Xeración e optimización.

Compiladores e interpretes - Práctica 1B

Barreiro Domínguez, Víctor Xesús

Xaneiro 2022

Índice

1. Introdución.

Realizaremos unha análise do impacto da técnica primeiro no código en ensamblador e posteriormente veremos as súas repercusión en canto a tempos de execución.

2. Ensamblador.

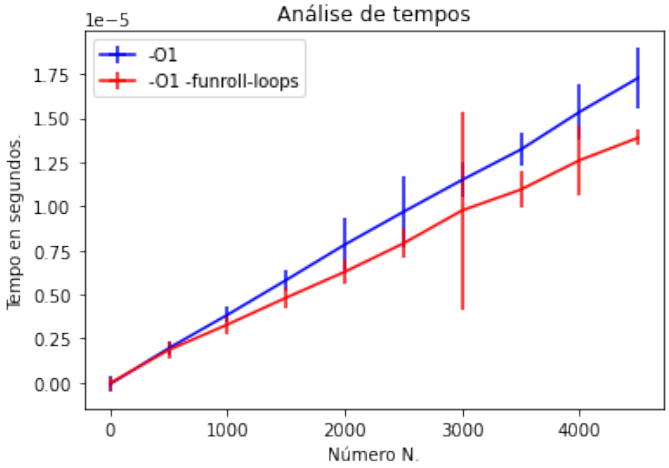
Na versión compilada co -O1, xa atopamos reordenadenacións do código respecto da secuencia escrita en C. Centarémonos nas diferencias entre as dúas versións -O1 e -O1 -funrooll-loops.

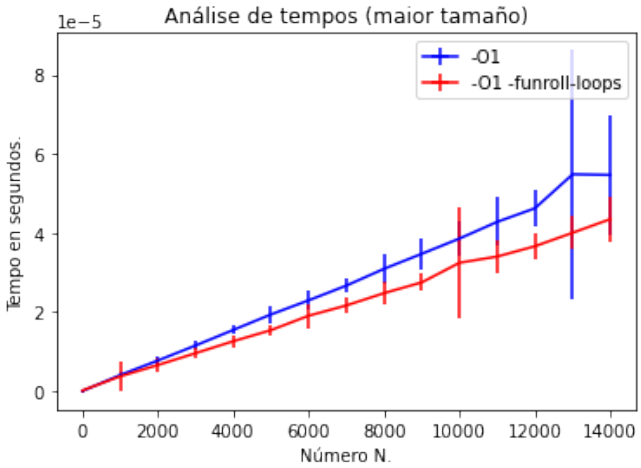
Salientar que cometemos un pequeno erro que pode ter impacto nos resultados, xa que como se pode ver no código do anexo, N xa non é unha constante que o compilador reemplace senon que é unha variable que é introducida dende a terminal. Isto fixémolo para permitir facer as probas de tempos axeitadamente e realizar suficientes execucións como para tomar datos que nos permitisen ver a influencia da técnica para distintos valores de N e poder ir reaxustando o intervalo a avaliar.

Con todo, pódese ver no código ensamblador como as instrucións asociadas a cada bucle aumentan moi considerablemente ao introducir *-funroll-loops*.

3. Resultados de tempos.

Para medir os tempos realizamos 100 execucións para cada talla e versión e cos datos obtidos calculamos a media e desviación típica que representamos sobre cada dato.





Atopacmos unha mellora nos tempos de execución na versións que emprega *-funroll-loops* e que se acrecenta a medida que aumentamos o tamaño dos bucles.

Debemos ter en conta que o tamaño dos bucles N é introducido en tempo de execución e non é posible substituír este valor no preprocesado. Por tanto, a mellora desta técnica podería variar fortemente se se coñecese este valor en tempo de compilación.

4. Código.

4.1. Código en C

```c
#include <sys/time.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char const *argv[]) {
  int i;
  double x;
  int N;

    struct timeval inicio, final;
    double tempo, tempo0;

  if (argc == 2)
    {
        N = atoi(argv[1]);
    }
    else
    {
        N = 1000;
    }

    double res[N];

    gettimeofday(&inicio,NULL);
    gettimeofday(&final,NULL);
    tempo0 = (final.tv_sec-inicio.tv_sec+(final.tv_usec-inicio.
    tv_usec)/1.e6);

    gettimeofday(&inicio,NULL);
  for(i = 0; i < N; i++)
    res[i] = 0.0004*i;

  for(i = 0; i < N; i++)
  {
    x = res[i];
    if (x < 10.0e6)
      x = x*x+0.0004;
    else
      x = x-900;

    res[i]+=x;
  }
  gettimeofday(&final,NULL);

  tempo = (final.tv_sec-inicio.tv_sec+(final.tv_usec-inicio.tv_usec
    )/1.e6) - tempo0;

    printf("%lf ", tempo);

  printf("resultado= %e\n", res[N-1]);

  return 0;
```

```
}
```

4.2. Ensamblador -O1

```
  .file "12.c"
  .text
  .section  .rodata.str1.1,"aMS",@progbits,1
.LC4:
  .string "%d "
.LC5:
  .string "resultado= %e\n"
  .text
  .globl  main
  .type main, @function
main:
.LFB39:
  .cfi_startproc
  endbr64
  pushq %rbp
  .cfi_def_cfa_offset 16
  .cfi_offset 6, -16
  movq  %rsp, %rbp
  .cfi_def_cfa_register 6
  pushq %r14
  pushq %r13
  pushq %r12
  pushq %rbx
  subq  $64, %rsp
  .cfi_offset 14, -24
  .cfi_offset 13, -32
  .cfi_offset 12, -40
  .cfi_offset 3, -48
  movq  %fs:40, %rax
  movq  %rax, -40(%rbp)
  xorl  %eax, %eax
  movl  $1000, %ebx
  cmpl  $2, %edi
  je  .L19
.L2:
  movslq  %ebx, %rax
  leaq  15(,%rax,8), %rax
  movq  %rax, %rdx
  andq  $-16, %rdx
  andq  $-4096, %rax
  movq  %rsp, %rsi
  subq  %rax, %rsi
  movq  %rsi, %rax
.L3:
  cmpq  %rax, %rsp
  je  .L4
  subq  $4096, %rsp
  orq $0, 4088(%rsp)
  jmp .L3
.L19:
```

```
  movq  8(%rsi), %rdi
  movl  $10, %edx
  movl  $0, %esi
  call  strtol@PLT
  movl  %eax, %ebx
  jmp .L2
.L4:
  movq  %rdx, %rax
  andl  $4095, %eax
  subq  %rax, %rsp
  testq %rax, %rax
  je  .L5
  orq $0, -8(%rsp,%rax)
.L5:
  movq  %rsp, %r12
  movq  %r12, %r13
  leaq  -80(%rbp), %r14
  movl  $0, %esi
  movq  %r14, %rdi
  call  gettimeofday@PLT
  leaq  -64(%rbp), %rdi
  movl  $0, %esi
  call  gettimeofday@PLT
  movq  -56(%rbp), %rax
  subq  -72(%rbp), %rax
  pxor  %xmm0, %xmm0
  cvtsi2sdq %rax, %xmm0
  divsd .LC0(%rip), %xmm0
  movq  -64(%rbp), %rax
  subq  -80(%rbp), %rax
  pxor  %xmm1, %xmm1
  cvtsi2sdq %rax, %xmm1
  addsd %xmm1, %xmm0
  movsd %xmm0, -88(%rbp)
  movl  $0, %esi
  movq  %r14, %rdi
  call  gettimeofday@PLT
  testl %ebx, %ebx
  jle .L6
  leal  -1(%rbx), %ecx
  movl  $0, %eax
  movsd .LC1(%rip), %xmm1
.L7:
  pxor  %xmm0, %xmm0
  cvtsi2sdl %eax, %xmm0
  mulsd %xmm1, %xmm0
  movsd %xmm0, 0(%r13,%rax,8)
  movq  %rax, %rdx
  addq  $1, %rax
  cmpq  %rcx, %rdx
  jne .L7
  movq  %r12, %rax
  leaq  8(%r12,%rcx,8), %rcx
  movsd .LC2(%rip), %xmm3
  movsd .LC3(%rip), %xmm5
  movsd .LC1(%rip), %xmm4
  jmp .L11
```

```
.L17:
  movapd   %xmm2, %xmm0
  subsd  %xmm5, %xmm0
.L10:
  addsd  %xmm2, %xmm0
  movsd  %xmm0, (%rdx)
  addq   $8, %rax
  cmpq   %rcx, %rax
  je   .L6
.L11:
  movq   %rax, %rdx
  movsd (%rax), %xmm2
  comisd   %xmm2, %xmm3
  jbe .L17
  movapd   %xmm2, %xmm1
  mulsd  %xmm2, %xmm1
  addsd  %xmm4, %xmm1
  movapd   %xmm1, %xmm0
  jmp .L10
.L6:
  leaq   -64(%rbp), %rdi
  movl   $0, %esi
  call   gettimeofday@PLT
  movq   -56(%rbp), %rax
  subq   -72(%rbp), %rax
  pxor   %xmm0, %xmm0
  cvtsi2sdq  %rax, %xmm0
  divsd  .LC0(%rip), %xmm0
  movq   -64(%rbp), %rax
  subq   -80(%rbp), %rax
  pxor   %xmm1, %xmm1
  cvtsi2sdq  %rax, %xmm1
  addsd  %xmm1, %xmm0
  subsd  -88(%rbp), %xmm0
  leaq   .LC4(%rip), %rsi
  movl   $1, %edi
  movl   $1, %eax
  call   __printf_chk@PLT
  subl   $1, %ebx
  movslq   %ebx, %rbx
  movsd (%r12,%rbx,8), %xmm0
  leaq   .LC5(%rip), %rsi
  movl   $1, %edi
  movl   $1, %eax
  call   __printf_chk@PLT
  movq   -40(%rbp), %rax
  xorq   %fs:40, %rax
  jne .L20
  movl   $0, %eax
  leaq   -32(%rbp), %rsp
  popq   %rbx
  popq   %r12
  popq   %r13
  popq   %r14
  popq   %rbp
  .cfi_remember_state
  .cfi_def_cfa 7, 8
```

```
  ret
.L20:
  .cfi_restore_state
  call   __stack_chk_fail@PLT
  .cfi_endproc
.LFE39:
  .size main, .-main
  .section   .rodata.cst8,"aM",@progbits,8
  .align 8
.LC0:
  .long 0
  .long 1093567616
  .align 8
.LC1:
  .long 3944497965
  .long 1060779746
  .align 8
.LC2:
  .long 0
  .long 1097011920
  .align 8
.LC3:
  .long 0
  .long 1082925056
  .ident  "GCC: (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0"
  .section   .note.GNU-stack,"",@progbits
  .section   .note.gnu.property,"a"
  .align 8
  .long   1f - 0f
  .long   4f - 1f
  .long   5
0:
  .string  "GNU"
1:
  .align 8
  .long   0xc0000002
  .long   3f - 2f
2:
  .long   0x3
3:
  .align 8
4:
```

4.3. Ensamblador -O1 -funroll-loops

```
  .file "12.c"
  .text
  .section   .rodata.str1.1,"aMS",@progbits,1
.LC4:
  .string "%d "
.LC5:
  .string "resultado= %e\n"
  .text
  .globl  main
```

```
  .type main, @function
main:
.LFB39:
  .cfi_startproc
  endbr64
  pushq %rbp
  .cfi_def_cfa_offset 16
  .cfi_offset 6, -16
  movq  %rsp, %rbp
  .cfi_def_cfa_register 6
  pushq %r14
  pushq %r13
  pushq %r12
  pushq %rbx
  subq  $64, %rsp
  .cfi_offset 14, -24
  .cfi_offset 13, -32
  .cfi_offset 12, -40
  .cfi_offset 3, -48
  movq  %fs:40, %rax
  movq  %rax, -40(%rbp)
  xorl  %eax, %eax
  movl  $1000, %ebx
  cmpl  $2, %edi
  je  .L73
.L2:
  movslq  %ebx, %rax
  leaq  15(,%rax,8), %rsi
  movq  %rsi, %rdx
  andq  $-16, %rdx
  andq  $-4096, %rsi
  movq  %rsp, %rax
  subq  %rsi, %rax
  movq  %rax, %rcx
.L3:
  cmpq  %rcx, %rsp
  je  .L4
  subq  $4096, %rsp
  orq $0, 4088(%rsp)
  jmp .L3
.L73:
  movq  8(%rsi), %rdi
  movl  $10, %edx
  movl  $0, %esi
  call  strtol@PLT
  movl  %eax, %ebx
  jmp .L2
.L4:
  andl  $4095, %edx
  subq  %rdx, %rsp
  testq %rdx, %rdx
  je  .L5
  orq $0, -8(%rsp,%rdx)
.L5:
  movq  %rsp, %r12
  movq  %r12, %r13
  leaq  -80(%rbp), %r14
```

```
66   movl   $0, %esi
67   movq   %r14, %rdi
68   call   gettimeofday@PLT
69   leaq   -64(%rbp), %rdi
70   movl   $0, %esi
71   call   gettimeofday@PLT
72   movq   -56(%rbp), %rdi
73   subq   -72(%rbp), %rdi
74   pxor   %xmm0, %xmm0
75   cvtsi2sdq %rdi, %xmm0
76   divsd  .LC0(%rip), %xmm0
77   movq   -64(%rbp), %r8
78   subq   -80(%rbp), %r8
79   pxor   %xmm1, %xmm1
80   cvtsi2sdq %r8, %xmm1
81   addsd  %xmm1, %xmm0
82   movsd  %xmm0, -88(%rbp)
83   movl   $0, %esi
84   movq   %r14, %rdi
85   call   gettimeofday@PLT
86   testl  %ebx, %ebx
87   jle  .L6
88   movl   $0, %r9d
89   movsd  .LC1(%rip), %xmm2
90   leal   -1(%rbx), %r10d
91   movq   %rbx, %r11
92   andl   $7, %r11d
93   je   .L7
94   cmpq   $1, %r11
95   je   .L53
96   cmpq   $2, %r11
97   je   .L54
98   cmpq   $3, %r11
99   je   .L55
100  cmpq   $4, %r11
101  je   .L56
102  cmpq   $5, %r11
103  je   .L57
104  cmpq   $6, %r11
105  je   .L58
106  pxor   %xmm3, %xmm3
107  cvtsi2sdl %r9d, %xmm3
108  mulsd  %xmm2, %xmm3
109  movsd  %xmm3, (%r12,%r9,8)
110  addq   $1, %r9
111 .L58:
112  pxor   %xmm4, %xmm4
113  cvtsi2sdl %r9d, %xmm4
114  mulsd  %xmm2, %xmm4
115  movsd  %xmm4, 0(%r13,%r9,8)
116  addq   $1, %r9
117 .L57:
118  pxor   %xmm5, %xmm5
119  cvtsi2sdl %r9d, %xmm5
120  mulsd  %xmm2, %xmm5
121  movsd  %xmm5, 0(%r13,%r9,8)
122  addq   $1, %r9
```

```
.L56:
  pxor   %xmm6, %xmm6
  cvtsi2sdl %r9d, %xmm6
  mulsd  %xmm2, %xmm6
  movsd  %xmm6, 0(%r13,%r9,8)
  addq   $1, %r9
.L55:
  pxor   %xmm7, %xmm7
  cvtsi2sdl %r9d, %xmm7
  mulsd  %xmm2, %xmm7
  movsd  %xmm7, 0(%r13,%r9,8)
  addq   $1, %r9
.L54:
  pxor   %xmm8, %xmm8
  cvtsi2sdl %r9d, %xmm8
  mulsd  %xmm2, %xmm8
  movsd  %xmm8, 0(%r13,%r9,8)
  addq   $1, %r9
.L53:
  pxor   %xmm9, %xmm9
  cvtsi2sdl %r9d, %xmm9
  mulsd  %xmm2, %xmm9
  movsd  %xmm9, 0(%r13,%r9,8)
  movq   %r9, %rsi
  addq   $1, %r9
  cmpq   %r10, %rsi
  je   .L71
.L7:
  pxor   %xmm10, %xmm10
  cvtsi2sdl %r9d, %xmm10
  mulsd  %xmm2, %xmm10
  movsd  %xmm10, 0(%r13,%r9,8)
  addq   $1, %r9
  pxor   %xmm11, %xmm11
  cvtsi2sdl %r9d, %xmm11
  mulsd  %xmm2, %xmm11
  movsd  %xmm11, 0(%r13,%r9,8)
  leaq   1(%r9), %rcx
  pxor   %xmm12, %xmm12
  cvtsi2sdl %ecx, %xmm12
  mulsd  %xmm2, %xmm12
  movsd  %xmm12, 0(%r13,%rcx,8)
  leaq   2(%r9), %rdx
  pxor   %xmm13, %xmm13
  cvtsi2sdl %edx, %xmm13
  mulsd  %xmm2, %xmm13
  movsd  %xmm13, 0(%r13,%rdx,8)
  leaq   3(%r9), %rax
  pxor   %xmm14, %xmm14
  cvtsi2sdl %eax, %xmm14
  mulsd  %xmm2, %xmm14
  movsd  %xmm14, 0(%r13,%rax,8)
  leaq   4(%r9), %r14
  pxor   %xmm15, %xmm15
  cvtsi2sdl %r14d, %xmm15
  mulsd  %xmm2, %xmm15
  movsd  %xmm15, 0(%r13,%r14,8)
```

```
180   leaq   5(%r9), %rdi
181   pxor   %xmm0, %xmm0
182   cvtsi2sdl %edi, %xmm0
183   mulsd  %xmm2, %xmm0
184   movsd  %xmm0, 0(%r13,%rdi,8)
185   leaq   6(%r9), %r8
186   pxor   %xmm1, %xmm1
187   cvtsi2sdl %r8d, %xmm1
188   mulsd  %xmm2, %xmm1
189   movsd  %xmm1, 0(%r13,%r8,8)
190   addq   $7, %r9
191   cmpq   %r10, %r8
192   jne .L7
193 .L71:
194   movq   %r12, %rax
195   leaq   8(%r12,%r10,8), %r13
196   movsd  .LC2(%rip), %xmm6
197   movsd  .LC3(%rip), %xmm4
198   movsd  .LC1(%rip), %xmm5
199   movq   %r13, %r9
200   subq   %r12, %r9
201   subq   $8, %r9
202   shrq   $3, %r9
203   addq   $1, %r9
204   andl   $3, %r9d
205   je   .L11
206   cmpq   $1, %r9
207   je   .L59
208   cmpq   $2, %r9
209   je   .L60
210   movq   %r12, %r10
211   movsd (%r12), %xmm3
212   comisd   %xmm3, %xmm6
213   ja   .L17
214   movapd   %xmm3, %xmm2
215   subsd  %xmm4, %xmm2
216 .L65:
217   addsd  %xmm3, %xmm2
218   movsd  %xmm2, (%r10)
219   addq   $8, %rax
220 .L60:
221   movq   %rax, %r11
222   movsd (%rax), %xmm7
223   comisd   %xmm7, %xmm6
224   ja   .L20
225   movapd   %xmm7, %xmm8
226   subsd  %xmm4, %xmm8
227 .L66:
228   addsd  %xmm7, %xmm8
229   movsd  %xmm8, (%r11)
230   addq   $8, %rax
231 .L59:
232   movq   %rax, %rsi
233   movsd (%rax), %xmm9
234   comisd   %xmm9, %xmm6
235   ja   .L23
236   movapd   %xmm9, %xmm10
```

```
237   subsd  %xmm4, %xmm10
238 .L67:
239   addsd  %xmm9, %xmm10
240   movsd  %xmm10, (%rsi)
241   addq   $8, %rax
242   cmpq   %r13, %rax
243   jne .L11
244 .L6:
245   leaq   -64(%rbp), %rdi
246   movl   $0, %esi
247   call   gettimeofday@PLT
248   movq   -56(%rbp), %r14
249   subq   -72(%rbp), %r14
250   pxor   %xmm0, %xmm0
251   cvtsi2sdq %r14, %xmm0
252   divsd  .LC0(%rip), %xmm0
253   movq   -64(%rbp), %rdi
254   subq   -80(%rbp), %rdi
255   pxor   %xmm6, %xmm6
256   cvtsi2sdq %rdi, %xmm6
257   addsd  %xmm6, %xmm0
258   subsd  -88(%rbp), %xmm0
259   leaq   .LC4(%rip), %rsi
260   movl   $1, %edi
261   movl   $1, %eax
262   call   __printf_chk@PLT
263   subl   $1, %ebx
264   movslq   %ebx, %rbx
265   movsd (%r12,%rbx,8), %xmm0
266   leaq   .LC5(%rip), %rsi
267   movl   $1, %edi
268   movl   $1, %eax
269   call   __printf_chk@PLT
270   movq   -40(%rbp), %rax
271   xorq   %fs:40, %rax
272   jne .L74
273   movl   $0, %eax
274   leaq   -32(%rbp), %rsp
275   popq   %rbx
276   popq   %r12
277   popq   %r13
278   popq   %r14
279   popq   %rbp
280   .cfi_remember_state
281   .cfi_def_cfa 7, 8
282   ret
283 .L63:
284   .cfi_restore_state
285   movapd   %xmm11, %xmm12
286   subsd  %xmm4, %xmm12
287   jmp .L10
288 .L74:
289   call   __stack_chk_fail@PLT
290 .L17:
291   movapd   %xmm3, %xmm1
292   mulsd  %xmm3, %xmm1
293   addsd  %xmm5, %xmm1
```

```
294   movapd   %xmm1, %xmm2
295   jmp .L65
296 .L20:
297   movapd   %xmm7, %xmm1
298   mulsd %xmm7, %xmm1
299   addsd %xmm5, %xmm1
300   movapd   %xmm1, %xmm8
301   jmp .L66
302 .L23:
303   movapd   %xmm9, %xmm10
304   mulsd %xmm9, %xmm10
305   addsd %xmm5, %xmm10
306   jmp .L67
307 .L26:
308   movapd   %xmm13, %xmm1
309   mulsd %xmm13, %xmm1
310   addsd %xmm5, %xmm1
311   movapd   %xmm1, %xmm14
312   jmp .L68
313 .L28:
314   movapd   %xmm15, %xmm1
315   mulsd %xmm15, %xmm1
316   addsd %xmm5, %xmm1
317   movapd   %xmm1, %xmm0
318   jmp .L69
319 .L30:
320   movapd   %xmm3, %xmm1
321   mulsd %xmm3, %xmm1
322   addsd %xmm5, %xmm1
323   movapd   %xmm1, %xmm2
324 .L70:
325   addsd %xmm3, %xmm2
326   movsd %xmm2, 16(%rdx)
327   leaq 24(%rdx), %rax
328   cmpq %r13, %rax
329   je .L6
330 .L11:
331   movq %rax, %rcx
332   movsd (%rax), %xmm11
333   comisd   %xmm11, %xmm6
334   jbe .L63
335   movapd   %xmm11, %xmm1
336   mulsd %xmm11, %xmm1
337   addsd %xmm5, %xmm1
338   movapd   %xmm1, %xmm12
339 .L10:
340   addsd %xmm11, %xmm12
341   movsd %xmm12, (%rcx)
342   leaq 8(%rax), %rdx
343   movsd 8(%rax), %xmm13
344   comisd   %xmm13, %xmm6
345   ja .L26
346   movapd   %xmm13, %xmm14
347   subsd %xmm4, %xmm14
348 .L68:
349   addsd %xmm13, %xmm14
350   movsd %xmm14, (%rdx)
```

```
  movsd 8(%rdx), %xmm15
  comisd %xmm15, %xmm6
  ja .L28
  movapd %xmm15, %xmm0
  subsd %xmm4, %xmm0
.L69:
  addsd %xmm15, %xmm0
  movsd %xmm0, 8(%rdx)
  movsd 16(%rdx), %xmm3
  comisd %xmm3, %xmm6
  ja .L30
  movapd %xmm3, %xmm2
  subsd %xmm4, %xmm2
  jmp .L70
  .cfi_endproc
.LFE39:
  .size main, .-main
  .section .rodata.cst8,"aM",@progbits,8
  .align 8
.LC0:
  .long 0
  .long 1093567616
  .align 8
.LC1:
  .long 3944497965
  .long 1060779746
  .align 8
.LC2:
  .long 0
  .long 1097011920
  .align 8
.LC3:
  .long 0
  .long 1082925056
  .ident "GCC: (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0"
  .section .note.GNU-stack,"",@progbits
  .section .note.gnu.property,"a"
  .align 8
  .long 1f - 0f
  .long 4f - 1f
  .long 5
0:
  .string "GNU"
1:
  .align 8
  .long 0xc0000002
  .long 3f - 2f
2:
  .long 0x3
3:
  .align 8
4:
```