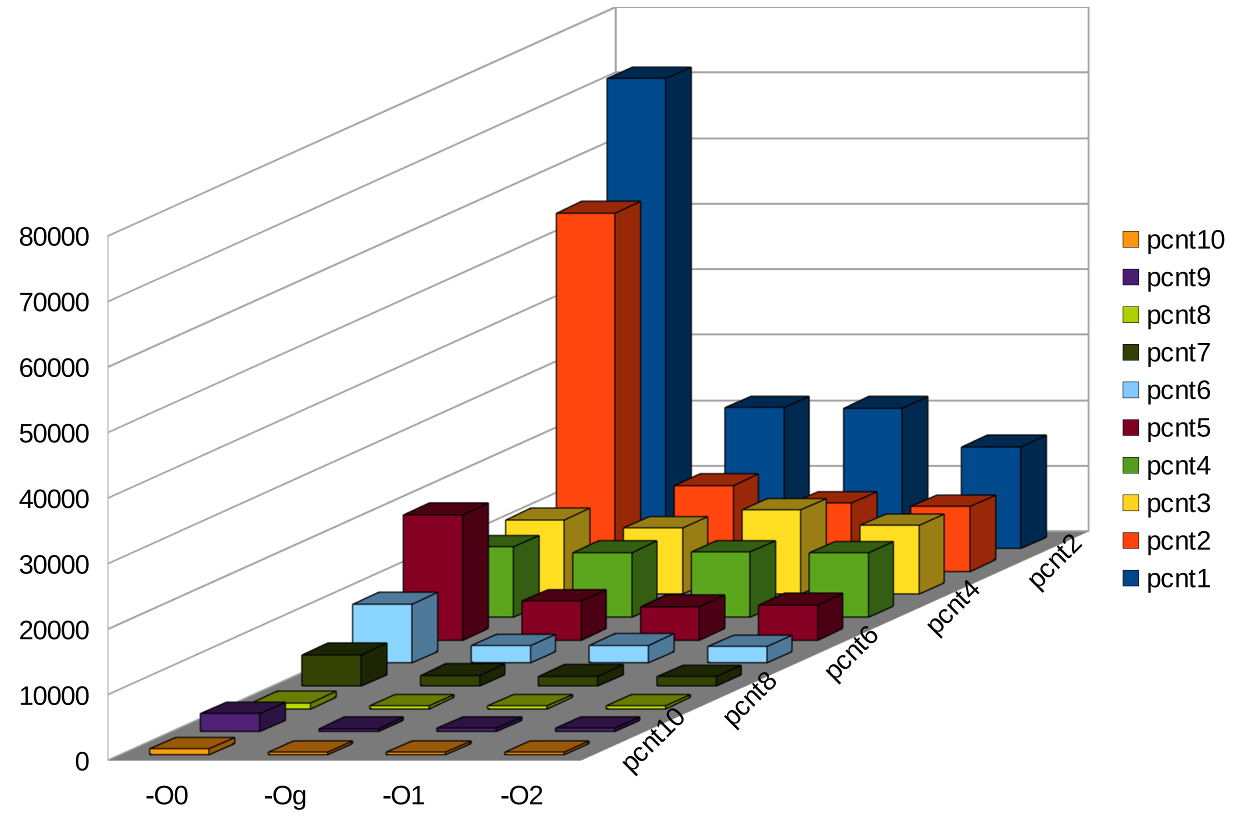
Práctica 3 Alumno: Francisco Javier Bolívar Expósito

Gráfica:



Código:

```
#include <stdio.h>                // para printf()
#include <stdlib.h>               // para exit()
#include <sys/time.h>             // para gettimeofday(), struct timeval

int resultado=0;
#ifndef TEST
#define TEST 5
#endif

#if TEST==1
    #define SIZE 4
    unsigned lista[SIZE]={0x80000000, 0x00400000, 0x00000200, 0x00000001};
    #define RESULT 4

#elif TEST==2
    #define SIZE 8
    unsigned lista[SIZE]={0x7fffffff, 0xffbfffff, 0xfffffdff, 0xfffffffe,
                          0x01000023, 0x00456700, 0x8900ab00, 0x00cd00ef};
    #define RESULT 156
```

```
#elif TEST==3
   #define SIZE 8
   unsigned lista[SIZE]={0x0       , 0x01020408, 0x35906a0c, 0x70b0d0e0,
                    0xffffffff, 0x12345678, 0x9abcdef0, 0xdeadbeef};
   #define RESULT 116

#elif TEST==4 || TEST==0
   #define NBITS 20
   #define SIZE (1<<NBITS)
   unsigned lista[SIZE];
   #define RESULT (SIZE * (NBITS/2))

#else
   #error "Definir TEST entre 0..4"
#endif

int popcount1(unsigned* array, size_t len)
{
   size_t i, j;
   int result = 0;
   unsigned x;

   for (i = 0; i < len; i++)
   {
      x = array[i];
      for (j = 0; j < 8*sizeof(unsigned); j++)
      {
         result += x & 0x1;
         x >>= 1;
      }
   }
   return result;
}

int popcount2(unsigned* array, size_t len)
{
   size_t i;
   unsigned x;
   int result = 0;

   for (i=0; i<len; i++)
   {
      x = array[i];
      while (x)
      {
         result += x & 0x1;
         x >>=1;
      }
   }

   return result;
}
```

```
int popcount3(unsigned* array, size_t len)
{
    size_t i;
    unsigned x;
    int result = 0;

    for (i=0; i<len; i++)
    {
        x = array[i];
        asm("\n"
        "ini3:          \n\t"
        "shr %[x]       \n\t"
        "adc $0, %[r]   \n\t"
        "test %[x], %[x]\n\t"
        "jnz ini3       \n\t"
        : [r]"+r" (result)
        : [x] "r" (x)       );
    }

    return result;
}

int popcount4(unsigned* array, size_t len)
{
    size_t i;
    unsigned x;
    int result = 0;

    for (i=0; i<len; i++)
    {
        x = array[i];
        asm("\n"
        "clc            \n\t"
        "ini4:          \n\t"
        "adc $0, %[r]   \n\t"
        "shr %[x]       \n\t"
        "jnz ini4       \n\t"
        "adc $0, %[r]       "
        : [r]"+r" (result)
        : [x] "r" (x)       );
    }

    return result;
}

int popcount5(unsigned* array, size_t len)
{
    size_t i, j;
    int result = 0, val=0;
    unsigned x;

    for (i=0; i<len; i++)
    {
        x = array[i];
        for (j = 0; j < 8; j++)
        {
            val += x & 0x01010101;
            x >>= 1;
        }
        val += (val >> 16);
        val += (val >> 8);
        result += val & 0xFF;
        val = 0;
    }

    return result;
}

int popcount6(unsigned* array, size_t len)
{
    size_t i;
    const unsigned M1 = 0x55555555, M2 = 0x33333333, M4 = 0x0f0f0f0f,
                   M8 = 0x00ff00ff, M16 = 0x0000ffff;
    int result = 0;
    unsigned x;

    for (i=0; i<len; i++)
    {
        x = array[i];

        x = (x & M1)  + ((x >> 1)  & M1);
        x = (x & M2)  + ((x >> 2)  & M2);
        x = (x & M4)  + ((x >> 4)  & M4);
        x = (x & M8)  + ((x >> 8)  & M8);
        x = (x & M16) + ((x >> 16) & M16);

        result += x;
    }

    return result;
}

int popcount7(unsigned* array, size_t len)
{
    size_t i;
    const unsigned long M1 = 0x5555555555555555,
    M2 = 0x3333333333333333, M4 = 0x0f0f0f0f0f0f0f0f,
    M8 = 0x00ff00ff00ff00ff, M16 = 0x0000ffff0000ffff,
    M32 = 0x00000000ffffffff;

    unsigned long x1, x2;
    int result = 0;

    if (len & 0x3) printf("leyendo 128b pero len no múltiplo de 4\n");

    for (i=0; i<len; i+=4)
    {
        x1 = *(unsigned long*) &array[i];
        x2 = *(unsigned long*) &array[i+2];

        x1 = (x1 & M1)  + ((x1 >> 1)  & M1);
        x1 = (x1 & M2)  + ((x1 >> 2)  & M2);
        x1 = (x1 & M4)  + ((x1 >> 4)  & M4);
        x1 = (x1 & M8)  + ((x1 >> 8)  & M8);
        x1 = (x1 & M16) + ((x1 >> 16) & M16);
        x1 = (x1 & M32) + ((x1 >> 32) & M32);

        x2 = (x2 & M1)  + ((x2 >> 1)  & M1);
        x2 = (x2 & M2)  + ((x2 >> 2)  & M2);
        x2 = (x2 & M4)  + ((x2 >> 4)  & M4);
        x2 = (x2 & M8)  + ((x2 >> 8)  & M8);
        x2 = (x2 & M16) + ((x2 >> 16) & M16);
        x2 = (x2 & M32) + ((x2 >> 32) & M32);

        result += x1+x2;
    }
    return result;
}

int popcount8(unsigned* array, size_t len)
{
    size_t i;
    int val, result=0;
    int SSE_mask[] = {0x0f0f0f0f, 0x0f0f0f0f, 0x0f0f0f0f, 0x0f0f0f0f};
    int SSE_LUTb[] = {0x02010100, 0x03020201, 0x03020201, 0x04030302};
                //   3 2 1 0     7 6 5 4    1110 8 8    15141312


    if (len & 0x3) printf("leyendo 128b pero len no múltiplo de 4\n");
    for (i=0; i<len; i+=4)
    {
        asm("movdqu    %[x], %%xmm0  \n\t"
            "movdqa  %%xmm0, %%xmm1  \n\t"
            "movdqu    %[m], %%xmm6  \n\t"
            "psrlw      $4 , %%xmm1  \n\t"
            "pand    %%xmm6, %%xmm0  \n\t"
            "pand    %%xmm6, %%xmm1  \n\t"

            "movdqu    %[l], %%xmm2  \n\t"
            "movdqa  %%xmm2, %%xmm3  \n\t"
            "pshufb  %%xmm0, %%xmm2  \n\t"
            "pshufb  %%xmm1, %%xmm3  \n\t"

            "paddb   %%xmm2, %%xmm3  \n\t"
            "pxor    %%xmm0, %%xmm0  \n\t"
            "psadbw  %%xmm0, %%xmm3  \n\t"
            "movhlps %%xmm3, %%xmm0  \n\t"
            "paddd   %%xmm3, %%xmm0  \n\t"
            "movd    %%xmm0, %[val]      "
            : [val]"=r" (val)
            :  [x]  "m" (array[i]),
               [m]  "m" (SSE_mask[0]),
               [l]  "m" (SSE_LUTb[0])
           );
        result += val;
    }
    return result;
}

int popcount9(unsigned* array, size_t len)
{
    size_t i;
    unsigned x;
    int val, result=0;

    for (i=0; i<len; i++)
    {
        x = array[i];
        asm("popcnt %[x],%[val]"
        : [val] "=r" (val)
        :   [x]  "r" (x)
        );
        result += val;
    }
    return result;
}

int popcount10(unsigned* array, size_t len)
{
    size_t i;
    unsigned long x1, x2;
    long val, result=0;

    if (len & 0x3) printf("leyendo 128b pero len no múltiplo de 4\n");
    for (i=0; i<len; i+=4)
    {
        x1 = *(unsigned long*) &array[i];
        x2 = *(unsigned long*) &array[i+2];
        asm("popcnt %[x1], %[val] \n\t"
            "popcnt %[x2], %%rdi  \n\t"
            "add    %%rdi, %[val]     "
        : [val]"=&r" (val)
        :  [x1]  "r" (x1),
           [x2]  "r" (x2)
        : "rdi"
        );
        result += val;
    }
    return result;
}
```

```
void crono(int (*func)(), char* msg){
    struct timeval tv1,tv2;                                        // gettimeofday() secs-usecs
    long           tv_usecs;                           // y sus cuentas

    gettimeofday(&tv1,NULL);
    resultado = func(lista, SIZE);
    gettimeofday(&tv2,NULL);

    tv_usecs=(tv2.tv_sec -tv1.tv_sec )*1E6+
             (tv2.tv_usec-tv1.tv_usec);
#if TEST==0
    printf(     "%ld" "\n",     tv_usecs);
#else
    printf("resultado = %d\t", resultado);
    printf("%s:%9ld us\n", msg, tv_usecs);
#endif
}

int main()
{
#if TEST==0 || TEST==4
    size_t i;
    for (i=0; i<SIZE; i++)
        lista[i]=i;
#endif

    crono(popcount1 , "popcount1 (lenguaje C -       for)");
    crono(popcount2 , "popcount2 (lenguaje C -     while)");
    crono(popcount3 , "popcount3 (leng.ASM-body while 4i)");
    crono(popcount4 , "popcount4 (leng.ASM-body while 3i)");
    crono(popcount5 , "popcount5 (CS:APP2e 3.49-group 8b)");
    crono(popcount6 , "popcount6 (Wikipedia- naive - 32b)");
    crono(popcount7 , "popcount7 (Wikipedia- naive -128b)");
    crono(popcount8 , "popcount8 (asm SSE3 - pshufb 128b)");
    crono(popcount9 , "popcount9 (asm SSE4- popcount 32b)");
    crono(popcount10, "popcount8 (asm SSE4- popcount128b)");

#if TEST!=0
    printf("calculado = %d\n", RESULT);
#endif

    exit(0);
}
```