1560	1535	1536	1555
popcount8 (asm SSE3 - pshufb 128b):	454	458	457	475	458	456	455	458	457	456	459	459
popcount9 (asm SSE4- popcount 32b):	523	528	526	527	528	526	532	529	572	527	529	532
popcount10(asm SSE4- popcount128b):	358	380	358	361	361	359	368	361	362	358	361	363

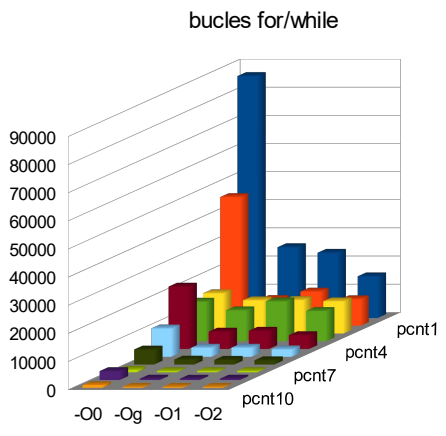
media	0	1
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		
#DIV/0!		

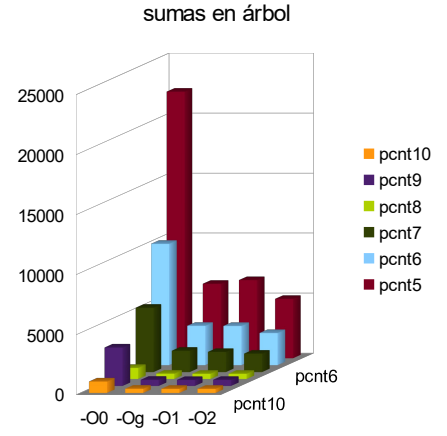
POPCOUNT:	-O0	-Og	-O1	-O2
pcnt1	86018	25209	**23229**	14862
pcnt2	45784	**9564**	12188	9483
pcnt3	14331	11846	**11948**	11496
pcnt4	14117	11050	**14228**	10817
pcnt5	22227	6219	6530	**4963**
pcnt6	10129	3296	3282	**2695**
pcnt7	5381	1797	1700	**1555**
pcnt8	947	466	459	**459**
pcnt9	3224	535	525	**532**
pcnt10	969	**369**	**362**	**363**

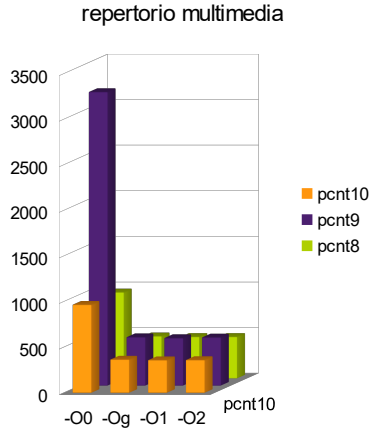
Ganancias:	-O0	-Og	-O1	-O2
pcnt1			1,00	
pcnt2		**2,43**		
pcnt3			1,94	
pcnt4			1,63	
pcnt5				**4,68**
pcnt6				**8,62**
pcnt7				**14,94**
pcnt8				**50,62**
pcnt9				43,63
pcnt10		**63,04**	64,26	64,01

Comentario

comparado con el for más rápido
el while es un 70% más rápido
ASM se queda en un 35%
o en un 43%
sumar en grupos 8b sale 3x más rápido
sumar en árbol 6x
lectura 128b sube a 10x
SSSE3 sube a 35x más rápido
SSE4 sólo 30x por leer 32b
SSE4 128b sube a 44x







ucir mediciones
nora medición 0