Nana Performance Summary

P.J.Maker

December 28, 2014

This document contains some measurements for the space and time costs for the nana library. Data provided includes:

- Time cost in ns.
- Space in bytes.
- Generated assembler code.
- Results for various compiler options such as optimisation.

These test results were generated using:

- Operating System, release and cpu: Linux 3.13.0-43-generic x86_64
- CPU speed is analysed by the supplied bogomips program which gives a value of 945.2..995.9 BogMips . BogoMips (bogus millions of instructions per seconds) are a standard unit of measure invented by Linus Torvalds for use in Linux. They represent something vaguely related to the number of instructions executed per second.
- Note: nana should be installed using I_DEFAULT=fast ./configure for these measurements.

The following table contains a summary of the results:

Code	Size	Time	Options
-0			
-0			
-0			
-0			
-0			
-0			
-0			
-0			
-0			

Note:

- assert() is your systems standard assert macro.
- TRAD_assert() is the traditional implementation of assert which calls fprintf() and exit() directly.
- I() is the nana equivelant of assert.

- DI() is implemented using the debugger. It is very space efficient but takes longer than inline C code (such as I()).
- I(A(...)) is checking that all 10 values in the array a are positive.
- now() measures the current time and returns a —double— value.
- L() optionally prints a debugging message.
- DL() is the debugger equivelant of L().

Note that measurement code depends on GNU CC extensions and is not a thing of great beauty.

1 How was is it measured?

See Makefile.am and measure.sh for the true story, a quick summary would be:

- The code fragments are stored one per line in a file such as summary.tst.
- The measure.sh program takes as arguments a set of compiler flags such as -O which are used for each line in the input file.
- The code fragment is copied 256 times by a macro inside a loop which in turn executes 1024 times. For small examples it is expected that the entire loop will fit inside the cache.
- Time is measured using the nana now() function.

The variables and code fragments used defined in prelude.c and postlude.c. All variables are declared volatile to prevent the compile optimising access to variables.

In addition all programs are compiled with the following options:

- -g debugger information is always turned on since we need it for parts of the nana library. Note that gcc happily optimises code with -g enabled.
- -fno-defer-pop gcc by default only pops arguments off the stack after a number of calls. This option causes each call to immediately pop its arguments off the stack.

2 Detailed results

This section contains some more detailed results.

2.1 Assert

Code	Size	Time	Options
assert(i >= 10);	28	1ns	-O0
assert(i >= 10);	29	1ns	-O1
assert(i >= 10);	13	$0 \mathrm{ns}$	-O3
<pre>BSD_assert(i >= 10);</pre>	28	1ns	-O0
<pre>BSD_assert(i >= 10);</pre>	4	$0 \mathrm{ns}$	-O1
<pre>BSD_assert(i >= 10);</pre>	4	$0 \mathrm{ns}$	-O3
<pre>TRAD_assert(i >= 10);</pre>	59	1ns	-O0
<pre>TRAD_assert(i >= 10);</pre>	63	1ns	-O1
<pre>TRAD_assert(i >= 10);</pre>	13	0ns	-O3
I(i >= 10);	28	1ns	-O0
I(i >= 10);	29	1ns	-O1
I(i >= 10);	13	$0 \mathrm{ns}$	-O3
DI(i >= 10);	10	13.5 us	-O0
DI(i >= 10);	10	13.3us	-O1
DI(i >= 10);	10	13.4us	-O3

2.2 Quantifiers

Code	Size	Time	Opt
<pre>I(A(char *p = str, *p != '\0', p++, islower(*p)));</pre>	130	46ns	-O0
<pre>I(A(char *p = str, *p != '\0', p++, islower(*p)));</pre>	50	13ns	-O1
<pre>I(A(char *p = str, *p != '\0', p++, islower(*p)));</pre>	69	9ns	-O3
I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));	174	418ns	-O0
I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));	84	125 ns	-O1
I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));	220	$76 \mathrm{ns}$	-O3

2.3 Log

Code	Size	Time	Options
<pre>printf("helloworldn");</pre>	15	-O0	
<pre>printf("helloworldn");</pre>	20	-O1	
<pre>printf("helloworldn");</pre>	-4529	-O3	
L("helloworldn");	30	-O0	
L("helloworldn");	27	-O1	
L("helloworldn");	-7089	-O3	
<pre>DL("helloworldn");</pre>	10	2.9 us	-O0
<pre>DL("helloworldn");</pre>	10	2.9 us	-O1
<pre>DL("helloworldn");</pre>	10	2.9 us	-O3
gi = 0; LG(gi & 0x10, "helloworldn");	53	1ns	-O0
gi = 0; LG(gi & 0x10, "helloworldn");	47	1ns	-O1
gi = 0; LG(gi & 0x10, "helloworldn");	24	0ns	-O3
gi = ~0; LG(gi & 0x10, "helloworldn");	53	-O0	
gi = ~0; LG(gi & 0x10, "helloworldn");	47	-O1	
gi = ~0; LG(gi & 0x10, "helloworldn");	24	-O3	
<pre>LHP(fprintf,log,"helloworldn");</pre>	27	$26\mathrm{ns}$	-O0
<pre>LHP(fprintf,log,"helloworldn");</pre>	23	$27\mathrm{ns}$	-O1
<pre>LHP(fprintf,log,"helloworldn");</pre>	-6065	$26\mathrm{ns}$	-O3
<pre>LHP(L_buffer_printf,buf,"helloworldn");</pre>	22	$78\mathrm{ns}$	-O0
<pre>LHP(L_buffer_printf,buf,"helloworldn");</pre>	18	$74\mathrm{ns}$	-O1
<pre>LHP(L_buffer_printf,buf,"helloworldn");</pre>	-4014	$72\mathrm{ns}$	-O3
<pre>LHP(syslog,LOG_USER, "helloworldn");</pre>	20	3.5 us	-O0
<pre>LHP(syslog,LOG_USER, "helloworldn");</pre>	25	3.9 us	-O1
<pre>LHP(syslog,LOG_USER, "helloworldn");</pre>	-5809	3.6 us	-O3

2.4 Nop

Code	Size	Time	Options
asm("");	0	0ns	-O0
asm("");	0	0ns	-O1
asm("");	0	0ns	-O3
asm("nop");	1	0ns	-O0
asm("nop");	1	0ns	-O1
asm("nop");	1	0ns	-O3
asm("nop;nop;");	2	0ns	-O0
asm("nop;nop;");	2	0ns	-O1
asm("nop;nop;");	2	0ns	-O3
asm("nop;nop;nop;");	3	0ns	-O0
<pre>asm("nop;nop;nop;");</pre>	3	0ns	-O1
<pre>asm("nop;nop;nop;");</pre>	3	0ns	-O3
<pre>asm("nop;nop;nop;");</pre>	4	0ns	-O0
<pre>asm("nop;nop;nop;nop;");</pre>	4	0ns	-O1
<pre>asm("nop;nop;nop;");</pre>	4	0ns	-O3
<pre>asm("nop;nop;nop;nop;");</pre>	5	1ns	-O0
<pre>asm("nop;nop;nop;nop;");</pre>	5	0ns	-O1
<pre>asm("nop;nop;nop;nop;");</pre>	5	0ns	-O3

2.5 C Operations

Code	Size	Time	Options
i = 4;	7	0ns	-O0
i = 4;	8	0ns	-O1
i = 4;	8	0ns	-O3
gi = 11;	10	0ns	-O0
gi = 11;	10	0ns	-O1
gi = 11;	10	0ns	-O3
f = 12.0;	9	0ns	-O0
f = 12.0;	14	0ns	-O1
f = 12.0;	8	0ns	-O3
gf = 12.0;	12	0ns	-O0
gf = 12.0;	16	0ns	-O1
gf = 12.0;	10	0ns	-O3
i++;	9	2ns	-O0
i++;	11	2ns	-O1
i++;	11	2ns	-O3
gi++;	15	2ns	-O0
gi++;	15	2ns	-O1
gi++;	15	2ns	-O3
j = a[i];	15	0ns	-O0
j = a[i];	17	1ns	-O1
j = a[i];	16	0ns	-O3

2.6 Data cache testing

These are just some tests using a large array which should hopefully exceed the size of the D-cache on your machine.

Code	Size	Time	Options
-DNT=16			

3 Code

This section contains a listing of all the generated code fragments.

 \bullet printf("helloworldn"); with gcc -g -00 produces:

```
.loc 1 119 0

#NO_APP

movl $.LC4, %edi
movl $0, %eax
call printf
.loc 1 120 0
```

```
• printf("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
         movl $.LC4, %esi
                 $1, %edi
         movl
               $0, %eax
         movl
         call __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• printf("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 104 0
         movl $.LC4, %esi
                 $1, %edi
         movl
                 %eax, %eax
         xorl
          call
                  __printf_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 120 0
• L("helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  stdout(%rip), %rax
         movq
         movq
                  %rax, %rcx
                  $11, %edx
         movl
         movl
                  $1, %esi
                  $.LC4, %edi
         movl
          call
                  fwrite
          .loc 1 120 0
• L("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
```

```
.LBB527:
         .loc 2 97 0
         \verb"movq" stdout(%rip), %rcx"
         movl $11, %edx
         movl $1, %esi
         movl $.LC4, %edi
         call fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• L("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 2 97 0
         movq stdout(%rip), %rcx
                $11, %edx
         movl
               $1, %esi
         movl
               $.LC4, %edi
         movl
         call fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
         .loc 1 120 0
• DL("helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
         movl
                $0, _dl_target(%rip)
         .loc 1 120 0
• DL("helloworldn"); with gcc -g -O1 produces:
          .loc 1 119 0
 #NO_APP
         movl
               $0, _dl_target(%rip)
         .loc 1 120 0
• DL("helloworldn"); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
                 $0, _dl_target(%rip)
         movl
          .loc 1 120 0
```

```
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $0, gi(%rip)
                  gi(%rip), %eax
          movl
                  $16, %eax
          andl
                  %eax, %eax
          testl
                   .L5
          jе
          .loc 1 119 0 is_stmt 0 discriminator 1 \,
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 120 0 is_stmt 1
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
          movl
                  $16, %al
          testb
          jе
                   .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
                  fwrite
          call
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O3 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
                  $16, %al
```

testb

```
.L1045
          jne
  .L5:
          .loc 1 120 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
          andl
                  $16, %eax
                  %eax, %eax
          testl
          jе
                  .L5
          .loc 1 119 0 is_stmt 0 discriminator 1
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
          movl
                  $.LC4, %edi
          call
                  fwrite
  .L5:
          .loc 1 120 0 is_stmt 1
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $-1, gi(%rip)
                  gi(%rip), %eax
          movl
                  $16, %al
          testb
          jе
                  .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
          movl
                  $.LC4, %edi
                  fwrite
          call
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
```

```
#NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
          testb $16, %al
                  .L1045
          jne
  .L5:
          .loc 1 120 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  -32(\%rbp), \%rax
          movq
                  %rax, %rcx
          movq
          movl
                  $11, %edx
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
          .loc 1 120 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
          movq
                 %rbx, %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• LHP(fprintf,log, "helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 97 0
          movl
                  $11, %edx
                  %rbx, %rcx
          movq
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
```

```
.LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 120 0
• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
          movq
                 -40(%rbp), %rax
                  $.LC4, %esi
          movl
                 %rax, %rdi
          movq
                  $0, %eax
          movl
                  L_buffer_printf
          call
          .loc 1 120 0
• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $.LC4, %esi
          movq %rbx, %rdi
          movl
                  $0, %eax
                 L_buffer_printf
          call
  .LVL6:
          .loc 1 120 0
• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                 $.LC4, %esi
          movq %rbx, %rdi
          xorl
                 %eax, %eax
          call
                 L_buffer_printf
  .LVL279:
          .loc 1 120 0
• LHP(syslog,LOG_USER, "helloworldn"); with gcc -g -OO produces:
          .loc 1 119 0
 #NO_APP
                  $.LC4, %esi
          movl
                  $8, %edi
          movl
```

• LHP(syslog,LOG_USER, "helloworldn"); with gcc -g -O1 produces:

\$0, %eax

syslog

movl call

.loc 1 120 0

```
.LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
          .loc 3 31 0
          movl $.LC4, %edx
          movl
                 $1, %esi
                $8, %edi
          movl
                  $0, %eax
          movl
          call
                  __syslog_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O3 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 3 31 0
          movl $.LC4, %edx
          movl
                 $1, %esi
                  $8, %edi
          movl
                 %eax, %eax
          xorl
                  __syslog_chk
          call
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 120 0
• printf("helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  $.LC4, %edi
          movl
          movl
                  $0, %eax
          call
                  printf
          .loc 1 120 0
\bullet printf("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
```

```
$.LC4, %esi
         movl
                  $1, %edi
         movl
                $0, %eax
         movl
                 __printf_chk
          call
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• printf("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 104 0
         movl $.LC4, %esi
                  $1, %edi
         movl
                  %eax, %eax
         xorl
          call
                 __printf_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 120 0
• L("helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
         movq
                stdout(%rip), %rax
         movq %rax, %rcx
         movl
                $11, %edx
         movl
               $1, %esi
                  $.LC4, %edi
         movl
          call
                  fwrite
          .loc 1 120 0
• L("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
         movq stdout(%rip), %rcx
         movl
                  $11, %edx
                  $1, %esi
         movl
                  $.LC4, %edi
         movl
          call
                 fwrite
```

```
.LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• L("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 97 0
          movq stdout(%rip), %rcx
                 $11, %edx
          movl
                 $1, %esi
          movl
                 $.LC4, %edi
          movl
          call
                  fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 120 0
• DL("helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  $0, _dl_target(%rip)
          movl
          .loc 1 120 0
• DL("helloworldn"); with gcc -g -O1 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $0, _dl_target(%rip)
          .loc 1 120 0
• DL("helloworldn"); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                $0, _dl_target(%rip)
          .loc 1 120 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  $0, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
```

```
$16, %eax
          andl
                  %eax, %eax
          testl
          jе
                  .L5
          .loc 1 119 0 is_stmt 0 discriminator 1
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 120 0 is_stmt 1
• assert(i >= 2); with gcc -g -0 produces:
          .loc 1 119 0
 #NO_APP
                  24(%rsp), %eax
          movl
          {\tt cmpl}
                  $1, %eax
                  .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
                  $119, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 120 0 is_stmt 1
• TRAD_assert(i >= 2); with gcc -g -O produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $1, %eax
                  .L5
          jg
  .LVL5:
  .LBB783:
  .LBB784:
  .LBB785:
          .loc 2 97 0 discriminator 1
                  $.LC4, %r9d
          movl
          movl
                  $119, %r8d
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
                  $1, %esi
          movl
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
          call
                  __fprintf_chk
```

```
.LVL6:
  .LBE785:
  .LBE784:
          .loc 1 119 0 discriminator 1
          movl
                $1, %edi
                  exit
          call
  .LVL7:
  .L5:
  .LBE783:
          .loc 1 120 0
• I(i >= 2); with gcc -g -O produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $1, %eax
                   .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1 \,
                  $119, %edx
          movl
          movl
                  $.LC4, %esi
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 120 0 is_stmt 1
• DI(i \geq= 2); with gcc -g -O produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 120 0
• I(A(int i=0, i!=10, i++, a[i]>=0)); with gcc -g -0 produces:
  .LVL5:
 #NO_APP
  .LBB269:
          .loc 1 119 0
          cmpl
                  $0, a(%rip)
                  .L5
          js
                  $a+4, %eax
          movl
          movl
                  $a+40, %edx
  .LVL6:
  .L7:
          .loc 1 119 0 is_stmt 0 discriminator 2
                  $0, (%rax)
          cmpl
          js
                   .L5
```

```
$4, %rax
         addq
                 %rdx, %rax
          cmpq
          jne
                  .L7
  .LVL7:
  .L6:
  .LBE269:
          .loc 1 120 0 is_stmt 1
• d = now(); with gcc -g -O produces:
          .loc 1 119 0
 #NO_APP
          call
                  now
  .LVL5:
         movsd %xmm0, 24(%rsp)
          .loc 1 120 0
• printf("helloworldn"); with gcc -g -O produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
         movl $.LC4, %esi
                 $1, %edi
         movl
               $0, %eax
         movl
          call
                 __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• L("helloworldn"); with gcc -g -O produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
         movq stdout(%rip), %rcx
                 $11, %edx
         movl
               $1, %esi
         movl
               $.LC4, %edi
         movl
          call
                 fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0
```

.loc 1 119 0 #NO_APP movl \$0, _dl_target(%rip) .loc 1 120 0 • I(A(int i=0, i < 1*1024, i++, za[i] == 0)); with gcc -g -DNT=16 produces: #NO_APP .LBB2: .loc 1 119 0 \$1, -2104(%rbp) movl movl \$0, -2100(%rbp) jmp .L5 .L8: .loc 1 119 0 is_stmt 0 discriminator 2 movl -2100(%rbp), %eax cltq za(, %rax, 4), %eax movl testl %eax, %eax jе .L6 .loc 1 119 0 discriminator 1 \$0, -2104(%rbp) movl jmp .L7 .L6: .loc 1 119 0 discriminator 2 \$1, -2100(%rbp) addl .L5: .loc 1 119 0 discriminator 1 \$1023, -2100(%rbp) cmpl jle .L8 .L7: -2104(%rbp), %eax movl .LBE2: testl %eax, %eax .L9 jne \$119, %edx movl\$.LC4, %esi movl \$.LC5, %edi movl call _I_default_handler .L9: .loc 1 120 0 is_stmt 1 • I(A(int i=0, i < 2*1024, i++, za[i] == 0)); with gcc -g -DNT=16 produces:

• DL("helloworldn"); with gcc -g -O produces:

#NO_APP

```
.LBB2:
          .loc 1 119 0
          movl
                  $1, -2104(%rbp)
          movl
                  $0, -2100(%rbp)
                  .L5
          jmp
  .L8:
          .loc 1 119 0 is_stmt 0 discriminator 2
                  -2100(%rbp), %eax
          movl
          cltq
                  za(, %rax, 4), %eax
          movl
                  %eax, %eax
          testl
                  .L6
          jе
          .loc 1 119 0 discriminator 1
          movl
                  $0, -2104(%rbp)
          jmp
                  .L7
  .L6:
          .loc 1 119 0 discriminator 2
                  $1, -2100(%rbp)
          addl
  .L5:
          .loc 1 119 0 discriminator 1
                  $2047, -2100(%rbp)
          cmpl
                  .L8
          jle
  .L7:
          movl
                  -2104(%rbp), %eax
  .LBE2:
                  %eax, %eax
          testl
          jne
                  .L9
                  $119, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L9:
          .loc 1 120 0 is_stmt 1
• I(A(int i=0, i < 4*1024, i++, za[i] == 0)); with gcc -g -DNT=16
 produces:
 #NO_APP
  .LBB2:
          .loc 1 119 0
          movl
                  $1, -2104(%rbp)
                  $0, -2100(%rbp)
          movl
                  .L5
          jmp
  .L8:
          .loc 1 119 0 is_stmt 0 discriminator 2
          movl
                  -2100(%rbp), %eax
          cltq
                  za(, %rax, 4), %eax
          movl
                  %eax, %eax
          testl
                   .L6
          jе
```

```
.loc 1 119 0 discriminator 1
                  $0, -2104(%rbp)
          movl
          jmp
                  .L7
  .L6:
          .loc 1 119 0 discriminator 2
                  $1, -2100(%rbp)
          addl
  .L5:
          .loc 1 119 0 discriminator 1
                  $4095, -2100(%rbp)
          cmpl
          jle
                  .L8
  .L7:
                  -2104(%rbp), %eax
          movl
  .LBE2:
          testl
                  %eax, %eax
          jne
                  .L9
                  $119, %edx
          movl
          movl
                  $.LC4, %esi
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L9:
          .loc 1 120 0 is_stmt 1
• I(A(int i=0, i < 8*1024, i++, za[i] == 0)); with gcc -g -DNT=16
 produces:
 #NO_APP
  .LBB2:
          .loc 1 119 0
                  $1, -2104(%rbp)
          movl
          movl
                  $0, -2100(%rbp)
                  .L5
          jmp
  .L8:
          .loc 1 119 0 is_stmt 0 discriminator 2
                  -2100(%rbp), %eax
          movl
          cltq
          movl
                  za(, %rax, 4), %eax
          testl
                  %eax, %eax
          jе
                  .L6
          .loc 1 119 0 discriminator 1
                  $0, -2104(%rbp)
          movl
                  .L7
          jmp
  .L6:
          .loc 1 119 0 discriminator 2
                  $1, -2100(%rbp)
          addl
  .L5:
          .loc 1 119 0 discriminator 1
          cmpl
                  $8191, -2100(%rbp)
                  .L8
          jle
  .L7:
          movl
                  -2104(%rbp), %eax
```

```
.LBE2:
                  %eax, %eax
          testl
          jne
                  .L9
          movl
                  $119, %edx
          movl
                  $.LC4, %esi
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L9:
          .loc 1 120 0 is_stmt 1
• I(A(int i=0, i < 16*1024, i++, za[i] == 0)); with gcc -g -DNT=16
 produces:
 #NO_APP
  .LBB2:
          .loc 1 119 0
                  $1, -2104(%rbp)
          movl
                  $0, -2100(%rbp)
          movl
          jmp
                  .L5
  .L8:
          .loc 1 119 0 is_stmt 0 discriminator 2
                  -2100(%rbp), %eax
          movl
          cltq
          movl
                  za(, %rax, 4), %eax
          testl
                  %eax, %eax
          jе
                  .L6
          .loc 1 119 0 discriminator 1
                  $0, -2104(%rbp)
          movl
                  .L7
          jmp
  .L6:
          .loc 1 119 0 discriminator 2
          addl
                  $1, -2100(%rbp)
  .L5:
          .loc 1 119 0 discriminator 1
          cmpl
                  $16383, -2100(%rbp)
          jle
                  .L8
  .L7:
          movl
                  -2104(%rbp), %eax
  .LBE2:
                  %eax, %eax
          testl
                  .L9
          jne
                  $119, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L9:
          .loc 1 120 0 is_stmt 1
• I(A(int i=0, i < 32*1024, i++, za[i] == 0)); with gcc -g -DNT=16
```

produces:

```
#NO_APP
  .LBB2:
          .loc 1 119 0
          movl
                  $1, -2104(%rbp)
          movl
                  $0, -2100(%rbp)
                   .L5
          jmp
  .L8:
          .loc 1 119 0 is_stmt 0 discriminator 2
          movl
                  -2100(%rbp), %eax
          cltq
                  za(, %rax, 4), %eax
          movl
                  %eax, %eax
          testl
                   .L6
          jе
          .loc 1 119 0 discriminator 1
          movl
                  $0, -2104(%rbp)
                   .L7
          jmp
  .L6:
          .loc 1 119 0 discriminator 2
                  $1, -2100(%rbp)
  .L5:
          .loc 1 119 0 discriminator 1
                  $32767, -2100(%rbp)
          cmpl
          jle
                   .L8
  .L7:
          movl
                  -2104(%rbp), %eax
  .LBE2:
                  %eax, %eax
          testl
                   .L9
          jne
                  $119, %edx
          movl
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L9:
          .loc 1 120 0 is_stmt 1
• I(A(int i=0, i < 64*1024, i++, za[i] == 0)); with gcc -g -DNT=16
 produces:
 #NO APP
  .LBB2:
          .loc 1 119 0
                  $1, -2104(%rbp)
          movl
                  $0, -2100(%rbp)
          movl
                   .L5
          jmp
  .L8:
          .loc 1 119 0 is_stmt 0 discriminator 2
          movl
                  -2100(%rbp), %eax
          cltq
                  za(, %rax, 4), %eax
          movl
                  %eax, %eax
          testl
```

```
jе
                  .L6
          .loc 1 119 0 discriminator 1
          movl
                  $0, -2104(%rbp)
          jmp
                  .L7
  .L6:
          .loc 1 119 0 discriminator 2
                  $1, -2100(%rbp)
          addl
  .L5:
          .loc 1 119 0 discriminator 1
                  $65535, -2100(%rbp)
          cmpl
          jle
                  .L8
  .L7:
                  -2104(%rbp), %eax
          movl
  .LBE2:
          testl
                  %eax, %eax
          jne
                  .L9
          movl
                  $119, %edx
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L9:
          .loc 1 120 0 is_stmt 1
• I(A(int i=0, i < 128*1024, i++, za[i] == 0)); with gcc -g -DNT=16
 produces:
 #NO_APP
  .LBB2:
          .loc 1 119 0
          movl
                  $1, -2104(%rbp)
          movl
                  $0, -2100(%rbp)
                  .L5
          jmp
  .L8:
          .loc 1 119 0 is_stmt 0 discriminator 2
          movl
                  -2100(%rbp), %eax
          cltq
                  za(,%rax,4), %eax
          movl
          testl
                  %eax, %eax
                  .L6
          jе
          .loc 1 119 0 discriminator 1
          movl
                  $0, -2104(%rbp)
          jmp
                  .L7
  .L6:
          .loc 1 119 0 discriminator 2
          addl
                  $1, -2100(%rbp)
  .L5:
          .loc 1 119 0 discriminator 1
                  $131071, -2100(%rbp)
          cmpl
                  .L8
          jle
  .L7:
```

```
movl -2104(\%rbp), \%eax
  .LBE2:
         testl %eax, %eax
                .L9
          jne
         movl $119, %edx
         movl $.LC4, %esi
         movl $.LC5, %edi
         call _I_default_handler
  .L9:
          .loc 1 120 0 is_stmt 1
• asm(""); with gcc -g -00 produces:
          .loc 1 119 0
• asm(""); with gcc -g -01 produces:
         .loc 1 119 0
• asm(""); with gcc -g -03 produces:
          .loc 1 119 0
• asm("nop"); with gcc -g -00 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
```

```
\bullet asm("nop;nop;"); with gcc -g -01 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
```

0 "" 2

```
\bullet asm(""); with gcc -g -00 produces:
          .loc 1 119 0
• asm(""); with gcc -g -01 produces:
          .loc 1 119 0
• asm(""); with gcc -g -03 produces:
          .loc 1 119 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
```

 \bullet asm("nop;nop;"); with gcc -g -01 produces:

.loc 1 119 0 # 119 "tmp.c" 1 nop;nop; # 0 "" 2

• asm("nop;nop;"); with gcc -g -03 produces:

```
.loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm(""); with gcc -g -00 produces:
          .loc 1 119 0
• asm(""); with gcc -g -01 produces:
          .loc 1 119 0
\bullet asm(""); with gcc -g -03 produces:
          .loc 1 119 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
```

• asm("nop"); with gcc -g -01 produces:

```
.loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;"); with gcc -g -03 produces:

```
# 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -O1 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
```

#NO_APP

movl \$4, -56(%rbp) .loc 1 120 0

• i = 4; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

movl \$4, 24(%rsp) .loc 1 120 0

• i = 4; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movl \$4, 24(%rsp) .loc 1 120 0

• gi = 11; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl \$11, gi(%rip) .loc 1 120 0

• gi = 11; with gcc - g - 01 produces:

.loc 1 119 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 120 0

• gi = 11; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movl \$11, gi(%rip) .loc 1 120 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl .LC4(%rip), %eax movl %eax, -48(%rbp) .loc 1 120 0

• f = 12.0; with gcc -g -01 produces:

movss .LC4(%rip), %xmm0 movss %xmm0, 28(%rsp)

.loc 1 120 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

#NO_APP

\$0x41400000, 28(%rsp) movl .loc 1 120 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

.LC4(%rip), %eax movl movl %eax, gf(%rip) .loc 1 120 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

.LC4(%rip), %xmm0 movss movss %xmm0, gf(%rip) .loc 1 120 0

• gf = 12.0; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

\$0x41400000, gf(%rip) movl .loc 1 120 0

• i++; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

-56(%rbp), %eax movl addl \$1, %eax %eax, -56(%rbp) movl

.loc 1 120 0

• i++; with gcc -g -01 produces:

```
movl
                  24(%rsp), %eax
          addl
                  $1, %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 120 0
• i++; with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
                  24(%rsp), %eax
          movl
          addl
                  $1, %eax
                  %eax, 24(%rsp)
          movl
          .loc 1 120 0
• gi++; with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  gi(%rip), %eax
          movl
                  $1, %eax
          addl
                  %eax, gi(%rip)
          movl
          .loc 1 120 0
• gi++; with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
                  gi(%rip), %eax
          movl
          addl
                  $1, %eax
          movl
                  %eax, gi(%rip)
          .loc 1 120 0
• gi++; with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  gi(%rip), %eax
          addl
                  $1, %eax
                  %eax, gi(%rip)
          movl
          .loc 1 120 0
• j = a[i]; with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  -60(%rbp), %eax
          cltq
                  a(, %rax, 4), %eax
          movl
                  %eax, -56(%rbp)
          movl
          .loc 1 120 0
```

#NO_APP

```
• j = a[i]; with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
                  20(%rsp), %eax
          movl
          cltq
          movl
                  a(,%rax,4), %eax
                  %eax, 24(%rsp)
          movl
          .loc 1 120 0
• j = a[i]; with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movslq 20(%rsp), %rax
                  a(,\%rax,4), \%eax
          movl
                  %eax, 24(%rsp)
          .loc 1 120 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
          movl
                  $119, %edx
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L5:
          .loc 1 120 0 is_stmt 1
• assert(i >= 10); with gcc -g -O1 produces:
          .loc 1 119 0
 #NO_APP
                  24(%rsp), %eax
          movl
                  $9, %eax
          cmpl
                  .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
                  $119, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 120 0 is_stmt 1
```

```
.loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 120 0
• BSD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  -56(%rbp), %eax
                  $9, %eax
          cmpl
                  .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
          movl
                  $.LC4, %edx
                  $119, %esi
          movl
          movl
                  $.LC5, %edi
                  __BSD_assert
          call
  .L5:
          .loc 1 120 0 is_stmt 1
• BSD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 120 0
• BSD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 120 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
  .LBB2:
          .loc 1 119 0 is_stmt 0 discriminator 1
                  stderr(%rip), %rax
          movq
```

• assert(i >= 10); with gcc -g -03 produces:

```
$.LC4, %r8d
          movl
                  $119, %ecx
          movl
                  $.LC5, %edx
          movl
          movl
                  $.LC6, %esi
                  %rax, %rdi
          movq
                  $0, %eax
          movl
                  fprintf
          call
                  $1, %edi
          movl
          call
                  exit
  .L5:
  .LBE2:
          .loc 1 120 0 is_stmt 1 \,
• TRAD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          {\tt cmpl}
                  $9, %eax
                   .L5
          jg
  .LVL5:
  .LBB783:
  .LBB784:
  .LBB785:
          .loc 2 97 0 discriminator 1
                $.LC4, %r9d
          movl
                  $119, %r8d
          movl
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
          movl
                  $1, %esi
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
          call
                  __fprintf_chk
  .LVL6:
  .LBE785:
  .LBE784:
          .loc 1 119 0 discriminator 1
                  $1, %edi
          movl
          call
                  exit
  .LVL7:
  .L5:
  .LBE783:
          .loc 1 120 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
```

```
$9, %eax
          cmpl
                  .L21
          jle
          .loc 1 120 0
• I(i \geq 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
          movl
                  $119, %edx
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L5:
          .loc 1 120 0 is_stmt 1
• I(i >= 10); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
          movl
                  $119, %edx
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 120 0 is_stmt 1
• I(i >= 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
          jle
                  .L274
  .L5:
          .loc 1 120 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 120 0
```

```
• DI(i >= 10); with gcc -g -O1 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 120 0
• DI(i \geq 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 120 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
 produces:
 #NO_APP
  .LBB2:
          .loc 1 119 0
          movl
                  $1, -3132(%rbp)
                  $str, -2096(%rbp)
          movq
                  .L5
          jmp
  .L8:
          .loc 1 119 0 is_stmt 0 discriminator 2
          call
                  __ctype_b_loc
                  (%rax), %rdx
          movq
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
                  %rax, %rax
          addq
          addq
                  %rdx, %rax
          movzwl (%rax), %eax
          movzwl %ax, %eax
                  $512, %eax
          andl
          testl %eax, %eax
          jne
                  .L6
          .loc 1 119 0 discriminator 1
                  $0, -3132(%rbp)
          movl
                  .L7
          jmp
  .L6:
          .loc 1 119 0 discriminator 2
          addq
                  $1, -2096(%rbp)
  .L5:
          .loc 1 119 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          testb
                  %al, %al
```

jne

.L8

```
.L7:
                  -3132(%rbp), %eax
          movl
  .LBE2:
          testl
                 %eax, %eax
                  .L9
          jne
                  $119, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L9:
          .loc 1 120 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -01
 produces:
  .LVL5:
 #NO_APP
  .LBB269:
          .loc 1 119 0
          movzbl str(%rip), %ebx
          testb
                  %bl, %bl
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
          movq
                  (%rax), %rdx
          movl
                  $str, %eax
  .LVL7:
  .L7:
          .loc 1 119 0 is_stmt 0 discriminator 2
          movsbq %bl, %rbx
                  $2, 1(%rdx,%rbx,2)
          testb
                  .L6
          jе
          addq
                  $1, %rax
  .LVL8:
          movzbl (%rax), %ebx
                  %bl, %bl
          testb
          jne
                  .L7
  .LVL9:
  .L5:
  .LBE269:
          .loc 1 120 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -03
 produces:
  .LVL5:
 #NO_APP
  .LBB271:
          .loc 1 119 0
```

```
movsbq str(%rip), %rbx
                  %bl, %bl
          testb
          jе
                  .L5
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
                  .L7
          jmp
  .LVL7:
          .p2align 4,,10
          .p2align 3
  .L2074:
          .loc 1 119 0 is_stmt 0 discriminator 2
          addq
                  $1, %rax
  .LVL8:
          movsbq (%rax), %rbx
          testb
                  %bl, %bl
                  .L5
          jе
  .LVL9:
  .L7:
                  $2, 1(%rdx,%rbx,2)
          testb
                  .L2074
          jne
  .LVL10:
  .LBE271:
          .loc 1 119 0
                  $119, %edx
          movl
          movl
                  $.LC11, %esi
                  $.LC12, %edi
          movl
                  _I_default_handler
          call
  .LVL11:
  .L5:
          .loc 1 120 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -00 produces:
 #NO_APP
  .LBB2:
          .loc 1 119 0
                  $1, -5188(%rbp)
          movl
                  $0, -5184(%rbp)
          movl
          jmp
                  .L5
  .L11:
  .LBB3:
          .loc 1 119 0 is_stmt 0 discriminator 2
          movq
                  $0, -2096(%rbp)
          movl
                  $0, -5180(%rbp)
                  .L6
          jmp
  .L8:
                  -5184(%rbp), %eax
          movl
```

```
cltq
                  a(, %rax, 4), %edx
          {\tt movl}
          movl
                  -5180(%rbp), %eax
          cltq
                  a(, %rax, 4), %eax
          movl
                  %eax, %edx
          cmpl
          jne
                  .L7
          .loc 1 119 0 discriminator 1
          addq
                  $1, -2096(%rbp)
  .L7:
          .loc 1 119 0 discriminator 2
                  $1, -5180(%rbp)
          addl
  .L6:
          .loc 1 119 0 discriminator 1
          cmpl
                  $9, -5180(%rbp)
          jle
                  .L8
          .loc 1 119 0 discriminator 3
          movq
                  -2096(%rbp), %rax
  .LBE3:
          cmpq
                  $1, %rax
                  .L9
          jе
          .loc 1 119 0 discriminator 1
                  $0, -5188(%rbp)
          movl
          jmp
                  .L10
  .L9:
          .loc 1 119 0 discriminator 2
          addl
                  $1, -5184(%rbp)
  .L5:
          .loc 1 119 0 discriminator 1
          cmpl
                  $9, -5184(%rbp)
                  .L11
          jle
  .L10:
                  -5188(%rbp), %eax
          movl
  .LBE2:
          testl
                  %eax, %eax
          jne
                  .L12
                  $119, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L12:
          .loc 1 120 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
          .loc 1 119 0
```

```
$0, %ecx
          movl
  .LBE526:
          .loc 1 118 0
          movl
                  $0, %eax
          movl
                  $0, %edx
                  .L5
          jmp
  .LVL6:
  .L1295:
  .LBB528:
                  $0, %eax
          movl
  .LVL7:
                  $0, %edx
          movl
  .LVL8:
  .L5:
  .LBB527:
          .loc 1 119 0 discriminator 2
          movslq %ecx, %rdi
          movslq %eax, %rsi
                  a(,%rsi,4), %ebx
          movl
          cmpl
                  %ebx, a(,%rdi,4)
                  %sil
          sete
          movzbl %sil, %esi
          addq
                  %rsi, %rdx
  .LVL9:
          addl
                  $1, %eax
  .LVL10:
          cmpl
                  $9, %eax
                  .L5
          jle
  .LBE527:
          .loc 1 119 0 is_stmt 0 discriminator 3
                  $1, %rdx
          cmpq
                  .L8
          jne
          .loc 1 119 0 discriminator 2
                  $1, %ecx
          addl
  .LVL11:
          cmpl
                  $9, %ecx
                  .L1295
          jle
  .LVL12:
  .L1038:
  .LBE528:
          .loc 1 120 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -03 produces:
  .LVL5:
 #NO_APP
  .LBB528:
  .LBB529:
          .loc 1 119 0
```

```
a(%rip), %r13d
        movl
                a+4(%rip), %r12d
        movl
        movl
                $a, %ecx
        movl
                a+8(%rip), %ebp
                a+12(%rip), %ebx
        movl
                a+16(%rip), %r11d
        movl
                a+20(%rip), %r10d
        movl
                a+24(%rip), %r9d
        movl
                a+28(%rip), %r8d
        movl
                a+32(%rip), %edi
        movl
                a+36(%rip), %esi
        movl
.LVL6:
.L5:
        movl
                (%rcx), %edx
        xorl
                %eax, %eax
                %r13d, %edx
        cmpl
        sete
                %al
.LVL7:
                %r14d, %r14d
        xorl
        cmpl
                %r12d, %edx
                %r14b
        sete
                %r14, %rax
        addq
.LVL8:
                %r14d, %r14d
        xorl
        cmpl
                %ebp, %edx
                %r14b
        sete
                %r14, %rax
        addq
.LVL9:
                %r14d, %r14d
        xorl
        cmpl
                %ebx, %edx
                %r14b
        sete
                %r14, %rax
        addq
.LVL10:
                %r14d, %r14d
        xorl
        cmpl
                %r11d, %edx
        sete
                %r14b
                %r14, %rax
        addq
.LVL11:
                %r14d, %r14d
        xorl
                %r10d, %edx
        cmpl
        sete
                %r14b
                %r14, %rax
        addq
.LVL12:
                %r14d, %r14d
        xorl
        cmpl
                %r9d, %edx
        sete
                %r14b
        {\tt addq}
                %r14, %rax
.LVL13:
                %r14d, %r14d
        xorl
        cmpl
                %r8d, %edx
```

```
%r14b
          sete
                  %r14, %rax
          addq
  .LVL14:
          xorl
                  %r14d, %r14d
          cmpl
                  %edi, %edx
                  %r14b
          sete
                  %r14, %rax
          addq
  .LVL15:
                  %esi, %edx
          cmpl
                  %dl
          sete
          movzbl %dl, %edx
          addq
                  %rdx, %rax
  .LVL16:
  .LBE529:
                  $1, %rax
          cmpq
                  .L4164
          jne
                  $4, %rcx
          addq
          .loc 1 119 0 is_stmt 0 discriminator 2
                  $a+40, %rcx
          cmpq
          jne
                  .L5
  .LVL17:
  .L6:
  .LBE528:
          .loc 1 120 0 is_stmt 1
• printf("helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $.LC4, %edi
                  $0, %eax
          movl
                  printf
          call
          .loc 1 120 0
• printf("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
                  $.LC4, %esi
          movl
          movl
                  $1, %edi
                  $0, %eax
          movl
          call
                  __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0
```

```
• printf("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 2 104 0
         movl $.LC4, %esi
                 $1, %edi
         movl
         xorl %eax, %eax
         call __printf_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
         .loc 1 120 0
• L("helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
               stdout(%rip), %rax
         movq
         movq %rax, %rcx
         movl
                 $11, %edx
         movl
                 $1, %esi
                 $.LC4, %edi
         movl
         call
                 fwrite
          .loc 1 120 0
• L("helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
         movq stdout(%rip), %rcx
                 $11, %edx
         movl
         movl
                 $1, %esi
               $.LC4, %edi
         movl
         call
               fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• L("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
```

```
.LBB1050:
  .LBB1051:
          .loc 2 97 0
         movq stdout(%rip), %rcx
         movl $11, %edx
         movl $1, %esi
               $.LC4, %edi
         movl
         call
                  fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 120 0
• DL("helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  $0, _dl_target(%rip)
         movl
          .loc 1 120 0
• DL("helloworldn"); with gcc -g -O1 produces:
          .loc 1 119 0
 #NO_APP
                  $0, _dl_target(%rip)
         movl
          .loc 1 120 0
• DL("helloworldn"); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
                  $0, _dl_target(%rip)
         movl
          .loc 1 120 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  $0, gi(%rip)
         movl
         movl
                  gi(%rip), %eax
                  $16, %eax
          andl
                  %eax, %eax
          testl
                  .L5
          jе
          .loc 1 119 0 is_stmt 0 discriminator 1
         movq
                  stdout(%rip), %rax
                  %rax, %rcx
         movq
                  $11, %edx
         movl
                  $1, %esi
          movl
                  $.LC4, %edi
         movl
```

```
call
                  fwrite
  .L5:
          .loc 1 120 0 is_stmt 1
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
                  $0, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
                  .L5
          jе
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
          movq
                  stdout(%rip), %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
          jne
                  .L1045
  .L5:
          .loc 1 120 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
          andl
                  $16, %eax
                  %eax, %eax
          testl
          jе
                  .L5
          .loc 1 119 0 is_stmt 0 discriminator 1
                  stdout(%rip), %rax
          movq
```

```
%rax, %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
          movl
                  $.LC4, %edi
          call
                  fwrite
  .L5:
          .loc 1 120 0 is_stmt 1
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
                  .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
          movq
                  stdout(%rip), %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
                  $-1, gi(%rip)
          movl
                  gi(%rip), %eax
          movl
                  $16, %al
          testb
          jne
                  .L1045
  .L5:
          .loc 1 120 0
• LHP(fprintf,log, "helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  -32(%rbp), %rax
          movq
                  %rax, %rcx
          movq
          movl
                  $11, %edx
```

```
$1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
          .loc 1 120 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
          movq %rbx, %rcx
                $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• LHP(fprintf,log, "helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 97 0
               $11, %edx
          movl
                  %rbx, %rcx
          movq
                 $1, %esi
          movl
                $.LC4, %edi
          movl
          call
                  fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 120 0
\bullet LHP(L_buffer_printf,buf,"helloworldn"); with gcc \, -g \, -00 \rm produces:
          .loc 1 119 0
 #NO_APP
          movq
                  -40(%rbp), %rax
          movl
                  $.LC4, %esi
                  %rax, %rdi
          movq
                  $0, %eax
          movl
          call
                  L_buffer_printf
```

.loc 1 120 0

• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -01 produces: .loc 1 119 0 #NO_APP \$.LC4, %esi movl movq %rbx, %rdi **\$0,** %eax movl L_buffer_printf call .LVL6: .loc 1 120 0 • LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -03 produces: .loc 1 119 0 #NO_APP \$.LC4, %esi movl %rbx, %rdi movq %eax, %eax xorl L_buffer_printf call .LVL279: .loc 1 120 0 • LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -OO produces: .loc 1 119 0 #NO_APP movl \$.LC4, %esi \$8, %edi movl **\$0,** %eax movl call syslog .loc 1 120 0 • LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces: .LVL5: #NO_APP .LBB526: .LBB527: .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h" .loc 3 31 0 \$.LC4, %edx movl \$1, %esi movl \$8, %edi movl movl \$0, %eax

__syslog_chk

call

.loc 1 120 0

.LVL6: .LBE527: .LBE526:

```
• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O3 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 3 31 0
          movl $.LC4, %edx movl $1, %esi
          movl $8, %edi
          xorl %eax, %eax
          call __syslog_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 120 0
• asm(""); with gcc -g -00 produces:
          .loc 1 119 0
• asm(""); with gcc -g -01 produces:
          .loc 1 119 0
\bullet asm(""); with gcc -g -03 produces:
          .loc 1 119 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
```

```
\bullet asm("nop;nop;"); with gcc -g -00 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
```

```
.loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
         movl $4, -56(%rbp)
         .loc 1 120 0
• i = 4; with gcc -g -01 produces:
         .loc 1 119 0
 #NO_APP
               $4, 24(%rsp)
         movl
          .loc 1 120 0
```

• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:

• i = 4; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movl \$4, 24(%rsp) .loc 1 120 0

• gi = 11; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl \$11, gi(%rip) .loc 1 120 0

• gi = 11; with gcc - g - 01 produces:

.loc 1 119 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 120 0

• gi = 11; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 120 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, -48(%rbp)
.loc 1 120 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

movss .LC4(%rip), %xmm0 movss %xmm0, 28(%rsp) .loc 1 120 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movl \$0x41400000, 28(%rsp)
.loc 1 120 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 120 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, gf(%rip)
.loc 1 120 0

• gf = 12.0; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movl \$0x41400000, gf(%rip) .loc 1 120 0

• i++; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl -56(%rbp), %eax addl \$1, %eax movl %eax, -56(%rbp) .loc 1 120 0

• i++; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

movl 24(%rsp), %eax addl \$1, %eax movl %eax, 24(%rsp) .loc 1 120 0

• i++; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movl 24(%rsp), %eax addl \$1, %eax movl %eax, 24(%rsp) .loc 1 120 0 • gi++; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 120 0

• gi++; with gcc -g -01 produces:

.loc 1 119 0
#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 120 0

• gi++; with gcc -g -03 produces:

.loc 1 119 0
#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 120 0

• j = a[i]; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl -60(%rbp), %eax
cltq
movl a(,%rax,4), %eax
movl %eax, -56(%rbp)
.loc 1 120 0

• j = a[i]; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

movl 20(%rsp), %eax
cltq
movl a(,%rax,4), %eax
movl %eax, 24(%rsp)
.loc 1 120 0

• j = a[i]; with gcc -g -03 produces:

```
.loc 1 119 0
 #NO_APP
          movslq 20(%rsp), %rax
          movl
                  a(,%rax,4), %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 120 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  -56(%rbp), %eax
                  $9, %eax
          cmpl
          jg
                  .L5
          .loc 1 119 0 is_stmt 0 discriminator 1
               $119, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L5:
          .loc 1 120 0 is_stmt 1
• assert(i >= 10); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
                  $9, %eax
          cmpl
          jg
                  .L5
          .loc 1 119 0 is_stmt 0 discriminator 1
                  $119, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 120 0 is_stmt 1
• assert(i \geq= 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 120 0
```

• BSD_assert(i >= 10); with gcc -g -00 produces:

```
.loc 1 119 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
               $.LC4, %edx
          movl
                  $119, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  __BSD_assert
  .L5:
          .loc 1 120 0 is_stmt 1
• BSD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 120 0
• BSD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 120 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  -56(%rbp), %eax
          movl
                  $9, %eax
          cmpl
          jg
                  .L5
  .LBB2:
          .loc 1 119 0 is_stmt 0 discriminator 1
          movq stderr(%rip), %rax
                  $.LC4, %r8d
          movl
                  $119, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
                  %rax, %rdi
          movq
                  $0, %eax
          movl
                  fprintf
          call
                  $1, %edi
          movl
          call
                  exit
  .L5:
  .LBE2:
          .loc 1 120 0 is_stmt 1
```

.loc 1 119 0 #NO_APP movl 24(%rsp), %eax \$9, %eax cmpl .L5 jg .LVL5: .LBB783: .LBB784: .LBB785: .loc 2 97 0 discriminator 1 movl \$.LC4, %r9d \$119, %r8d movl \$.LC5, %ecx movl \$.LC6, %edx movl movl \$1, %esi movq stderr(%rip), %rdi \$0, %eax movl call __fprintf_chk .LVL6: .LBE785: .LBE784: .loc 1 119 0 discriminator 1 \$1, %edi movl call exit .LVL7: .L5: .LBE783: .loc 1 120 0 • TRAD_assert(i >= 10); with gcc -g -03 produces: .loc 1 119 0 #NO_APP movl 24(%rsp), %eax cmpl \$9, %eax jle .L21 .loc 1 120 0 • I(i >= 10); with gcc -g -00 produces: .loc 1 119 0 #NO_APP -56(%rbp), %eax movl cmpl\$9, %eax jg .L5 .loc 1 119 0 is_stmt 0 discriminator 1 movl \$119, %edx

• TRAD_assert(i >= 10); with gcc -g -01 produces:

```
$.LC4, %esi
          movl
                   $.LC5, %edi
          movl
          call
                   _I_default_handler
  .L5:
          .loc 1 120 0 is_stmt 1
• I(i >= 10); with gcc -g -01 produces:
           .loc 1 119 0
  #NO_APP
                   24(%rsp), %eax
          {\tt movl}
                   $9, %eax
          cmpl
          jg
                   .L5
          .loc 1 119 0 is_stmt 0 discriminator 1
          movl
                   $119, %edx
                   $.LC4, %esi
          movl
                   $.LC5, %edi
          movl
          call
                   _{\rm I\_default\_handler}
  .LVL5:
  .L5:
          .loc 1 120 0 is_stmt 1
• I(i \geq= 10); with gcc -g -03 produces:
           .loc 1 119 0
  #NO_APP
          movl
                   24(%rsp), %eax
          {\tt cmpl}
                   $9, %eax
                   .L274
          jle
  .L5:
          .loc 1 120 0
• DI(i \geq 10); with gcc -g -00 produces:
           .loc 1 119 0
  #NO_APP
                   $0, _di_target(%rip)
          movl
          .loc 1 120 0
• DI(i \geq= 10); with gcc -g -01 produces:
           .loc 1 119 0
  #NO_APP
          movl
                   $0, _di_target(%rip)
          .loc 1 120 0
```

• DI(i >= 10); with gcc -g -03 produces:

```
.loc 1 119 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 120 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
 produces:
 #NO_APP
  .LBB2:
          .loc 1 119 0
          movl
                  $1, -3132(%rbp)
                  $str, -2096(%rbp)
          movq
          jmp
                  .L5
  .L8:
          .loc 1 119 0 is_stmt 0 discriminator 2
                  __ctype_b_loc
          call
                  (%rax), %rdx
          movq
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
                  %rax, %rax
          addq
          addq
                  %rdx, %rax
          movzwl (%rax), %eax
          movzwl %ax, %eax
                  $512, %eax
          andl
                  %eax, %eax
          testl
                  .L6
          jne
          .loc 1 119 0 discriminator 1
          movl
                  $0, -3132(%rbp)
                  .L7
          jmp
  .L6:
          .loc 1 119 0 discriminator 2
                  $1, -2096(%rbp)
  .L5:
          .loc 1 119 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          testb
                  %al, %al
                  .L8
          jne
  .L7:
                  -3132(%rbp), %eax
          movl
  .LBE2:
                  %eax, %eax
          testl
          jne
                  .L9
                  $119, %edx
          movl
          movl
                  $.LC4, %esi
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L9:
```

.loc 1 120 0 is_stmt 1

```
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -01
 produces:
  .LVL5:
 #NO_APP
  .LBB269:
          .loc 1 119 0
          movzbl str(%rip), %ebx
          testb
                  %bl, %bl
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
                  $str, %eax
          movl
  .LVL7:
  .L7:
          .loc 1 119 0 is_stmt 0 discriminator 2 \,
          movsbq %bl, %rbx
          testb
                  $2, 1(%rdx,%rbx,2)
          jе
                  .L6
          addq
                  $1, %rax
  .LVL8:
          movzbl (%rax), %ebx
          testb
                  %bl, %bl
                  .L7
          jne
  .LVL9:
  .L5:
  .LBE269:
          .loc 1 120 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -03
 produces:
  .LVL5:
 #NO_APP
  .LBB271:
          .loc 1 119 0
          movsbq str(%rip), %rbx
          testb
                 %bl, %bl
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
                  $str, %eax
          movl
                  .L7
          jmp
  .LVL7:
          .p2align 4,,10
```

```
.p2align 3
  .L2074:
          .loc 1 119 0 is_stmt 0 discriminator 2
          addq
                  $1, %rax
  .LVL8:
          movsbq (%rax), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
  .LVL9:
  .L7:
                  $2, 1(%rdx,%rbx,2)
          testb
          jne
                  .L2074
  .LVL10:
  .LBE271:
          .loc 1 119 0
                  $119, %edx
          movl
                  $.LC11, %esi
          movl
                  $.LC12, %edi
          movl
          call
                  _I_default_handler
  .LVL11:
  .L5:
          .loc 1 120 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -00 produces:
 #NO_APP
  .LBB2:
          .loc 1 119 0
                  $1, -5188(%rbp)
          movl
                  $0, -5184(%rbp)
                  .L5
          jmp
  .L11:
  .LBB3:
          .loc 1 119 0 is_stmt 0 discriminator 2
          movq
                  $0, -2096(%rbp)
                  $0, -5180(%rbp)
          movl
          jmp
                  .L6
  .L8:
                  -5184(%rbp), %eax
          movl
          cltq
          movl
                  a(,\%rax,4), \%edx
                  -5180(%rbp), %eax
          movl
          cltq
          movl
                  a(, %rax, 4), %eax
          cmpl
                  %eax, %edx
          jne
                  .L7
          .loc 1 119 0 discriminator 1
                  $1, -2096(%rbp)
          addq
  .L7:
```

```
.loc 1 119 0 discriminator 2
          addl
               $1, -5180(%rbp)
  .L6:
          .loc 1 119 0 discriminator 1
          cmpl $9, -5180(%rbp)
          jle
                  .L8
          .loc 1 119 0 discriminator 3
                  -2096(%rbp), %rax
         movq
  .LBE3:
                  $1, %rax
          cmpq
          jе
                  .L9
          .loc 1 119 0 discriminator 1
                  $0, -5188(%rbp)
         movl
          jmp
                  .L10
  .L9:
          .loc 1 119 0 discriminator 2
          addl
                $1, -5184(%rbp)
  .L5:
          .loc 1 119 0 discriminator 1
          cmpl
                  $9, -5184(%rbp)
                  .L11
          jle
  .L10:
         movl
                  -5188(%rbp), %eax
  .LBE2:
          testl
                 %eax, %eax
                  .L12
          jne
                  $119, %edx
         movl
                  $.LC4, %esi
         movl
                  $.LC5, %edi
         movl
          call
                  _I_default_handler
  .L12:
          .loc 1 120 0 is_stmt 1
• asm(""); with gcc -g -00 produces:
          .loc 1 119 0
• asm(""); with gcc -g -01 produces:
          .loc 1 119 0
• asm(""); with gcc -g -03 produces:
          .loc 1 119 0
• asm("nop"); with gcc -g -00 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
```

```
\bullet asm("nop"); with gcc -g -01 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
        nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
```

0 "" 2

```
• asm("nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm(""); with gcc -g -00 produces:
          .loc 1 119 0
• asm(""); with gcc -g -01 produces:
          .loc 1 119 0
• asm(""); with gcc -g -03 produces:
          .loc 1 119 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
\bullet asm("nop"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
```

• asm("nop;nop;"); with gcc -g -01 produces:

0 "" 2

```
.loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm(""); with gcc -g -00 produces:
          .loc 1 119 0
• asm(""); with gcc -g -01 produces:
          .loc 1 119 0
• asm(""); with gcc -g -03 produces:
          .loc 1 119 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
```

• asm("nop"); with gcc -g -03 produces:

```
.loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:

```
.loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
         movl $4, -56(%rbp)
          .loc 1 120 0
• i = 4; with gcc -g -01 produces:
```

.loc 1 119 0

#NO_APP

movl \$4, 24(%rsp) .loc 1 120 0

• i = 4; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movl \$4, 24(%rsp) .loc 1 120 0

• gi = 11; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 120 0

• gi = 11; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 120 0

• gi = 11; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 120 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl .LC4(%rip), %eax movl %eax, -48(%rbp) .loc 1 120 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

movss .LC4(%rip), %xmm0 movss %xmm0, 28(%rsp) .loc 1 120 0 • f = 12.0; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

\$0x41400000, 28(%rsp) movl .loc 1 120 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl .LC4(%rip), %eax %eax, gf(%rip) movl .loc 1 120 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

.LC4(%rip), %xmm0 movss %xmm0, gf(%rip) movss .loc 1 120 0

• gf = 12.0; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

\$0x41400000, gf(%rip) movl .loc 1 120 0

• i++; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

-56(%rbp), %eax movl \$1, %eax addl %eax, -56(%rbp) movl

.loc 1 120 0

• i++; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

24(%rsp), %eax movl addl \$1, %eax movl %eax, 24(%rsp) .loc 1 120 0

• i++; with gcc -g -03 produces:

```
.loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          addl
                  $1, %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 120 0
• gi++; with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  gi(%rip), %eax
                  $1, %eax
          addl
                  %eax, gi(%rip)
          movl
          .loc 1 120 0
• gi++; with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
                  gi(%rip), %eax
          movl
          addl
                  $1, %eax
          movl
                  %eax, gi(%rip)
          .loc 1 120 0
• gi++; with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
                  gi(%rip), %eax
          movl
                  $1, %eax
          addl
                  %eax, gi(%rip)
          movl
          .loc 1 120 0
• j = a[i]; with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  -60(%rbp), %eax
          movl
          cltq
          movl
                  a(,%rax,4), %eax
                  %eax, -56(%rbp)
          movl
          .loc 1 120 0
• j = a[i]; with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
```

20(%rsp), %eax

movl

```
cltq
                  a(, %rax, 4), %eax
          movl
                  %eax, 24(%rsp)
          movl
          .loc 1 120 0
• j = a[i]; with gcc -g -03 produces:
          .loc 1 119 0
  #NO_APP
          movslq 20(%rsp), %rax
                  a(,%rax,4), %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 120 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
  #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
          jg
                  .L5
          .loc 1 119 0 is_stmt 0 discriminator 1
                  $119, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L5:
          .loc 1 120 0 is_stmt 1
• assert(i >= 10); with gcc -g -O1 produces:
          .loc 1 119 0
  #NO_APP
          movl
                  24(%rsp), %eax
          {\tt cmpl}
                  $9, %eax
                  .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
                  $119, %edx
          movl
          movl
                  $.LC4, %esi
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 120 0 is_stmt 1
• assert(i \geq= 10); with gcc -g -03 produces:
          .loc 1 119 0
  #NO_APP
```

```
24(%rsp), %eax
          movl
                  $9, %eax
          cmpl
          jle
                  .L274
  .L5:
          .loc 1 120 0
• BSD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          {\tt cmpl}
                  $9, %eax
                  .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
          movl
                  $.LC4, %edx
          movl
                  $119, %esi
                  $.LC5, %edi
          movl
          call
                  __BSD_assert
  .L5:
          .loc 1 120 0 is_stmt 1
• BSD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
                  24(%rsp), %eax
          movl
          .loc 1 120 0
• BSD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 120 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          cmpl
                  $9, %eax
                  .L5
          jg
  .LBB2:
          .loc 1 119 0 is_stmt 0 discriminator 1
          movq
                  stderr(%rip), %rax
          movl
                  $.LC4, %r8d
                  $119, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
```

```
%rax, %rdi
          movq
                  $0, %eax
          {\tt movl}
          call
                  fprintf
          movl
                  $1, %edi
          call
                  exit
  .L5:
  .LBE2:
          .loc 1 120 0 is_stmt 1
• TRAD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                   .L5
          jg
  .LVL5:
  .LBB783:
  .LBB784:
  .LBB785:
          .loc 2 97 0 discriminator 1
                $.LC4, %r9d
          movl
          movl
                  $119, %r8d
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
                  $1, %esi
          movl
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
                  __fprintf_chk
          call
  .LVL6:
  .LBE785:
  .LBE784:
          .loc 1 119 0 discriminator 1
                  $1, %edi
          call
                  exit
  .LVL7:
  .L5:
  .LBE783:
          .loc 1 120 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
          jle
                  .L21
          .loc 1 120 0
```

```
.loc 1 119 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                   .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
                  $119, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L5:
          .loc 1 120 0 is_stmt 1
• I(i >= 10); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
                  24(%rsp), %eax
          movl
          {\tt cmpl}
                  $9, %eax
                   .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
                  $119, %edx
          movl
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 120 0 is_stmt 1
• I(i \geq= 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
          jle
                   .L274
  .L5:
          .loc 1 120 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 120 0
```

• DI(i >= 10); with gcc -g -01 produces:

```
.loc 1 119 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 120 0
• DI(i >= 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 120 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
 produces:
 #NO_APP
  .LBB2:
          .loc 1 119 0
          movl
                  $1, -3132(%rbp)
          movq
                  $str, -2096(%rbp)
          jmp
                  .L5
  .L8:
          .loc 1 119 0 is_stmt 0 discriminator 2
          call
                  __ctype_b_loc
          movq
                  (%rax), %rdx
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
          addq
                  %rax, %rax
                  %rdx, %rax
          addq
          movzwl (%rax), %eax
          movzwl %ax, %eax
          andl
                  $512, %eax
                  %eax, %eax
          testl
          jne
                  .L6
          .loc 1 119 0 discriminator 1
                  $0, -3132(%rbp)
          movl
          jmp
                  .L7
  .L6:
          .loc 1 119 0 discriminator 2
          addq
                  $1, -2096(%rbp)
  .L5:
          .loc 1 119 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          testb
                  %al, %al
                  .L8
          jne
  .L7:
          movl
                  -3132(%rbp), %eax
```

```
testl %eax, %eax
          jne
                  .L9
          movl $119, %edx
          movl $.LC4, %esi
          movl $.LC5, %edi
          {\tt call} \qquad {\tt \_I\_default\_handler}
  .L9:
          .loc 1 120 0 is_stmt 1
• asm(""); with gcc -g -00 produces:
          .loc 1 119 0
• asm(""); with gcc -g -01 produces:
          .loc 1 119 0
• asm(""); with gcc -g -03 produces:
          .loc 1 119 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
```

.LBE2:

```
\bullet asm("nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm(""); with gcc -g -00 produces:
          .loc 1 119 0
• asm(""); with gcc -g -01 produces:
          .loc 1 119 0
• asm(""); with gcc -g -03 produces:
          .loc 1 119 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
 # 0 "" 2
```

• asm("nop"); with gcc -g -01 produces:

```
.loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;"); with gcc -g -03 produces:

```
.loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm(""); with gcc -g -00 produces:
          .loc 1 119 0
• asm(""); with gcc -g -01 produces:
          .loc 1 119 0
• asm(""); with gcc -g -03 produces:
          .loc 1 119 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
```

• asm("nop;nop;"); with gcc -g -01 produces:

```
.loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:

```
.loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -O1 produces:
          .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 119 0
 # 119 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
         .loc 1 119 0
 #NO_APP
         movl $4, -56(%rbp)
          .loc 1 120 0
• i = 4; with gcc -g -01 produces:
         .loc 1 119 0
 #NO_APP
         movl $4, 24(%rsp)
         .loc 1 120 0
• i = 4; with gcc -g -03 produces:
         .loc 1 119 0
 #NO_APP
         movl $4, 24(%rsp)
         .loc 1 120 0
```

• gi = 11; with gcc - g - 00 produces:

.loc 1 119 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 120 0

• gi = 11; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 120 0

• gi = 11; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 120 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl .LC4(%rip), %eax movl %eax, -48(%rbp) .loc 1 120 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

movss .LC4(%rip), %xmm0 movss %xmm0, 28(%rsp) .loc 1 120 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movl \$0x41400000, 28(%rsp)
.loc 1 120 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 120 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, gf(%rip)
.loc 1 120 0

• gf = 12.0; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movl \$0x41400000, gf(%rip)
.loc 1 120 0

• i++; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl -56(%rbp), %eax addl \$1, %eax movl %eax, -56(%rbp) .loc 1 120 0

• i++; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

movl 24(%rsp), %eax addl \$1, %eax movl %eax, 24(%rsp) .loc 1 120 0

• i++; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movl 24(%rsp), %eax addl \$1, %eax movl %eax, 24(%rsp) .loc 1 120 0

• gi++; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 120 0

• gi++; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 120 0

• gi++; with gcc -g -03 produces:

.loc 1 119 0
#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 120 0

• j = a[i]; with gcc -g -00 produces:

.loc 1 119 0

#NO_APP

movl -60(%rbp), %eax
cltq
movl a(,%rax,4), %eax
movl %eax, -56(%rbp)
.loc 1 120 0

• j = a[i]; with gcc -g -01 produces:

.loc 1 119 0

#NO_APP

movl 20(%rsp), %eax
cltq
movl a(,%rax,4), %eax
movl %eax, 24(%rsp)
.loc 1 120 0

• j = a[i]; with gcc -g -03 produces:

.loc 1 119 0

#NO_APP

movslq 20(%rsp), %rax
movl a(,%rax,4), %eax
movl %eax, 24(%rsp)
.loc 1 120 0

• assert(i >= 10); with gcc -g -00 produces:

```
.loc 1 119 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                   .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
                  $119, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L5:
          .loc 1 120 0 is_stmt 1
• assert(i >= 10); with gcc -g -O1 produces:
          .loc 1 119 0
 #NO_APP
          {\tt movl}
                  24(%rsp), %eax
          {\tt cmpl}
                  $9, %eax
                   .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
                  $119, %edx
          movl
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 120 0 is_stmt 1
• assert(i \geq= 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
          jle
                   .L274
  .L5:
          .loc 1 120 0
• BSD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                   .L5
          jg
          .loc 1 119 0 is_stmt 0 discriminator 1
                  $.LC4, %edx
          movl
                  $119, %esi
          movl
```

```
$.LC5, %edi
          movl
          call
                  __BSD_assert
  .L5:
          .loc 1 120 0 is_stmt 1
• BSD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 120 0
• BSD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
                  24(%rsp), %eax
          movl
          .loc 1 120 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  -56(%rbp), %eax
          movl
                  $9, %eax
          cmpl
                  .L5
          jg
  .LBB2:
          .loc 1 119 0 is_stmt 0 discriminator 1
                  stderr(%rip), %rax
          movq
                  $.LC4, %r8d
          movl
                  $119, %ecx
          movl
                  $.LC5, %edx
          movl
          movl
                  $.LC6, %esi
                  %rax, %rdi
          movq
                  $0, %eax
          movl
                  fprintf
          call
                  $1, %edi
          movl
          call
                  exit
  .L5:
  .LBE2:
          .loc 1 120 0 is_stmt 1
• TRAD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
                  .L5
          jg
```

```
.LVL5:
  .LBB783:
  .LBB784:
  .LBB785:
          .loc 2 97 0 discriminator 1
          movl $.LC4, %r9d
                $119, %r8d
          movl
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
                  $1, %esi
          movl
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
                  __fprintf_chk
          call
  .LVL6:
  .LBE785:
  .LBE784:
          .loc 1 119 0 discriminator 1
                  $1, %edi
          movl
          call
                  exit
  .LVL7:
  .L5:
  .LBE783:
          .loc 1 120 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
          jle
                  .L21
          .loc 1 120 0
• I(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          cmpl
                  $9, %eax
          jg
                  .L5
          .loc 1 119 0 is_stmt 0 discriminator 1
                  $119, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L5:
          .loc 1 120 0 is_stmt 1
```

• I(i >= 10); with gcc -g -01 produces:

```
.loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L5
          .loc 1 119 0 is_stmt 0 discriminator 1
                $119, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 120 0 is_stmt 1
• I(i >= 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  24(%rsp), %eax
                  $9, %eax
          cmpl
          jle
                  .L274
  .L5:
          .loc 1 120 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 120 0
• DI(i >= 10); with gcc -g -O1 produces:
          .loc 1 119 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 120 0
• DI(i \ge 10); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
                 $0, _di_target(%rip)
          movl
          .loc 1 120 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
 produces:
```

```
#NO_APP
  .LBB2:
          .loc 1 119 0
                  $1, -3132(%rbp)
          movl
          movq
                  $str, -2096(%rbp)
                  .L5
          jmp
  .L8:
          .loc 1 119 0 is_stmt 0 discriminator 2
          call
                  __ctype_b_loc
                  (\%rax), \%rdx
          movq
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
          addq
                  %rax, %rax
          addq
                  %rdx, %rax
          movzwl (%rax), %eax
          movzwl %ax, %eax
                  $512, %eax
          andl
                  %eax, %eax
          testl
          jne
                  .L6
          .loc 1 119 0 discriminator 1
                  $0, -3132(%rbp)
          movl
                  .L7
          jmp
  .L6:
          .loc 1 119 0 discriminator 2
                  $1, -2096(%rbp)
          addq
  .L5:
          .loc 1 119 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl
                 (%rax), %eax
          testb
                  %al, %al
                  .L8
          jne
  .L7:
                  -3132(%rbp), %eax
          movl
  .LBE2:
          testl
                 %eax, %eax
                  .L9
          jne
                  $119, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L9:
          .loc 1 120 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -01
 produces:
  .LVL5:
 #NO_APP
  .LBB269:
```

```
.loc 1 119 0
          movzbl str(%rip), %ebx
          testb
                  %bl, %bl
          jе
                  .L5
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
  .LVL7:
  .L7:
          .loc 1 119 0 is_stmt 0 discriminator 2
          movsbq %bl, %rbx
          testb
                  $2, 1(%rdx,%rbx,2)
          jе
                  .L6
          addq
                  $1, %rax
  .LVL8:
          movzbl (%rax), %ebx
                  %bl, %bl
          testb
                  .L7
          jne
  .LVL9:
  .L5:
  .LBE269:
          .loc 1 120 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -03
 produces:
  .LVL5:
 #NO_APP
  .LBB271:
          .loc 1 119 0
          movsbq str(%rip), %rbx
                  %bl, %bl
          testb
          jе
                  .L5
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
                  .L7
          jmp
  .LVL7:
          .p2align 4,,10
          .p2align 3
  .L2074:
          .loc 1 119 0 is_stmt 0 discriminator 2
          addq
                $1, %rax
  .LVL8:
          movsbq (%rax), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
  .LVL9:
```

```
.L7:
                  $2, 1(%rdx,%rbx,2)
          testb
          jne
                  .L2074
  .LVL10:
  .LBE271:
          .loc 1 119 0
          movl
                  $119, %edx
                  $.LC11, %esi
          movl
                  $.LC12, %edi
          movl
                  _I_default_handler
          call
  .LVL11:
  .L5:
          .loc 1 120 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -00 produces:
 #NO_APP
  .LBB2:
          .loc 1 119 0
          movl
                  $1, -5188(%rbp)
                  $0, -5184(%rbp)
          movl
                  .L5
          jmp
  .L11:
  .LBB3:
          .loc 1 119 0 is_stmt 0 discriminator 2
                  $0, -2096(%rbp)
          movq
                  $0, -5180(%rbp)
          movl
                  .L6
          jmp
  .L8:
                  -5184(%rbp), %eax
          movl
          cltq
                  a(, %rax, 4), %edx
          movl
          movl
                  -5180(%rbp), %eax
          cltq
          movl
                  a(, %rax, 4), %eax
                  %eax, %edx
          cmpl
          jne
                  .L7
          .loc 1 119 0 discriminator 1
                  $1, -2096(%rbp)
          addq
  .L7:
          .loc 1 119 0 discriminator 2
                  $1, -5180(%rbp)
          addl
  .L6:
          .loc 1 119 0 discriminator 1
          cmpl
                  $9, -5180(%rbp)
          jle
                  .L8
          .loc 1 119 0 discriminator 3
                  -2096(%rbp), %rax
          movq
  .LBE3:
```

```
$1, %rax
          cmpq
                  .L9
          jе
          .loc 1 119 0 discriminator 1
          movl
                  $0, -5188(%rbp)
                  .L10
          jmp
  .L9:
          .loc 1 119 0 discriminator 2
          addl
                  $1, -5184(%rbp)
  .L5:
          .loc 1 119 0 discriminator 1
                  $9, -5184(%rbp)
          cmpl
          jle
                  .L11
  .L10:
          movl
                  -5188(%rbp), %eax
  .LBE2:
                  %eax, %eax
          testl
                  .L12
          jne
                  $119, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
                  _I_default_handler
          call
  .L12:
          .loc 1 120 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
          .loc 1 119 0
                 $0, %ecx
          movl
  .LBE526:
          .loc 1 118 0
          movl
                  $0, %eax
          movl
                  $0, %edx
                  .L5
          jmp
  .LVL6:
  .L1295:
  .LBB528:
                  $0, %eax
          movl
  .LVL7:
                  $0, %edx
          movl
  .LVL8:
  .L5:
  .LBB527:
          .loc 1 119 0 discriminator 2
          movslq %ecx, %rdi
          movslq %eax, %rsi
                  a(,%rsi,4), %ebx
          movl
```

```
%ebx, a(,%rdi,4)
          cmpl
                  %sil
          sete
          movzbl %sil, %esi
          addq
                  %rsi, %rdx
  .LVL9:
          addl
                  $1, %eax
  .LVL10:
                  $9, %eax
          cmpl
          jle
                  .L5
  .LBE527:
          .loc 1 119 0 is_stmt 0 discriminator 3
                  $1, %rdx
          cmpq
                  .L8
          jne
          .loc 1 119 0 discriminator 2
          addl
                  $1, %ecx
  .LVL11:
                  $9, %ecx
          cmpl
                  .L1295
          jle
  .LVL12:
  .L1038:
  .LBE528:
          .loc 1 120 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -03 produces:
  .LVL5:
 #NO_APP
  .LBB528:
  .LBB529:
          .loc 1 119 0
                  a(%rip), %r13d
          movl
                  a+4(%rip), %r12d
          movl
                  $a, %ecx
          movl
          movl
                  a+8(%rip), %ebp
                  a+12(%rip), %ebx
          movl
                  a+16(%rip), %r11d
          movl
                  a+20(%rip), %r10d
          movl
                  a+24(%rip), %r9d
          movl
                  a+28(%rip), %r8d
          movl
                  a+32(%rip), %edi
          movl
                  a+36(%rip), %esi
          movl
  .LVL6:
  .L5:
          movl
                  (%rcx), %edx
          xorl
                  %eax, %eax
          {\tt cmpl}
                  %r13d, %edx
                  %al
          sete
  .LVL7:
          xorl
                  %r14d, %r14d
```

```
%r12d, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
.LVL8:
                %r14d, %r14d
       xorl
                %ebp, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
.LVL9:
                %r14d, %r14d
        xorl
                %ebx, %edx
        cmpl
        sete
                %r14b
                %r14, %rax
        addq
.LVL10:
        xorl
                %r14d, %r14d
                %r11d, %edx
        cmpl
        sete
                %r14b
                %r14, %rax
        addq
.LVL11:
        xorl
                %r14d, %r14d
                %r10d, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
.LVL12:
        xorl
                %r14d, %r14d
                %r9d, %edx
        cmpl
        sete
                %r14b
                %r14, %rax
        addq
.LVL13:
                %r14d, %r14d
        xorl
        cmpl
                %r8d, %edx
                %r14b
        sete
                %r14, %rax
        addq
.LVL14:
        xorl
                %r14d, %r14d
                %edi, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
.LVL15:
                %esi, %edx
        cmpl
        sete
                %dl
                %dl, %edx
       movzbl
                %rdx, %rax
        addq
.LVL16:
.LBE529:
        cmpq
                $1, %rax
        jne
                .L4164
                $4, %rcx
        addq
        .loc 1 119 0 is_stmt 0 discriminator 2
        cmpq
                $a+40, %rcx
```

```
jne
                 .L5
  .LVL17:
  .L6:
  .LBE528:
          .loc 1 120 0 is_stmt 1
• printf("helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  $.LC4, %edi
         movl
                  $0, %eax
         movl
          call
                  printf
          .loc 1 120 0
• printf("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
                $.LC4, %esi
         movl
                  $1, %edi
         movl
                $0, %eax
         movl
          call
                  __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• printf("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 104 0
         movl $.LC4, %esi
                 $1, %edi
         movl
                 %eax, %eax
         xorl
         call
                __printf_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 120 0
```

 \bullet L("helloworldn"); with gcc -g -00 produces:

```
.loc 1 119 0
 #NO_APP
               stdout(%rip), %rax
         movq
         movq %rax, %rcx
         movl
               $11, %edx
              $1, %esi
         movl
               $.LC4, %edi
         movl
         call fwrite
         .loc 1 120 0
• L("helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
         .loc 2 97 0
         movq stdout(%rip), %rcx
               $11, %edx
         movl
         movl
              $1, %esi
         movl $.LC4, %edi
         call fwrite
  .LVL6:
  .LBE527:
  .LBE526:
         .loc 1 120 0
• L("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 2 97 0
         movq stdout(%rip), %rcx
         movl $11, %edx
         movl $1, %esi
         movl $.LC4, %edi
         call
                fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
         .loc 1 120 0
• DL("helloworldn"); with gcc -g -00 produces:
         .loc 1 119 0
 #NO_APP
                 $0, _dl_target(%rip)
         movl
         .loc 1 120 0
```

```
• DL("helloworldn"); with gcc -g -O1 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $0, _dl_target(%rip)
          .loc 1 120 0
• DL("helloworldn"); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $0, _dl_target(%rip)
          .loc 1 120 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
          movl
                  $0, gi(%rip)
                  gi(%rip), %eax
          movl
                  $16, %eax
          andl
          testl %eax, %eax
          jе
                  .L5
          .loc 1 119 0 is_stmt 0 discriminator 1
          movq stdout(%rip), %rax
                 %rax, %rcx
          movq
          movl
                 $11, %edx
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 120 0 is_stmt 1
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O1 produces:
          .loc 1 119 0
 #NO_APP
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
          movl
                  $16, %al
          testb
          jе
                  .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
```

```
call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
                  $0, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
                  $16, %al
          testb
                  .L1045
          jne
  .L5:
          .loc 1 120 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
          andl
                  $16, %eax
                  %eax, %eax
          testl
          jе
                   .L5
          .loc 1 119 0 is_stmt 0 discriminator 1 \,
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 120 0 is_stmt 1
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 119 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
          jе
                   .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
```

```
stdout(%rip), %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
          movl
                  $.LC4, %edi
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 120 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
                  $-1, gi(%rip)
          movl
                  gi(%rip), %eax
          movl
                  $16, %al
          testb
          jne
                  .L1045
  .L5:
          .loc 1 120 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -00 produces:
          .loc 1 119 0
 #NO_APP
                  -32(%rbp), %rax
          movq
                  %rax, %rcx
          movq
                  $11, %edx
          movl
          movl
                  $1, %esi
                  $.LC4, %edi
          movl
          call
                  fwrite
          .loc 1 120 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
                %rbx, %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
          movl
                  $.LC4, %edi
          call
                  fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0
```

```
• LHP(fprintf,log,"helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 97 0
          movl $11, %edx
          movq
                  %rbx, %rcx
                $1, %esi
          movl
                $.LC4, %edi
          movl
          call
                  fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 120 0
• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -OO produces:
          .loc 1 119 0
 #NO_APP
                  -40(%rbp), %rax
          movq
          movl
                  $.LC4, %esi
                  %rax, %rdi
          movq
                  $0, %eax
          movl
                  L_buffer_printf
          call
          .loc 1 120 0
\bullet LHP(L_buffer_printf,buf,"helloworldn"); with gcc \, -g \, -O1 produces:
          .loc 1 119 0
 #NO_APP
                  $.LC4, %esi
          movl
          movq
                  %rbx, %rdi
                  $0, %eax
          movl
                  L_buffer_printf
          call
  .LVL6:
          .loc 1 120 0
• LHP(L_buffer_printf,buf, "helloworldn"); with gcc -g -03 produces:
          .loc 1 119 0
 #NO_APP
                  $.LC4, %esi
          movl
          movq
                  %rbx, %rdi
          xorl
                  %eax, %eax
                  L_buffer_printf
          call
  .LVL279:
          .loc 1 120 0
```

• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -OO produces: .loc 1 119 0 #NO_APP \$.LC4, %esi movl \$8, %edi movl **\$0,** %eax movl call syslog .loc 1 120 0 \bullet LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces: .LVL5: #NO_APP .LBB526: .LBB527: .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h" .loc 3 31 0 \$.LC4, %edx movl movl \$1, %esi movl \$8, %edi movl \$0, %eax call __syslog_chk .LVL6: .LBE527: .LBE526: .loc 1 120 0 • LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O3 produces: .LVL277: #NO_APP .LBB1050: .LBB1051: .loc 3 31 0 movl \$.LC4, %edx movl \$1, %esi \$8, %edi movl xorl %eax, %eax call __syslog_chk .LVL278: .LBE1051: .LBE1050: .loc 1 120 0

• asm(""); with gcc -g -00 produces:

.loc 1 120 0

```
.loc 1 120 0
• asm(""); with gcc -g -03 produces:
          .loc 1 120 0
• asm("nop"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
```

 \bullet asm(""); with gcc -g -01 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
        nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;"); with gcc -g -00 produces:

```
• asm("nop;nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl
               $4, -56(%rbp)
          .loc 1 121 0
• i = 4; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
         movl
               $4, 24(%rsp)
         .loc 1 121 0
• i = 4; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
         movl $4, 24(%rsp)
         .loc 1 121 0
• gi = 11; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl $11, gi(%rip)
          .loc 1 121 0
• gi = 11; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
```

\$11, gi(%rip)

movl

.loc 1 121 0

• gi = 11; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, -48(%rbp)
.loc 1 121 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0 movss %xmm0, 28(%rsp) .loc 1 121 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, 28(%rsp) .loc 1 121 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
                  $0x41400000, gf(%rip)
          movl
          .loc 1 121 0
• i++; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          addl
                  $1, %eax
                  %eax, -56(%rbp)
          .loc 1 121 0
• i++; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          addl
                  $1, %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• i++; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          addl
                  $1, %eax
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• gi++; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  gi(%rip), %eax
          addl
                  $1, %eax
                  %eax, gi(%rip)
          movl
          .loc 1 121 0
• gi++; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  gi(%rip), %eax
```

\$1, %eax

%eax, gi(%rip)

addl

movl

.loc 1 121 0

```
• gi++; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  gi(%rip), %eax
          movl
          addl
                  $1, %eax
          movl
                  %eax, gi(%rip)
          .loc 1 121 0
• j = a[i]; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -60(%rbp), %eax
          movl
          cltq
          movl
                  a(, %rax, 4), %eax
                  %eax, -56(%rbp)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  20(%rsp), %eax
          movl
          cltq
                  a(, %rax, 4), %eax
          movl
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movslq 20(%rsp), %rax
                  a(, %rax, 4), %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          cmpl
                  $9, %eax
                   .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
          .loc 1 121 0 is_stmt 1
```

• assert(i >= 10); with gcc -g -O1 produces: .loc 1 120 0 #NO_APP 24(%rsp), %eax movl \$9, %eax cmpljg .L5 .loc 1 120 0 is_stmt 0 discriminator 1 \$120, %edx movl \$.LC4, %esi movl \$.LC5, %edi movl call _I_default_handler .LVL5: .L5: .loc 1 121 0 is_stmt 1 • assert(i >= 10); with gcc -g -03 produces: .loc 1 120 0 #NO_APP movl 24(%rsp), %eax cmpl\$9, %eax jle .L274 .L5: .loc 1 121 0 • BSD_assert(i >= 10); with gcc -g -00 produces: .loc 1 120 0 #NO_APP -56(%rbp), %eax movl cmpl \$9, %eax .L5 jg .loc 1 120 0 is_stmt 0 discriminator 1 movl \$.LC4, %edx movl \$120, %esi \$.LC5, %edi movl call __BSD_assert .L5: .loc 1 121 0 is_stmt 1 • BSD_assert(i >= 10); with gcc -g -01 produces: .loc 1 120 0 #NO_APP movl24(%rsp), %eax .loc 1 121 0

• BSD_assert(i >= 10); with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movq
                  stderr(%rip), %rax
          movl
                  $.LC4, %r8d
                  $120, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
                  %rax, %rdi
          movq
                  $0, %eax
          movl
          call
                  fprintf
                  $1, %edi
          movl
          call
                  exit
  .L5:
          .loc 1 121 0 is_stmt 1
• TRAD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
                  .L5
          jg
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                 $.LC4, %r9d
          movl
          movl
                  $120, %r8d
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
                  $1, %esi
          movl
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
          call
                  __fprintf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
```

```
$1, %edi
          movl
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
                  $9, %eax
          cmpl
          jle
                  .L21
          .loc 1 121 0
• I(i \geq 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
          .loc 1 121 0 is_stmt 1
• I(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
```

• I(i \geq 10); with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 121 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
• DI(i \ge 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• DI(i \ge 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
 produces:
 #NO_APP
  .LBB2:
          .loc 1 120 0
          movl $1, -3132(%rbp)
                  $str, -2096(%rbp)
          movq
          jmp
                  .L5
  .L8:
          .loc 1 120 0 is_stmt 0 discriminator 2
                  __ctype_b_loc
          call
                  (\%rax), \%rdx
          movq
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
          addq
                  %rax, %rax
                  %rdx, %rax
          addq
          movzwl (%rax), %eax
          movzwl %ax, %eax
```

```
$512, %eax
          andl
                  %eax, %eax
          testl
          jne
                  .L6
          .loc 1 120 0 discriminator 1
          movl
                  $0, -3132(%rbp)
                  .L7
          jmp
  .L6:
          .loc 1 120 0 discriminator 2
                  $1, -2096(%rbp)
          addq
  .L5:
          .loc 1 120 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl
                  (%rax), %eax
          testb
                  %al, %al
          jne
                  .L8
  .L7:
                  -3132(%rbp), %eax
          movl
  .LBE2:
                  %eax, %eax
          testl
          jne
                  .L9
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L9:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -01
 produces:
  .LVL5:
 #NO_APP
  .LBB269:
          .loc 1 120 0
          movzbl str(%rip), %ebx
          testb
                  %bl, %bl
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
                  $str, %eax
          movl
  .LVL7:
  .L7:
          .loc 1 120 0 is_stmt 0 discriminator 2
          movsbq %bl, %rbx
          testb
                  $2, 1(%rdx,%rbx,2)
          jе
                  .L6
                  $1, %rax
          addq
  .LVL8:
          movzbl (%rax), %ebx
```

```
jne
                  .L7
  .LVL9:
  .L5:
  .LBE269:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -03
 produces:
  .LVL5:
 #NO_APP
  .LBB271:
          .loc 1 120 0
          movsbq str(%rip), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
                  .L7
          jmp
  .LVL7:
          .p2align 4,,10
          .p2align 3
  .L2074:
          .loc 1 120 0 is_stmt 0 discriminator 2
          addq
                  $1, %rax
  .LVL8:
          movsbq (%rax), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
  .LVL9:
  .L7:
                  $2, 1(%rdx,%rbx,2)
          testb
          jne
                  .L2074
  .LVL10:
  .LBE271:
          .loc 1 120 0
          movl
                  $120, %edx
                  $.LC11, %esi
          movl
                  $.LC12, %edi
          movl
                  _I_default_handler
          call
  .LVL11:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -00 produces:
```

%bl, %bl

testb

```
#NO_APP
.LBB2:
        .loc 1 120 0
        movl
                $1, -5188(%rbp)
        movl
                $0, -5184(%rbp)
                 .L5
        jmp
.L11:
.LBB3:
        .loc 1 120 0 is_stmt 0 discriminator 2
                $0, -2096(%rbp)
        movq
                $0, -5180(%rbp)
        movl
                 .L6
        jmp
.L8:
        movl
                -5184(%rbp), %eax
        cltq
                a(, %rax, 4), %edx
        movl
                -5180(%rbp), %eax
        movl
        cltq
                a(, %rax, 4), %eax
        movl
        cmpl
                %eax, %edx
                 .L7
        jne
        .loc 1 120 0 discriminator 1
        addq
                $1, -2096(%rbp)
.L7:
        .loc 1 120 0 discriminator 2
        addl
                $1, -5180(%rbp)
.L6:
        .loc 1 120 0 discriminator 1
                $9, -5180(%rbp)
        cmpl
        jle
                 .L8
        .loc 1 120 0 discriminator 3
                -2096(%rbp), %rax
        movq
.LBE3:
                $1, %rax
        cmpq
        jе
                 .L9
        .loc 1 120 0 discriminator 1
                $0, -5188(%rbp)
        movl
        jmp
                 .L10
.L9:
        .loc 1 120 0 discriminator 2
        addl
                $1, -5184(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
                $9, -5184(%rbp)
        cmpl
        jle
                .L11
.L10:
        movl
                -5188(%rbp), %eax
.LBE2:
                %eax, %eax
        testl
        jne
                 .L12
```

```
$120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L12:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
          .loc 1 120 0
               $0, %ecx
  .LBE526:
          .loc 1 119 0
                  $0, %eax
          movl
                  $0, %edx
          movl
          jmp
                  .L5
  .LVL6:
  .L1295:
  .LBB528:
          movl
                  $0, %eax
  .LVL7:
                  $0, %edx
          movl
  .LVL8:
  .L5:
  .LBB527:
          .loc 1 120 0 discriminator 2
          movslq %ecx, %rdi
          movslq %eax, %rsi
                  a(, %rsi, 4), %ebx
          movl
          cmpl
                  %ebx, a(, %rdi, 4)
          sete
                  %sil
          movzbl %sil, %esi
                  %rsi, %rdx
          addq
  .LVL9:
          addl
                  $1, %eax
  .LVL10:
          cmpl
                  $9, %eax
          jle
                  .L5
  .LBE527:
          .loc 1 120 0 is_stmt 0 discriminator 3
          cmpq
                  $1, %rdx
          jne
                  .L8
          .loc 1 120 0 discriminator 2
                  $1, %ecx
          addl
  .LVL11:
          cmpl
                  $9, %ecx
```

```
jle .L1295
  .LVL12:
  .L1038:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• asm(""); with gcc -g -00 produces:
          .loc 1 120 0
• asm(""); with gcc -g -01 produces:
          .loc 1 120 0
• asm(""); with gcc -g -03 produces:
          .loc 1 120 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop
 # 0 "" 2
\bullet asm("nop"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
```

• asm("nop;nop;"); with gcc -g -01 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -O1 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
         .loc 1 120 0
 #NO_APP
         movl $4, -56(%rbp)
          .loc 1 121 0
• i = 4; with gcc -g -01 produces:
         .loc 1 120 0
 #NO_APP
         movl $4, 24(%rsp)
         .loc 1 121 0
• i = 4; with gcc -g -03 produces:
         .loc 1 120 0
 #NO_APP
         movl $4, 24(%rsp)
         .loc 1 121 0
```

• gi = 11; with gcc - g - 00 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• gi = 11; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• gi = 11; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, -48(%rbp)
.loc 1 121 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0 movss %xmm0, 28(%rsp) .loc 1 121 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, 28(%rsp) .loc 1 121 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, gf(%rip) .loc 1 121 0

• i++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl -56(%rbp), %eax addl \$1, %eax movl %eax, -56(%rbp) .loc 1 121 0

• i++; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl 24(%rsp), %eax addl \$1, %eax movl %eax, 24(%rsp) .loc 1 121 0

• i++; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl 24(%rsp), %eax addl \$1, %eax movl %eax, 24(%rsp) .loc 1 121 0

• gi++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• gi++; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• gi++; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• j = a[i]; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl -60(%rbp), %eax
cltq
movl a(,%rax,4), %eax
movl %eax, -56(%rbp)
.loc 1 121 0

• j = a[i]; with gcc -g -01 produces:

• j = a[i]; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movslq 20(%rsp), %rax
movl a(,%rax,4), %eax
movl %eax, 24(%rsp)
.loc 1 121 0

• assert(i >= 10); with gcc -g -00 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                   .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L5:
          .loc 1 121 0 is_stmt 1
• assert(i >= 10); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          {\tt cmpl}
                  $9, %eax
                   .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
          jle
                  .L274
  .L5:
          .loc 1 121 0
• BSD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $.LC4, %edx
          movl
          movl
                  $120, %esi
```

```
$.LC5, %edi
          call
                  __BSD_assert
  .L5:
          .loc 1 121 0 is_stmt 1
• BSD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 121 0
• BSD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          cmpl
                  $9, %eax
                   .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1 \,
                  stderr(%rip), %rax
          movq
                  $.LC4, %r8d
          movl
                  $120, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
          movq
                  %rax, %rdi
                  $0, %eax
          movl
                  {\tt fprintf}
          call
                  $1, %edi
          movl
          call
                  exit
  .L5:
          .loc 1 121 0 is_stmt 1
• TRAD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          {\tt cmpl}
                  $9, %eax
                   .L5
          jg
  .LVL5:
  .LBB526:
```

movl

```
.LBB527:
          .loc 2 97 0 discriminator 1
                 $.LC4, %r9d
          movl
          movl
                  $120, %r8d
                $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
          movl
               $1, %esi
          movq stderr(%rip), %rdi
                  $0, %eax
          movl
          call
                  __fprintf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
                $1, %edi
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
          jle
                  .L21
          .loc 1 121 0
• I(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
          jg
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L5:
          .loc 1 121 0 is_stmt 1
• asm(""); with gcc -g -00 produces:
          .loc 1 120 0
```

• asm(""); with gcc -g -01 produces:

.loc 1 120 0

• asm(""); with gcc -g -03 produces:

.loc 1 120 0

• asm("nop"); with gcc -g -00 produces:

• asm("nop"); with gcc -g -01 produces:

• asm("nop"); with gcc -g -03 produces:

 \bullet asm("nop;nop;"); with gcc -g -00 produces:

.loc 1 120 0 # 120 "tmp.c" 1 nop;nop; # 0 "" 2

• asm("nop;nop;"); with gcc -g -01 produces:

.loc 1 120 0 # 120 "tmp.c" 1 nop;nop; # 0 "" 2

• asm("nop;nop;"); with gcc -g -03 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;"); with gcc -g -00 produces:

```
• asm("nop;nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl
               $4, -56(%rbp)
          .loc 1 121 0
• i = 4; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
         movl
               $4, 24(%rsp)
         .loc 1 121 0
• i = 4; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
         movl $4, 24(%rsp)
         .loc 1 121 0
• gi = 11; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl $11, gi(%rip)
          .loc 1 121 0
• gi = 11; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
```

\$11, gi(%rip)

movl

.loc 1 121 0

• gi = 11; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, -48(%rbp)
.loc 1 121 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0 movss %xmm0, 28(%rsp) .loc 1 121 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, 28(%rsp) .loc 1 121 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0 movss %xmm0, gf(%rip) .loc 1 121 0

• gf = 12.0; with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
                  $0x41400000, gf(%rip)
          movl
          .loc 1 121 0
• i++; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          addl
                  $1, %eax
                  %eax, -56(%rbp)
          .loc 1 121 0
• i++; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          addl
                  $1, %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• i++; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          addl
                  $1, %eax
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• gi++; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  gi(%rip), %eax
          addl
                  $1, %eax
                  %eax, gi(%rip)
          movl
          .loc 1 121 0
• gi++; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  gi(%rip), %eax
                  $1, %eax
          addl
```

%eax, gi(%rip)

movl

.loc 1 121 0

```
• gi++; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  gi(%rip), %eax
          movl
          addl
                  $1, %eax
          movl
                  %eax, gi(%rip)
          .loc 1 121 0
• j = a[i]; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -60(%rbp), %eax
          movl
          cltq
          movl
                  a(, %rax, 4), %eax
                  %eax, -56(%rbp)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  20(%rsp), %eax
          movl
          cltq
                  a(, %rax, 4), %eax
          movl
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movslq 20(%rsp), %rax
                  a(, %rax, 4), %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          cmpl
                  $9, %eax
                   .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
          .loc 1 121 0 is_stmt 1
```

.loc 1 120 0 #NO_APP 24(%rsp), %eax movl \$9, %eax cmpljg .L5 .loc 1 120 0 is_stmt 0 discriminator 1 \$120, %edx movl \$.LC4, %esi movl \$.LC5, %edi movl call _I_default_handler .LVL5: .L5: .loc 1 121 0 is_stmt 1 • assert(i >= 10); with gcc -g -03 produces: .loc 1 120 0 #NO_APP movl 24(%rsp), %eax cmpl\$9, %eax jle .L274 .L5: .loc 1 121 0 • BSD_assert(i >= 10); with gcc -g -00 produces: .loc 1 120 0 #NO_APP -56(%rbp), %eax movl cmpl \$9, %eax .L5 jg .loc 1 120 0 is_stmt 0 discriminator 1 movl \$.LC4, %edx movl \$120, %esi \$.LC5, %edi movl call __BSD_assert .L5: .loc 1 121 0 is_stmt 1 • BSD_assert(i >= 10); with gcc -g -01 produces: .loc 1 120 0 #NO_APP movl24(%rsp), %eax .loc 1 121 0

• assert(i >= 10); with gcc -g -O1 produces:

• BSD_assert(i >= 10); with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movq
                  stderr(%rip), %rax
          movl
                  $.LC4, %r8d
                  $120, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
                  %rax, %rdi
          movq
                  $0, %eax
          movl
          call
                  fprintf
                  $1, %edi
          movl
          call
                  exit
  .L5:
          .loc 1 121 0 is_stmt 1
• TRAD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
                  .L5
          jg
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                 $.LC4, %r9d
          movl
          movl
                  $120, %r8d
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
                  $1, %esi
          movl
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
          call
                  __fprintf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
```

```
$1, %edi
          movl
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
                  $9, %eax
          cmpl
          jle
                  .L21
          .loc 1 121 0
• I(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
          .loc 1 121 0 is_stmt 1
• I(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
```

• I(i \geq 10); with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 121 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
• DI(i \ge 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• DI(i \ge 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
 produces:
 #NO_APP
  .LBB2:
          .loc 1 120 0
          movl $1, -3132(%rbp)
                  $str, -2096(%rbp)
          movq
          jmp
                  .L5
  .L8:
          .loc 1 120 0 is_stmt 0 discriminator 2
                  __ctype_b_loc
          call
                  (\%rax), \%rdx
          movq
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
          addq
                  %rax, %rax
          addq
                  %rdx, %rax
          movzwl (%rax), %eax
          movzwl %ax, %eax
```

```
$512, %eax
          andl
                  %eax, %eax
          testl
          jne
                  .L6
          .loc 1 120 0 discriminator 1
          movl
                  $0, -3132(%rbp)
                  .L7
          jmp
  .L6:
          .loc 1 120 0 discriminator 2
                  $1, -2096(%rbp)
          addq
  .L5:
          .loc 1 120 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl
                  (%rax), %eax
          testb
                  %al, %al
          jne
                  .L8
  .L7:
                  -3132(%rbp), %eax
          movl
  .LBE2:
                  %eax, %eax
          testl
          jne
                  .L9
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L9:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -01
 produces:
  .LVL5:
 #NO_APP
  .LBB269:
          .loc 1 120 0
          movzbl str(%rip), %ebx
          testb
                  %bl, %bl
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
                  $str, %eax
          movl
  .LVL7:
  .L7:
          .loc 1 120 0 is_stmt 0 discriminator 2
          movsbq %bl, %rbx
          testb
                  $2, 1(%rdx,%rbx,2)
          jе
                  .L6
                  $1, %rax
          addq
  .LVL8:
          movzbl (%rax), %ebx
```

```
%bl, %bl
          testb
                  .L7
          jne
  .LVL9:
  .L5:
  .LBE269:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -03
 produces:
  .LVL5:
 #NO_APP
  .LBB271:
          .loc 1 120 0
          movsbq str(%rip), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
                  .L7
          jmp
  .LVL7:
          .p2align 4,,10
          .p2align 3
  .L2074:
          .loc 1 120 0 is_stmt 0 discriminator 2
          addq
                  $1, %rax
  .LVL8:
          movsbq (%rax), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
  .LVL9:
  .L7:
                  $2, 1(%rdx,%rbx,2)
          testb
          jne
                  .L2074
  .LVL10:
  .LBE271:
          .loc 1 120 0
          movl
                  $120, %edx
                  $.LC10, %esi
          movl
                  $.LC11, %edi
          movl
                  _I_default_handler
          call
  .LVL11:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -00 produces:
```

```
#NO_APP
.LBB2:
        .loc 1 120 0
        movl
                $1, -5188(%rbp)
        movl
                $0, -5184(%rbp)
                 .L5
        jmp
.L11:
.LBB3:
        .loc 1 120 0 is_stmt 0 discriminator 2
                $0, -2096(%rbp)
        movq
                $0, -5180(%rbp)
        movl
                 .L6
        jmp
.L8:
        movl
                -5184(%rbp), %eax
        cltq
                a(, %rax, 4), %edx
        movl
                -5180(%rbp), %eax
        movl
        cltq
                a(, %rax, 4), %eax
        movl
        cmpl
                %eax, %edx
        jne
                 .L7
        .loc 1 120 0 discriminator 1
        addq
                $1, -2096(%rbp)
.L7:
        .loc 1 120 0 discriminator 2
        addl
                $1, -5180(%rbp)
.L6:
        .loc 1 120 0 discriminator 1
                $9, -5180(%rbp)
        cmpl
        jle
                 .L8
        .loc 1 120 0 discriminator 3
                -2096(%rbp), %rax
        movq
.LBE3:
                $1, %rax
        cmpq
        jе
                 .L9
        .loc 1 120 0 discriminator 1
                $0, -5188(%rbp)
        movl
        jmp
                 .L10
.L9:
        .loc 1 120 0 discriminator 2
        addl
                $1, -5184(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
                $9, -5184(%rbp)
        cmpl
        jle
                .L11
.L10:
        movl
                -5188(%rbp), %eax
.LBE2:
                %eax, %eax
        testl
        jne
                 .L12
```

```
$120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L12:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
          .loc 1 120 0
               $0, %ecx
  .LBE526:
          .loc 1 119 0
                  $0, %eax
          movl
                  $0, %edx
          movl
          jmp
                  .L5
  .LVL6:
  .L1295:
  .LBB528:
          movl
                  $0, %eax
  .LVL7:
                  $0, %edx
          movl
  .LVL8:
  .L5:
  .LBB527:
          .loc 1 120 0 discriminator 2
          movslq %ecx, %rdi
          movslq %eax, %rsi
                  a(, %rsi, 4), %ebx
          movl
          cmpl
                  %ebx, a(, %rdi, 4)
          sete
                  %sil
          movzbl %sil, %esi
                  %rsi, %rdx
          addq
  .LVL9:
          addl
                  $1, %eax
  .LVL10:
          cmpl
                  $9, %eax
          jle
                  .L5
  .LBE527:
          .loc 1 120 0 is_stmt 0 discriminator 3
          cmpq
                  $1, %rdx
          jne
                  .L8
          .loc 1 120 0 discriminator 2
                  $1, %ecx
          addl
  .LVL11:
          cmpl
                  $9, %ecx
```

```
.L1295
          jle
  .LVL12:
  .L1038:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -03 produces:
  .LVL5:
 #NO_APP
  .LBB528:
  .LBB529:
          .loc 1 120 0
          movl
                  a(%rip), %r13d
                  a+4(%rip), %r12d
          movl
                  $a, %ecx
          movl
                  a+8(%rip), %ebp
          movl
                  a+12(%rip), %ebx
          movl
          movl
                  a+16(%rip), %r11d
                  a+20(%rip), %r10d
          movl
                  a+24(%rip), %r9d
          movl
                  a+28(%rip), %r8d
          movl
                  a+32(%rip), %edi
          movl
          movl
                  a+36(%rip), %esi
  .LVL6:
  .L5:
                  (%rcx), %edx
          movl
                  %eax, %eax
          xorl
          cmpl
                  %r13d, %edx
          sete
                  %al
  .LVL7:
                  %r14d, %r14d
          xorl
                  %r12d, %edx
          cmpl
          sete
                  %r14b
          addq
                  %r14, %rax
  .LVL8:
                  %r14d, %r14d
          xorl
                  %ebp, %edx
          cmpl
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL9:
                  %r14d, %r14d
          xorl
                  %ebx, %edx
          cmpl
          sete
                  %r14b
          addq
                  %r14, %rax
  .LVL10:
                  %r14d, %r14d
          xorl
                  %r11d, %edx
          cmpl
```

sete

%r14b

```
%r14, %rax
          addq
  .LVL11:
                  %r14d, %r14d
          xorl
          cmpl
                  %r10d, %edx
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL12:
                  %r14d, %r14d
          xorl
                  %r9d, %edx
          cmpl
                  %r14b
          sete
                  %r14, %rax
          addq
  .LVL13:
                  %r14d, %r14d
          xorl
          cmpl
                  %r8d, %edx
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL14:
                  %r14d, %r14d
          xorl
                  %edi, %edx
          cmpl
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL15:
          cmpl
                  %esi, %edx
                  %dl
          sete
          movzbl %dl, %edx
          addq
                  %rdx, %rax
  .LVL16:
  .LBE529:
                  $1, %rax
          cmpq
          jne
                  .L4164
          addq
                  $4, %rcx
          .loc 1 120 0 is_stmt 0 discriminator 2 \,
                  $a+40, %rcx
          cmpq
          jne
                   .L5
  .LVL17:
  .L6:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• printf("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $.LC4, %edi
          movl
          movl
                  $0, %eax
```

• printf("helloworldn"); with gcc -g -01 produces:

printf

call

.loc 1 121 0

```
.LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
         movl $.LC4, %esi
         movl $1, %edi
         movl
               $0, %eax
         call
                __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 104 0
         movl $.LC4, %esi
                 $1, %edi
         movl
         xorl %eax, %eax
          call
                 __printf_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                 stdout(%rip), %rax
         movq
                 %rax, %rcx
         movq
                 $11, %edx
         movl
                 $1, %esi
         movl
         movl
                 $.LC4, %edi
          call
                 fwrite
          .loc 1 121 0
• L("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
```

```
stdout(%rip), %rcx
         movq
                 $11, %edx
         movl
                 $1, %esi
         movl
         movl
               $.LC4, %edi
         call fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 2 97 0
         movq stdout(%rip), %rcx
                $11, %edx
         movl
         movl $1, %esi
         movl
               $.LC4, %edi
         call
                 fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
         .loc 1 121 0
• DL("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl $0, _dl_target(%rip)
         .loc 1 121 0
• DL("helloworldn"); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
         movl
                $0, _dl_target(%rip)
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                 $0, _dl_target(%rip)
         .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
```

```
.loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
          andl
                  $16, %eax
                  %eax, %eax
          testl
          jе
          .loc 1 120 0 is_stmt 0 discriminator 1
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  $0, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
                  $16, %al
          testb
                  .L5
          jе
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
                  .L1045
          jne
  .L5:
          .loc 1 121 0
```

```
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $-1, gi(%rip)
                  gi(%rip), %eax
          movl
                  $16, %eax
          andl
                  %eax, %eax
          testl
                   .L5
          jе
          .loc 1 120 0 is_stmt 0 discriminator 1 \,
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  $-1, gi(%rip)
          movl
                  gi(%rip), %eax
          movl
                  $16, %al
          testb
          jе
                   .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
                  fwrite
          call
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $-1, gi(%rip)
          movl
                  gi(%rip), %eax
                  $16, %al
```

testb

```
.L1045
          jne
  .L5:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -OO produces:
          .loc 1 120 0
 #NO_APP
                  -32(\%rbp), \%rax
          movq
          movq
                  %rax, %rcx
          movl
                  $11, %edx
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
          movq %rbx, %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 97 0
                 $11, %edx
          movl
                  %rbx, %rcx
          movq
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
```

• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -00 produces:

```
.loc 1 120 0

#NO_APP

movq -40(%rbp), %rax
movl $.LC4, %esi
movq %rax, %rdi
movl $0, %eax
call L_buffer_printf
.loc 1 121 0
```

• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```
.loc 1 120 0

#NO_APP

movl $.LC4, %esi
movq %rbx, %rdi
movl $0, %eax
call L_buffer_printf
.LVL6:
.loc 1 121 0
```

• LHP(L_buffer_printf,buf, "helloworldn"); with gcc -g -03 produces:

```
.loc 1 120 0

#NO_APP

movl $.LC4, %esi
movq %rbx, %rdi
xorl %eax, %eax
call L_buffer_printf
.LVL279:
.loc 1 121 0
```

• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -OO produces:

```
.loc 1 120 0
#NO_APP

movl $.LC4, %esi
movl $8, %edi
movl $0, %eax
call syslog
.loc 1 121 0
```

• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:

```
.LVL5:
#NO_APP
.LBB526:
.LBB527:
```

```
.file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
          .loc 3 31 0
               $.LC4, %edx
         movl
               $1, %esi
         movl
         movl $8, %edi
         movl $0, %eax
         call __syslog_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• LHP(syslog,LOG_USER, "helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 3 31 0
         movl $.LC4, %edx
         movl $1, %esi
               $8, %edi
         movl
                %eax, %eax
         xorl
         call
               __syslog_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
         .loc 1 121 0
• asm(""); with gcc -g -00 produces:
          .loc 1 120 0
• asm(""); with gcc -g -01 produces:
          .loc 1 120 0
• asm(""); with gcc -g -03 produces:
          .loc 1 120 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
```

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;"); with gcc -g -03 produces:

```
# 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -O1 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
```

.loc 1 120 0

.loc 1 120 0

#NO_APP

movl \$4, -56(%rbp) .loc 1 121 0

• i = 4; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl \$4, 24(%rsp) .loc 1 121 0

• i = 4; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$4, 24(%rsp) .loc 1 121 0

• gi = 11; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• gi = 11; with gcc - g - 01 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• gi = 11; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax movl %eax, -48(%rbp) .loc 1 121 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 120 0
#NO_APP
movss .LC4(%rip), %xmm0
movss %xmm0, 28(%rsp)
.loc 1 121 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 120 0 #NO_APP

movl \$0x41400000, 28(%rsp)
.loc 1 121 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 120 0
#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0

movss %xmm0, gf(%rip)

.loc 1 121 0

• gf = 12.0; with gcc -g -03 produces:

.loc 1 120 0 #NO_APP

movl \$0x41400000, gf(%rip)
.loc 1 121 0

• i++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl -56(%rbp), %eax
addl \$1, %eax
movl %eax, -56(%rbp)
.loc 1 121 0

• i++; with gcc -g -01 produces:

```
movl
                  24(%rsp), %eax
          addl
                  $1, %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• i++; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          addl
                  $1, %eax
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• gi++; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  gi(%rip), %eax
          movl
                  $1, %eax
          addl
                  %eax, gi(%rip)
          movl
          .loc 1 121 0
• gi++; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  gi(%rip), %eax
          movl
          addl
                  $1, %eax
          movl
                  %eax, gi(%rip)
          .loc 1 121 0
• gi++; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  gi(%rip), %eax
          movl
          addl
                  $1, %eax
                  %eax, gi(%rip)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -60(%rbp), %eax
          cltq
                  a(, %rax, 4), %eax
          movl
                  %eax, -56(%rbp)
          movl
          .loc 1 121 0
```

.loc 1 120 0

#NO_APP

```
• j = a[i]; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  20(%rsp), %eax
          movl
          cltq
          movl
                  a(,%rax,4), %eax
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movslq 20(%rsp), %rax
                  a(,\%rax,4), \%eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L5:
          .loc 1 121 0 is_stmt 1
• assert(i >= 10); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
                  $9, %eax
          cmpl
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
```

.loc 1 120 0 #NO_APP 24(%rsp), %eax movl cmpl \$9, %eax .L274 jle .L5: .loc 1 121 0 • BSD_assert(i >= 10); with gcc -g -00 produces: .loc 1 120 0 #NO_APP movl -56(%rbp), %eax \$9, %eax cmpl .L5 jg .loc 1 120 0 is_stmt 0 discriminator 1 movl \$.LC4, %edx \$120, %esi movl movl \$.LC5, %edi __BSD_assert call .L5: .loc 1 121 0 is_stmt 1 • BSD_assert(i >= 10); with gcc -g -01 produces: .loc 1 120 0 #NO_APP movl 24(%rsp), %eax .loc 1 121 0 • BSD_assert(i >= 10); with gcc -g -03 produces: .loc 1 120 0 #NO_APP movl 24(%rsp), %eax .loc 1 121 0 • TRAD_assert(i >= 10); with gcc -g -00 produces: .loc 1 120 0 #NO_APP movl -56(%rbp), %eax cmpl \$9, %eax .L5 jg .loc 1 120 0 is_stmt 0 discriminator 1 stderr(%rip), %rax movq movl \$.LC4, %r8d

• assert(i >= 10); with gcc -g -03 produces:

```
$120, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
          movq
                  %rax, %rdi
                  $0, %eax
          movl
                  fprintf
          call
          movl
                  $1, %edi
          call
                  exit
  .L5:
          .loc 1 121 0 is_stmt 1
• TRAD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
                  .L5
          jg
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
          movl
                  $.LC4, %r9d
                  $120, %r8d
          movl
          movl
                  $.LC5, %ecx
                  $.LC6, %edx
          movl
          movl
                  $1, %esi
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
          call
                  __fprintf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
          movl
                  $1, %edi
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
          jle
                  .L21
          .loc 1 121 0
```

• I(i >= 10); with gcc -g -00 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                   .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L5:
          .loc 1 121 0 is_stmt 1
• I(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          {\tt cmpl}
                  $9, %eax
                   .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
          movl
                  $.LC4, %esi
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(i \geq= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
          jle
                   .L274
  .L5:
          .loc 1 121 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
```

• DI(i >= 10); with gcc -g -01 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• DI(i \geq 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
 produces:
 #NO_APP
  .LBB2:
          .loc 1 120 0
          movl
                  $1, -3132(%rbp)
          movq
                  $str, -2096(%rbp)
          jmp
                  .L5
  .L8:
          .loc 1 120 0 is_stmt 0 discriminator 2
          call
                  __ctype_b_loc
          movq
                  (%rax), %rdx
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
          addq
                  %rax, %rax
                  %rdx, %rax
          addq
          movzwl (%rax), %eax
          movzwl %ax, %eax
          andl
                  $512, %eax
                  %eax, %eax
          testl
          jne
                  .L6
          .loc 1 120 0 discriminator 1
                  $0, -3132(%rbp)
          movl
          jmp
                  .L7
  .L6:
          .loc 1 120 0 discriminator 2
          addq
                  $1, -2096(%rbp)
  .L5:
          .loc 1 120 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          testb
                  %al, %al
                  .L8
          jne
  .L7:
          movl
                  -3132(%rbp), %eax
```

```
.LBE2:
                  %eax, %eax
          testl
          jne
                  .L9
          movl
                  $120, %edx
          movl
                  $.LC4, %esi
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L9:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -01
 produces:
  .LVL5:
 #NO_APP
  .LBB269:
          .loc 1 120 0
          movzbl str(%rip), %ebx
          testb
                  %bl, %bl
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
  .LVL7:
  .L7:
          .loc 1 120 0 is_stmt 0 discriminator 2
          movsbq %bl, %rbx
          testb
                  $2, 1(%rdx,%rbx,2)
          jе
                  .L6
          addq
                  $1, %rax
  .LVL8:
          movzbl
                 (%rax), %ebx
                  %bl, %bl
          testb
                  .L7
          jne
  .LVL9:
  .L5:
  .LBE269:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -03
 produces:
  .LVL5:
 #NO_APP
  .LBB271:
          .loc 1 120 0
          movsbq str(%rip), %rbx
          testb %bl, %bl
```

```
jе
                  .L5
                  __ctype_b_loc
          call
  .LVL6:
          movq
                  (%rax), %rdx
          movl
                  $str, %eax
                  .L7
          jmp
  .LVL7:
          .p2align 4,,10
          .p2align 3
  .L2074:
          .loc 1 120 0 is_stmt 0 discriminator 2
          addq
                  $1, %rax
  .LVL8:
          movsbq (%rax), %rbx
          testb
                  %bl, %bl
                  .L5
          jе
  .LVL9:
  .L7:
                  $2, 1(%rdx,%rbx,2)
          testb
          jne
                  .L2074
  .LVL10:
  .LBE271:
          .loc 1 120 0
                  $120, %edx
          movl
          movl
                  $.LC10, %esi
                  $.LC11, %edi
          movl
          call
                  _I_default_handler
  .LVL11:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
  with gcc -g -00 produces:
 #NO_APP
  .LBB2:
          .loc 1 120 0
                  $1, -5188(%rbp)
          movl
          movl
                  $0, -5184(%rbp)
                  .L5
          jmp
  .L11:
  .LBB3:
          .loc 1 120 0 is_stmt 0 discriminator 2
                  $0, -2096(%rbp)
          movq
          movl
                  $0, -5180(%rbp)
          jmp
                  .L6
  .L8:
                  -5184(%rbp), %eax
          movl
          cltq
          movl
                  a(,\%rax,4), \%edx
```

```
-5180(%rbp), %eax
          movl
          cltq
          movl
                  a(, %rax, 4), %eax
          cmpl
                  %eax, %edx
          jne
                  .L7
          .loc 1 120 0 discriminator 1
                  $1, -2096(%rbp)
          addq
  .L7:
          .loc 1 120 0 discriminator 2
          addl
                  $1, -5180(%rbp)
  .L6:
          .loc 1 120 0 discriminator 1 \,
                  $9, -5180(%rbp)
          cmpl
          jle
                  .L8
          .loc 1 120 0 discriminator 3
                  -2096(%rbp), %rax
          movq
  .LBE3:
                  $1, %rax
          cmpq
          jе
                  .L9
          .loc 1 120 0 discriminator 1
                  $0, -5188(%rbp)
          movl
                  .L10
          jmp
  .L9:
          .loc 1 120 0 discriminator 2
          addl
                  $1, -5184(%rbp)
  .L5:
          .loc 1 120 0 discriminator 1
                  $9, -5184(%rbp)
          cmpl
                  .L11
          jle
  .L10:
                  -5188(%rbp), %eax
          movl
  .LBE2:
                  %eax, %eax
          testl
          jne
                  .L12
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L12:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
          .loc 1 120 0
                 $0, %ecx
          movl
  .LBE526:
```

```
.loc 1 119 0
                  $0, %eax
          movl
                  $0, %edx
          movl
          jmp
                  .L5
  .LVL6:
  .L1295:
  .LBB528:
          movl
                  $0, %eax
  .LVL7:
                  $0, %edx
          movl
  .LVL8:
  .L5:
  .LBB527:
          .loc 1 120 0 discriminator 2
          movslq %ecx, %rdi
          movslq %eax, %rsi
                  a(,%rsi,4), %ebx
          movl
                  %ebx, a(,%rdi,4)
          cmpl
          sete
                  %sil
          movzbl %sil, %esi
                  %rsi, %rdx
          addq
  .LVL9:
          addl
                  $1, %eax
  .LVL10:
          cmpl
                  $9, %eax
          jle
                  .L5
  .LBE527:
          .loc 1 120 0 is_stmt 0 discriminator 3
                  $1, %rdx
          cmpq
          jne
                  .L8
          .loc 1 120 0 discriminator 2
                  $1, %ecx
          addl
  .LVL11:
                  $9, %ecx
          cmpl
          jle
                  .L1295
  .LVL12:
  .L1038:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -03 produces:
  .LVL5:
 #NO_APP
  .LBB528:
  .LBB529:
          .loc 1 120 0
                  a(%rip), %r13d
          movl
          movl
                  a+4(%rip), %r12d
```

```
$a, %ecx
       movl
                a+8(%rip), %ebp
       movl
       movl
                a+12(%rip), %ebx
       movl
                a+16(%rip), %r11d
                a+20(%rip), %r10d
       movl
                a+24(%rip), %r9d
       movl
                a+28(%rip), %r8d
       movl
                a+32(%rip), %edi
       movl
                a+36(%rip), %esi
       movl
.LVL6:
.L5:
                (%rcx), %edx
       movl
                %eax, %eax
        xorl
        cmpl
                %r13d, %edx
        sete
                %al
.LVL7:
                %r14d, %r14d
       xorl
                %r12d, %edx
        cmpl
                %r14b
        sete
        addq
                %r14, %rax
.LVL8:
                %r14d, %r14d
        xorl
        cmpl
                %ebp, %edx
                %r14b
        sete
        addq
                %r14, %rax
.LVL9:
                %r14d, %r14d
        xorl
                %ebx, %edx
        cmpl
                %r14b
        sete
        addq
                %r14, %rax
.LVL10:
                %r14d, %r14d
        xorl
                %r11d, %edx
        cmpl
        sete
                %r14b
        addq
                %r14, %rax
.LVL11:
                %r14d, %r14d
        xorl
                %r10d, %edx
        cmpl
        sete
                %r14b
        addq
                %r14, %rax
.LVL12:
                %r14d, %r14d
        xorl
                %r9d, %edx
        cmpl
                %r14b
        sete
        addq
                %r14, %rax
.LVL13:
                %r14d, %r14d
        xorl
                %r8d, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
```

```
.LVL14:
                  %r14d, %r14d
          xorl
                  %edi, %edx
          cmpl
          sete
                  %r14b
          addq
                  %r14, %rax
  .LVL15:
                  %esi, %edx
          cmpl
          sete
                  %dl
          movzbl %dl, %edx
          addq
                  %rdx, %rax
  .LVL16:
  .LBE529:
                  $1, %rax
          cmpq
          jne
                  .L4164
          addq
                  $4, %rcx
          .loc 1 120 0 is_stmt 0 discriminator 2
                  $a+40, %rcx
          cmpq
                  .L5
          jne
  .LVL17:
  .L6:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• printf("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $.LC4, %edi
          movl
                  $0, %eax
          movl
          call
                  printf
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
                  $.LC4, %esi
          movl
                  $1, %edi
          movl
                  $0, %eax
          movl
          call
                  __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
```

• printf("helloworldn"); with gcc -g -03 produces:

```
.LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 104 0
         movl $.LC4, %esi
         movl $1, %edi
         xorl
                 %eax, %eax
         call
                  __printf_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movq
                  stdout(%rip), %rax
                  %rax, %rcx
         movq
                  $11, %edx
         movl
                  $1, %esi
         movl
                  $.LC4, %edi
         movl
         call
                  fwrite
          .loc 1 121 0
• L("helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
         movq stdout(%rip), %rcx
         movl
                  $11, %edx
                  $1, %esi
         movl
                  $.LC4, %edi
         movl
          call
                 fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
```

```
.loc 2 97 0
         movq stdout(%rip), %rcx
                 $11, %edx
         movl
         movl
                $1, %esi
                $.LC4, %edi
         movl
                  fwrite
         call
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl
                  $0, _dl_target(%rip)
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
         movl
               $0, _dl_target(%rip)
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _dl_target(%rip)
         movl
          .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl
                  $0, gi(%rip)
                  gi(%rip), %eax
         movl
                  $16, %eax
          andl
          testl %eax, %eax
                  .L5
          jе
          .loc 1 120 0 is_stmt 0 discriminator 1
                  stdout(%rip), %rax
                  %rax, %rcx
         movq
                  $11, %edx
         movl
                  $1, %esi
          movl
         movl
                  $.LC4, %edi
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
```

```
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
                  gi(%rip), %eax
          movl
                  $16, %al
          testb
          jе
                  .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
          movq stdout(%rip), %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
                  gi(%rip), %eax
          movl
          testb
                  $16, %al
                  .L1045
          jne
  .L5:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
                  $16, %eax
          andl
                  %eax, %eax
          testl
                  .L5
          jе
          .loc 1 120 0 is_stmt 0 discriminator 1
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
          movq
          movl
                  $11, %edx
          movl
                  $1, %esi
                  $.LC4, %edi
          movl
```

```
call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
                   .L5
          jе
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
          movq
                  stdout(%rip), %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $-1, gi(%rip)
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
          jne
                  .L1045
  .L5:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -32(\%rbp), \%rax
          movq
          movq
                  %rax, %rcx
                  $11, %edx
          movl
          movl
                  $1, %esi
                  $.LC4, %edi
          movl
                  fwrite
          call
          .loc 1 121 0
```

```
• LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
          movq %rbx, %rcx
                 $11, %edx
          movl
               $1, %esi
          movl
          movl $.LC4, %edi
               fwrite
          call
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 97 0
                 $11, %edx
          movl
                  %rbx, %rcx
          movq
                  $1, %esi
          movl
          movl
                  $.LC4, %edi
                  fwrite
          call
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
\bullet LHP(L_buffer_printf,buf,"helloworldn"); with gcc \, -00 produces:
          .loc 1 120 0
 #NO_APP
                  -40(%rbp), %rax
          movq
          movl
                  $.LC4, %esi
                  %rax, %rdi
          movq
          movl
                  $0, %eax
                 L_buffer_printf
          call
          .loc 1 121 0
• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -01 produces:
         .loc 1 120 0
 #NO_APP
```

```
$.LC4, %esi
          movl
                  %rbx, %rdi
          movq
          movl
                  $0, %eax
          call
                  L_buffer_printf
  .LVL6:
          .loc 1 121 0
• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:
          .loc 1 120 0
 #NO_APP
                  $.LC4, %esi
          movl
                  %rbx, %rdi
          movq
                  %eax, %eax
          xorl
          call
                  L_buffer_printf
  .LVL279:
          .loc 1 121 0
• LHP(syslog,LOG_USER, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $.LC4, %esi
          movl
          movl
                  $8, %edi
                  $0, %eax
          movl
          call
                  syslog
          .loc 1 121 0
• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
          .loc 3 31 0
          movl $.LC4, %edx
                 $1, %esi
          movl
               $8, %edi
          movl
                  $0, %eax
          movl
          call
                __syslog_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
```

• LHP(syslog,LOG_USER, "helloworldn"); with gcc -g -O3 produces:

```
.LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 3 31 0
         movl $.LC4, %edx
         movl $1, %esi
         movl $8, %edi
         xorl %eax, %eax
         call __syslog_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
• asm(""); with gcc -g -00 produces:
          .loc 1 120 0
• asm(""); with gcc -g -01 produces:
          .loc 1 120 0
• asm(""); with gcc -g -03 produces:
          .loc 1 120 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
```

• asm("nop;nop;"); with gcc -g -00 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
        nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
         .loc 1 120 0
 #NO_APP
         movl $4, -56(%rbp)
         .loc 1 121 0
• i = 4; with gcc -g -01 produces:
         .loc 1 120 0
 #NO_APP
         movl $4, 24(%rsp)
         .loc 1 121 0
```

• i = 4; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$4, 24(%rsp) .loc 1 121 0

• gi = 11; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip) .loc 1 121 0

• gi = 11; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• gi = 11; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax movl %eax, -48(%rbp) .loc 1 121 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0 movss %xmm0, 28(%rsp) .loc 1 121 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, 28(%rsp)
.loc 1 121 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, gf(%rip) .loc 1 121 0

• i++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl -56(%rbp), %eax addl \$1, %eax movl %eax, -56(%rbp) .loc 1 121 0

• i++; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl 24(%rsp), %eax addl \$1, %eax movl %eax, 24(%rsp) .loc 1 121 0

• i++; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl 24(%rsp), %eax addl \$1, %eax movl %eax, 24(%rsp) .loc 1 121 0

• gi++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• gi++; with gcc -g -01 produces:

.loc 1 120 0
#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• gi++; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• j = a[i]; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl -60(%rbp), %eax
cltq
movl a(,%rax,4), %eax
movl %eax, -56(%rbp)
.loc 1 121 0

• j = a[i]; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl 20(%rsp), %eax
cltq
movl a(,%rax,4), %eax
movl %eax, 24(%rsp)
.loc 1 121 0

• j = a[i]; with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movslq 20(%rsp), %rax
          movl
                  a(,%rax,4), %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
                  $9, %eax
          cmpl
          jg
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
               $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L5:
          .loc 1 121 0 is_stmt 1
• assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
                  $9, %eax
          cmpl
          jg
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• assert(i \geq= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 121 0
```

• BSD_assert(i >= 10); with gcc -g -00 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
               $.LC4, %edx
          movl
                  $120, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  __BSD_assert
  .L5:
          .loc 1 121 0 is_stmt 1
• BSD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 121 0
• BSD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
                  $9, %eax
          cmpl
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movq
                  stderr(%rip), %rax
          movl
                  $.LC4, %r8d
                  $120, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
                  %rax, %rdi
          movq
                  $0, %eax
          movl
          call
                  fprintf
          movl
                  $1, %edi
          call
                  exit
  .L5:
          .loc 1 121 0 is_stmt 1
```

• TRAD_assert(i >= 10); with gcc -g -01 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  $.LC4, %r9d
          movl
                  $120, %r8d
          movl
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
          movl
                  $1, %esi
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
                  __fprintf_chk
          call
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
                  $1, %edi
          movl
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
                  $9, %eax
          cmpl
          jle
                  .L21
          .loc 1 121 0
• I(i \geq 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
          .loc 1 121 0 is_stmt 1
```

```
.loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
          jg
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(i \geq 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 121 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
• DI(i >= 10); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• DI(i \ge 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
```

• I(i \geq = 10); with gcc -g -01 produces:

produces:

```
#NO_APP
  .LBB2:
          .loc 1 120 0
                  $1, -3132(%rbp)
          movl
          movq
                  $str, -2096(%rbp)
                  .L5
          jmp
  .L8:
          .loc 1 120 0 is_stmt 0 discriminator 2
          call
                  __ctype_b_loc
                  (\%rax), \%rdx
          movq
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
          addq
                  %rax, %rax
          addq
                  %rdx, %rax
          movzwl (%rax), %eax
          movzwl %ax, %eax
                  $512, %eax
          andl
                  %eax, %eax
          testl
          jne
                  .L6
          .loc 1 120 0 discriminator 1
                  $0, -3132(%rbp)
          movl
                  .L7
          jmp
  .L6:
          .loc 1 120 0 discriminator 2
                  $1, -2096(%rbp)
          addq
  .L5:
          .loc 1 120 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl
                 (%rax), %eax
          testb
                  %al, %al
                  .L8
          jne
  .L7:
                  -3132(%rbp), %eax
          movl
  .LBE2:
          testl
                 %eax, %eax
                  .L9
          jne
                  $120, %edx
          movl
          movl
                  $.LC4, %esi
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L9:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -01
 produces:
  .LVL5:
 #NO_APP
  .LBB269:
```

```
.loc 1 120 0
          movzbl str(%rip), %ebx
          testb
                  %bl, %bl
          jе
                  .L5
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
  .LVL7:
  .L7:
          .loc 1 120 0 is_stmt 0 discriminator 2
          movsbq %bl, %rbx
          testb
                  $2, 1(%rdx,%rbx,2)
          jе
                  .L6
          addq
                  $1, %rax
  .LVL8:
          movzbl (%rax), %ebx
                  %bl, %bl
          testb
                  .L7
          jne
  .LVL9:
  .L5:
  .LBE269:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -03
 produces:
  .LVL5:
 #NO_APP
  .LBB271:
          .loc 1 120 0
          movsbq str(%rip), %rbx
                  %bl, %bl
          testb
          jе
                  .L5
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
                  .L7
          jmp
  .LVL7:
          .p2align 4,,10
          .p2align 3
  .L2074:
          .loc 1 120 0 is_stmt 0 discriminator 2
          addq
                  $1, %rax
  .LVL8:
          movsbq (%rax), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
  .LVL9:
```

```
.L7:
                  $2, 1(%rdx,%rbx,2)
          testb
          jne
                  .L2074
  .LVL10:
  .LBE271:
          .loc 1 120 0
          movl
                  $120, %edx
                  $.LC10, %esi
          movl
                  $.LC11, %edi
          movl
                  _I_default_handler
          call
  .LVL11:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -00 produces:
 #NO_APP
  .LBB2:
          .loc 1 120 0
          movl
                  $1, -5188(%rbp)
                  $0, -5184(%rbp)
          movl
                  .L5
          jmp
  .L11:
  .LBB3:
          .loc 1 120 0 is_stmt 0 discriminator 2
                  $0, -2096(%rbp)
          movq
                  $0, -5180(%rbp)
          movl
                  .L6
          jmp
  .L8:
                  -5184(%rbp), %eax
          movl
          cltq
                  a(, %rax, 4), %edx
          movl
          movl
                  -5180(%rbp), %eax
          cltq
          movl
                  a(, %rax, 4), %eax
                  %eax, %edx
          cmpl
          jne
                  .L7
          .loc 1 120 0 discriminator 1
                  $1, -2096(%rbp)
          addq
  .L7:
          .loc 1 120 0 discriminator 2
                  $1, -5180(%rbp)
          addl
  .L6:
          .loc 1 120 0 discriminator 1
          cmpl
                  $9, -5180(%rbp)
          jle
                  .L8
          .loc 1 120 0 discriminator 3
                  -2096(%rbp), %rax
          movq
  .LBE3:
```

```
$1, %rax
          cmpq
                  .L9
          jе
          .loc 1 120 0 discriminator 1
          movl
                  $0, -5188(%rbp)
                  .L10
          jmp
  .L9:
          .loc 1 120 0 discriminator 2
          addl
                  $1, -5184(%rbp)
  .L5:
          .loc 1 120 0 discriminator 1
                  $9, -5184(%rbp)
          cmpl
          jle
                  .L11
  .L10:
          movl
                  -5188(%rbp), %eax
  .LBE2:
                  %eax, %eax
          testl
                  .L12
          jne
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
                  _I_default_handler
          call
  .L12:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
          .loc 1 120 0
                 $0, %ecx
          movl
  .LBE526:
          .loc 1 119 0
          movl
                  $0, %eax
          movl
                  $0, %edx
                  .L5
          jmp
  .LVL6:
  .L1295:
  .LBB528:
                  $0, %eax
          movl
  .LVL7:
                  $0, %edx
          movl
  .LVL8:
  .L5:
  .LBB527:
          .loc 1 120 0 discriminator 2
          movslq %ecx, %rdi
          movslq %eax, %rsi
                  a(,%rsi,4), %ebx
          movl
```

```
%ebx, a(,%rdi,4)
          cmpl
                  %sil
          sete
          movzbl %sil, %esi
          addq
                  %rsi, %rdx
  .LVL9:
          addl
                  $1, %eax
  .LVL10:
                  $9, %eax
          cmpl
          jle
                  .L5
  .LBE527:
          .loc 1 120 0 is_stmt 0 discriminator 3
                  $1, %rdx
          cmpq
          jne
                  .L8
          .loc 1 120 0 discriminator 2
          addl
                  $1, %ecx
  .LVL11:
                  $9, %ecx
          cmpl
                  .L1295
          jle
  .LVL12:
  .L1038:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -03 produces:
  .LVL5:
 #NO_APP
  .LBB528:
  .LBB529:
          .loc 1 120 0
                  a(%rip), %r13d
          movl
                  a+4(%rip), %r12d
          movl
                  $a, %ecx
          movl
          movl
                  a+8(%rip), %ebp
                  a+12(%rip), %ebx
          movl
                  a+16(%rip), %r11d
          movl
                  a+20(%rip), %r10d
          movl
                  a+24(%rip), %r9d
          movl
                  a+28(%rip), %r8d
          movl
                  a+32(%rip), %edi
          movl
                  a+36(%rip), %esi
          movl
  .LVL6:
  .L5:
          movl
                  (%rcx), %edx
          xorl
                  %eax, %eax
          {\tt cmpl}
                  %r13d, %edx
                  %al
          sete
  .LVL7:
          xorl
                  %r14d, %r14d
```

```
%r12d, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
.LVL8:
                %r14d, %r14d
       xorl
                %ebp, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
.LVL9:
                %r14d, %r14d
        xorl
                %ebx, %edx
        cmpl
        sete
                %r14b
                %r14, %rax
        addq
.LVL10:
        xorl
                %r14d, %r14d
                %r11d, %edx
        cmpl
        sete
                %r14b
                %r14, %rax
        addq
.LVL11:
        xorl
                %r14d, %r14d
                %r10d, %edx
        cmpl
                %r14b
        sete
        addq
                %r14, %rax
.LVL12:
        xorl
                %r14d, %r14d
                %r9d, %edx
        cmpl
        sete
                %r14b
                %r14, %rax
        addq
.LVL13:
                %r14d, %r14d
        xorl
        cmpl
                %r8d, %edx
                %r14b
        sete
                %r14, %rax
        addq
.LVL14:
        xorl
                %r14d, %r14d
                %edi, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
.LVL15:
                %esi, %edx
        cmpl
        sete
                %dl
                %dl, %edx
       movzbl
                %rdx, %rax
        addq
.LVL16:
.LBE529:
        cmpq
                $1, %rax
        jne
                .L4164
                $4, %rcx
        addq
        .loc 1 120 0 is_stmt 0 discriminator 2
        cmpq
                $a+40, %rcx
```

```
jne
                .L5
  .LVL17:
  .L6:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• printf("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $.LC4, %edi
         movl
                  $0, %eax
         movl
          call
                  printf
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
               $.LC4, %esi
         movl
                  $1, %edi
         movl
                $0, %eax
         movl
          call
                  __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 104 0
         movl $.LC4, %esi
                 $1, %edi
         movl
                 %eax, %eax
         xorl
         call
                __printf_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
```

 \bullet L("helloworldn"); with gcc -g -00 produces:

```
.loc 1 120 0
 #NO_APP
               stdout(%rip), %rax
         movq
         movq %rax, %rcx
         movl
              $11, %edx
              $1, %esi
         movl
               $.LC4, %edi
         movl
         call fwrite
         .loc 1 121 0
• L("helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
         .loc 2 97 0
         movq stdout(%rip), %rcx
               $11, %edx
         movl
         movl $1, %esi
         movl $.LC4, %edi
         call fwrite
  .LVL6:
  .LBE527:
  .LBE526:
         .loc 1 121 0
• L("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 2 97 0
         movq stdout(%rip), %rcx
         movl $11, %edx
         movl $1, %esi
         movl $.LC4, %edi
         call
                fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
         .loc 1 121 0
• DL("helloworldn"); with gcc -g -00 produces:
         .loc 1 120 0
 #NO_APP
                 $0, _dl_target(%rip)
         movl
         .loc 1 121 0
```

```
• DL("helloworldn"); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _dl_target(%rip)
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _dl_target(%rip)
          .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
                  gi(%rip), %eax
          movl
                  $16, %eax
          andl
          testl %eax, %eax
          jе
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
          movq stdout(%rip), %rax
                %rax, %rcx
          movq
          movl
                 $11, %edx
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
          movl
          testb
                  $16, %al
          jе
                  .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
```

```
call
                   fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
           .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
           .loc 1 120 0
  #NO_APP
                   $0, gi(%rip)
          movl
          movl
                   gi(%rip), %eax
                   $16, %al
          testb
                   .L1045
          jne
  .L5:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
           .loc 1 120 0
  #NO_APP
                   $-1, gi(%rip)
          movl
          movl
                   gi(%rip), %eax
          \quad \text{andl} \quad
                   $16, %eax
                   %eax, %eax
          testl
          jе
                   .L5
          .loc 1 120 0 is_stmt 0 discriminator 1 \,
                   stdout(%rip), %rax
          movq
                   %rax, %rcx
          movq
                   $11, %edx
          movl
                   $1, %esi
          movl
                   $.LC4, %edi
          movl
          call
                   fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
           .loc 1 120 0
  #NO_APP
                   $-1, gi(%rip)
          movl
          movl
                   gi(%rip), %eax
          testb
                   $16, %al
          jе
                   .L5
  .LVL5:
  .LBB526:
  .LBB527:
           .loc 2 97 0 discriminator 1
```

```
stdout(%rip), %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
          movl
                  $.LC4, %edi
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
                  $16, %al
          testb
          jne
                  .L1045
  .L5:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -32(%rbp), %rax
          movq
                  %rax, %rcx
          movq
                  $11, %edx
          movl
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          call
                  fwrite
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
                %rbx, %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
          movl
                  $.LC4, %edi
          call
                 fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
```

• LHP(fprintf,log,"helloworldn"); with gcc -g -03 produces: .LVL277: #NO_APP .LBB1050: .LBB1051: .loc 2 97 0 movl \$11, %edx movq %rbx, %rcx \$1, %esi movl \$.LC4, %edi movl call fwrite .LVL278: .LBE1051: .LBE1050: .loc 1 121 0 • LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -OO produces: .loc 1 120 0 #NO_APP -40(%rbp), %rax movq movl \$.LC4, %esi %rax, %rdi movq **\$0,** %eax movl L_buffer_printf call .loc 1 121 0 \bullet LHP(L_buffer_printf,buf,"helloworldn"); with gcc $\,$ -g $\,$ -O1 produces: .loc 1 120 0 #NO_APP \$.LC4, %esi movl movq %rbx, %rdi \$0, %eax movl L_buffer_printf call .LVL6: .loc 1 121 0 • LHP(L_buffer_printf,buf, "helloworldn"); with gcc -g -03 produces: .loc 1 120 0 #NO_APP \$.LC4, %esi movl movq %rbx, %rdi xorl %eax, %eax L_buffer_printf call .LVL279: .loc 1 121 0

• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -OO produces: .loc 1 120 0 #NO_APP \$.LC4, %esi movl \$8, %edi movl **\$0,** %eax movl call syslog .loc 1 121 0 \bullet LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces: .LVL5: #NO_APP .LBB526: .LBB527: .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h" .loc 3 31 0 \$.LC4, %edx movl movl \$1, %esi movl \$8, %edi movl \$0, %eax call __syslog_chk .LVL6: .LBE527: .LBE526: .loc 1 121 0 • LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O3 produces: .LVL277: #NO_APP .LBB1050: .LBB1051: .loc 3 31 0 movl \$.LC4, %edx movl \$1, %esi \$8, %edi movl xorl %eax, %eax call __syslog_chk .LVL278: .LBE1051: .LBE1050: .loc 1 121 0

• asm(""); with gcc -g -00 produces:

.loc 1 120 0

```
.loc 1 120 0
• asm(""); with gcc -g -03 produces:
          .loc 1 120 0
• asm("nop"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
```

 \bullet asm(""); with gcc -g -01 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
        nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;"); with gcc -g -00 produces:

```
• asm("nop;nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl
               $4, -56(%rbp)
          .loc 1 121 0
• i = 4; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
         movl
               $4, 24(%rsp)
         .loc 1 121 0
• i = 4; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
         movl $4, 24(%rsp)
         .loc 1 121 0
• gi = 11; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl $11, gi(%rip)
          .loc 1 121 0
• gi = 11; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
```

\$11, gi(%rip)

movl

.loc 1 121 0

• gi = 11; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, -48(%rbp)
.loc 1 121 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0 movss %xmm0, 28(%rsp) .loc 1 121 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, 28(%rsp) .loc 1 121 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -03 produces:

```
.loc 1 120 0

#NO_APP

movl $0x41400000, gf(%rip)
.loc 1 121 0
```

• i++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl -56(%rbp), %eax
addl \$1, %eax
movl %eax, -56(%rbp)
.loc 1 121 0

• i++; with gcc -g -01 produces:

• i++; with gcc -g -03 produces:

.loc 1 120 0
#NO_APP

movl 24(%rsp), %eax
addl \$1, %eax
movl %eax, 24(%rsp)
.loc 1 121 0

• gi++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• asm(""); with gcc -g -00 produces:

.loc 1 120 0

• asm(""); with gcc -g -01 produces:

.loc 1 120 0

 \bullet asm(""); with gcc -g -03 produces:

.loc 1 120 0

• asm("nop"); with gcc -g -00 produces:

.loc 1 120 0 # 120 "tmp.c" 1 nop # 0 "" 2

 \bullet asm("nop"); with gcc -g -01 produces:

.loc 1 120 0 # 120 "tmp.c" 1 nop # 0 "" 2

• asm("nop"); with gcc -g -03 produces:

.loc 1 120 0 # 120 "tmp.c" 1 nop # 0 "" 2

• asm("nop;nop;"); with gcc -g -00 produces:

.loc 1 120 0 # 120 "tmp.c" 1 nop;nop; # 0 "" 2

• asm("nop;nop;"); with gcc -g -01 produces:

.loc 1 120 0 # 120 "tmp.c" 1 nop;nop; # 0 "" 2

• asm("nop;nop;"); with gcc -g -03 produces:

.loc 1 120 0 # 120 "tmp.c" 1 nop;nop; # 0 "" 2

• asm("nop;nop;nop;"); with gcc -g -00 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;nop;nop;"); with gcc -g -01 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
         .loc 1 120 0
 #NO_APP
         movl
                $4, -56(%rbp)
         .loc 1 121 0
• i = 4; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
               $4, 24(%rsp)
         movl
         .loc 1 121 0
• i = 4; with gcc -g -03 produces:
         .loc 1 120 0
 #NO_APP
         movl
               $4, 24(%rsp)
          .loc 1 121 0
• gi = 11; with gcc -g -00 produces:
         .loc 1 120 0
 #NO_APP
               $11, gi(%rip)
         movl
         .loc 1 121 0
• gi = 11; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
         movl $11, gi(%rip)
         .loc 1 121 0
```

• gi = 11; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax movl %eax, -48(%rbp) .loc 1 121 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, 28(%rsp)
.loc 1 121 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, 28(%rsp)
.loc 1 121 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, gf(%rip)
.loc 1 121 0

• i++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl -56(%rbp), %eax
addl \$1, %eax
movl %eax, -56(%rbp)
.loc 1 121 0

• i++; with gcc -g -01 produces:

• i++; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl 24(%rsp), %eax
addl \$1, %eax
movl %eax, 24(%rsp)
.loc 1 121 0

• gi++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• gi++; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• gi++; with gcc -g -03 produces:

```
#NO_APP
          movl
                  gi(%rip), %eax
          addl
                  $1, %eax
          movl
                  %eax, gi(%rip)
          .loc 1 121 0
• j = a[i]; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -60(%rbp), %eax
          movl
          cltq
          movl
                  a(, %rax, 4), %eax
          movl
                  %eax, -56(%rbp)
          .loc 1 121 0
• j = a[i]; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  20(%rsp), %eax
          movl
          cltq
                  a(, %rax, 4), %eax
          movl
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movslq 20(%rsp), %rax
                  a(, %rax, 4), %eax
          movl
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          cmpl
                  $9, %eax
          jg
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
          .loc 1 121 0 is_stmt 1
```

.loc 1 120 0

• assert(i >= 10); with gcc -g -O1 produces: .loc 1 120 0 #NO_APP 24(%rsp), %eax movl \$9, %eax cmpljg .L5 .loc 1 120 0 is_stmt 0 discriminator 1 \$120, %edx movl \$.LC4, %esi movl \$.LC5, %edi movl call _I_default_handler .LVL5: .L5: .loc 1 121 0 is_stmt 1 • assert(i >= 10); with gcc -g -03 produces: .loc 1 120 0 #NO_APP movl 24(%rsp), %eax cmpl\$9, %eax jle .L274 .L5: .loc 1 121 0 • BSD_assert(i >= 10); with gcc -g -00 produces: .loc 1 120 0 #NO_APP -56(%rbp), %eax movl cmpl \$9, %eax .L5 jg .loc 1 120 0 is_stmt 0 discriminator 1 movl \$.LC4, %edx movl \$120, %esi \$.LC5, %edi movl call __BSD_assert .L5: .loc 1 121 0 is_stmt 1 • BSD_assert(i >= 10); with gcc -g -01 produces: .loc 1 120 0 #NO_APP movl24(%rsp), %eax .loc 1 121 0

• BSD_assert(i >= 10); with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movq
                  stderr(%rip), %rax
          movl
                  $.LC4, %r8d
                  $120, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
                  %rax, %rdi
          movq
                  $0, %eax
          movl
          call
                  fprintf
                  $1, %edi
          movl
          call
                  exit
  .L5:
          .loc 1 121 0 is_stmt 1
• TRAD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
                  .L5
          jg
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1 \,
                 $.LC4, %r9d
          movl
          movl
                  $120, %r8d
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
                  $1, %esi
          movl
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
          call
                  __fprintf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
```

```
$1, %edi
          movl
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
                  $9, %eax
          cmpl
          jle
                  .L21
          .loc 1 121 0
• I(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
          .loc 1 121 0 is_stmt 1
• I(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
```

• I(i \geq 10); with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 121 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
• DI(i \ge 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• DI(i \ge 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
 produces:
 #NO_APP
  .LBB2:
          .loc 1 120 0
          movl $1, -3132(%rbp)
                  $str, -2096(%rbp)
          movq
          jmp
                  .L5
  .L8:
          .loc 1 120 0 is_stmt 0 discriminator 2
                  __ctype_b_loc
          call
                  (\%rax), \%rdx
          movq
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
          addq
                  %rax, %rax
          addq
                  %rdx, %rax
          movzwl (%rax), %eax
          movzwl %ax, %eax
```

```
$512, %eax
          andl
                  %eax, %eax
          testl
          jne
                  .L6
          .loc 1 120 0 discriminator 1
          movl
                  $0, -3132(%rbp)
                  .L7
          jmp
  .L6:
          .loc 1 120 0 discriminator 2
                  $1, -2096(%rbp)
          addq
  .L5:
          .loc 1 120 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl
                  (%rax), %eax
          testb
                  %al, %al
          jne
                  .L8
  .L7:
                  -3132(%rbp), %eax
          movl
  .LBE2:
                  %eax, %eax
          testl
          jne
                  .L9
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L9:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -01
 produces:
  .LVL5:
 #NO_APP
  .LBB269:
          .loc 1 120 0
          movzbl str(%rip), %ebx
          testb
                  %bl, %bl
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
                  $str, %eax
          movl
  .LVL7:
  .L7:
          .loc 1 120 0 is_stmt 0 discriminator 2
          movsbq %bl, %rbx
          testb
                  $2, 1(%rdx,%rbx,2)
          jе
                  .L6
                  $1, %rax
          addq
  .LVL8:
          movzbl (%rax), %ebx
```

```
.L5:
  .LBE269:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -03
 produces:
  .LVL5:
 #NO_APP
  .LBB271:
          .loc 1 120 0
          movsbq str(%rip), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
                  .L7
          jmp
  .LVL7:
          .p2align 4,,10
          .p2align 3
  .L2074:
          .loc 1 120 0 is_stmt 0 discriminator 2
          addq
                  $1, %rax
  .LVL8:
          movsbq (%rax), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
  .LVL9:
  .L7:
                  $2, 1(%rdx,%rbx,2)
          testb
          jne
                  .L2074
  .LVL10:
  .LBE271:
          .loc 1 120 0
          movl
                  $120, %edx
                  $.LC10, %esi
          movl
                  $.LC11, %edi
          movl
                  _I_default_handler
          call
  .LVL11:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -00 produces:
```

%bl, %bl

.L7

testb jne

.LVL9:

```
#NO_APP
.LBB2:
        .loc 1 120 0
        movl
                $1, -5188(%rbp)
        movl
                $0, -5184(%rbp)
                .L5
        jmp
.L11:
.LBB3:
        .loc 1 120 0 is_stmt 0 discriminator 2
                $0, -2096(%rbp)
        movq
                $0, -5180(%rbp)
        movl
                .L6
        jmp
.L8:
        movl
                -5184(%rbp), %eax
        cltq
                a(, %rax, 4), %edx
        movl
                -5180(%rbp), %eax
        movl
        cltq
                a(, %rax, 4), %eax
        movl
        cmpl
                %eax, %edx
                .L7
        jne
        .loc 1 120 0 discriminator 1
        addq
                $1, -2096(%rbp)
.L7:
        .loc 1 120 0 discriminator 2
        addl
                $1, -5180(%rbp)
.L6:
        .loc 1 120 0 discriminator 1
                $9, -5180(%rbp)
        cmpl
        jle
                .L8
        .loc 1 120 0 discriminator 3
                -2096(%rbp), %rax
        movq
.LBE3:
                $1, %rax
        cmpq
        jе
                .L9
        .loc 1 120 0 discriminator 1
                $0, -5188(%rbp)
        movl
        jmp
                .L10
.L9:
        .loc 1 120 0 discriminator 2
        addl
                $1, -5184(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
                $9, -5184(%rbp)
        cmpl
        jle
                .L11
.L10:
        movl
                -5188(%rbp), %eax
.LBE2:
                %eax, %eax
        testl
        jne
                .L12
```

```
$120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L12:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
          .loc 1 120 0
               $0, %ecx
  .LBE526:
          .loc 1 119 0
                  $0, %eax
          movl
                  $0, %edx
          movl
          jmp
                  .L5
  .LVL6:
  .L1295:
  .LBB528:
          movl
                  $0, %eax
  .LVL7:
                  $0, %edx
          movl
  .LVL8:
  .L5:
  .LBB527:
          .loc 1 120 0 discriminator 2
          movslq %ecx, %rdi
          movslq %eax, %rsi
                  a(,%rsi,4), %ebx
          movl
          cmpl
                  %ebx, a(, %rdi, 4)
          sete
                  %sil
          movzbl %sil, %esi
                  %rsi, %rdx
          addq
  .LVL9:
          addl
                  $1, %eax
  .LVL10:
          cmpl
                  $9, %eax
          jle
                  .L5
  .LBE527:
          .loc 1 120 0 is_stmt 0 discriminator 3
          cmpq
                  $1, %rdx
          jne
                  .L8
          .loc 1 120 0 discriminator 2
                  $1, %ecx
          addl
  .LVL11:
          cmpl
                  $9, %ecx
```

```
.L1295
          jle
  .LVL12:
  .L1038:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -03 produces:
  .LVL5:
 #NO_APP
  .LBB528:
  .LBB529:
          .loc 1 120 0
          movl
                  a(%rip), %r13d
                  a+4(%rip), %r12d
          movl
                  $a, %ecx
          movl
                  a+8(%rip), %ebp
          movl
                  a+12(%rip), %ebx
          movl
          movl
                  a+16(%rip), %r11d
                  a+20(%rip), %r10d
          movl
                  a+24(%rip), %r9d
          movl
                  a+28(%rip), %r8d
          movl
                  a+32(%rip), %edi
          movl
          movl
                  a+36(%rip), %esi
  .LVL6:
  .L5:
                  (%rcx), %edx
          movl
                  %eax, %eax
          xorl
          cmpl
                  %r13d, %edx
          sete
                  %al
  .LVL7:
                  %r14d, %r14d
          xorl
                  %r12d, %edx
          cmpl
          sete
                  %r14b
          addq
                  %r14, %rax
  .LVL8:
                  %r14d, %r14d
          xorl
                  %ebp, %edx
          cmpl
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL9:
                  %r14d, %r14d
          xorl
                  %ebx, %edx
          cmpl
          sete
                  %r14b
          addq
                  %r14, %rax
  .LVL10:
                  %r14d, %r14d
          xorl
                  %r11d, %edx
          cmpl
```

sete

%r14b

```
%r14, %rax
          addq
  .LVL11:
                  %r14d, %r14d
          xorl
          cmpl
                  %r10d, %edx
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL12:
                  %r14d, %r14d
          xorl
                  %r9d, %edx
          cmpl
                  %r14b
          sete
                  %r14, %rax
          addq
  .LVL13:
                  %r14d, %r14d
          xorl
          cmpl
                  %r8d, %edx
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL14:
                  %r14d, %r14d
          xorl
                  %edi, %edx
          cmpl
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL15:
          cmpl
                  %esi, %edx
                  %dl
          sete
          movzbl %dl, %edx
          addq
                  %rdx, %rax
  .LVL16:
  .LBE529:
                  $1, %rax
          cmpq
          jne
                  .L4164
          addq
                  $4, %rcx
          .loc 1 120 0 is_stmt 0 discriminator 2 \,
                  $a+40, %rcx
          cmpq
          jne
                   .L5
  .LVL17:
  .L6:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• printf("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $.LC4, %edi
          movl
          movl
                  $0, %eax
          call
                  printf
```

• printf("helloworldn"); with gcc -g -01 produces:

.loc 1 121 0

```
.LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
         movl $.LC4, %esi
         movl $1, %edi
         movl
               $0, %eax
         call
                __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 104 0
         movl $.LC4, %esi
                 $1, %edi
         movl
         xorl %eax, %eax
         call
                 __printf_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                 stdout(%rip), %rax
         movq
                 %rax, %rcx
         movq
                 $11, %edx
         movl
                 $1, %esi
         movl
         movl
                 $.LC4, %edi
          call
                 fwrite
          .loc 1 121 0
• L("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
```

```
stdout(%rip), %rcx
         movq
                 $11, %edx
         movl
                 $1, %esi
         movl
         movl
               $.LC4, %edi
         call fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 2 97 0
         movq stdout(%rip), %rcx
                $11, %edx
         movl
         movl $1, %esi
         movl
               $.LC4, %edi
         call
                 fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
         .loc 1 121 0
• DL("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl $0, _dl_target(%rip)
         .loc 1 121 0
• DL("helloworldn"); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
         movl
                $0, _dl_target(%rip)
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                 $0, _dl_target(%rip)
         .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
```

```
.loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
          andl
                  $16, %eax
                  %eax, %eax
          testl
          jе
          .loc 1 120 0 is_stmt 0 discriminator 1
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  $0, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
                  $16, %al
          testb
                  .L5
          jе
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
                  .L1045
          jne
  .L5:
          .loc 1 121 0
```

```
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $-1, gi(%rip)
                  gi(%rip), %eax
          movl
                  $16, %eax
          andl
                  %eax, %eax
          testl
                   .L5
          jе
          .loc 1 120 0 is_stmt 0 discriminator 1 \,
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  $-1, gi(%rip)
          movl
                  gi(%rip), %eax
          movl
                  $16, %al
          testb
          jе
                   .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
                  fwrite
          call
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $-1, gi(%rip)
          movl
                  gi(%rip), %eax
                  $16, %al
```

testb

```
.L1045
          jne
  .L5:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -OO produces:
          .loc 1 120 0
 #NO_APP
                  -32(\%rbp), \%rax
          movq
          movq
                  %rax, %rcx
          movl
                  $11, %edx
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
          movq %rbx, %rcx
          movl
                  $11, %edx
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 97 0
                 $11, %edx
          movl
                  %rbx, %rcx
          movq
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
```

• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -00 produces:

```
.loc 1 120 0

#NO_APP

movq -40(%rbp), %rax
movl $.LC4, %esi
movq %rax, %rdi
movl $0, %eax
call L_buffer_printf
.loc 1 121 0
```

 \bullet LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -01 produces:

```
.loc 1 120 0

#NO_APP

movl $.LC4, %esi
movq %rbx, %rdi
movl $0, %eax
call L_buffer_printf
.LVL6:
.loc 1 121 0
```

• LHP(L_buffer_printf,buf, "helloworldn"); with gcc -g -03 produces:

```
.loc 1 120 0

#NO_APP

movl $.LC4, %esi
movq %rbx, %rdi
xorl %eax, %eax
call L_buffer_printf
.LVL279:
.loc 1 121 0
```

• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -OO produces:

```
.loc 1 120 0
#NO_APP

movl $.LC4, %esi
movl $8, %edi
movl $0, %eax
call syslog
.loc 1 121 0
```

• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:

```
.LVL5:
#NO_APP
.LBB526:
.LBB527:
```

```
.file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
          .loc 3 31 0
         movl $.LC4, %edx
               $1, %esi
         movl
         movl $8, %edi
         movl $0, %eax
         call __syslog_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• LHP(syslog,LOG_USER, "helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 3 31 0
         movl $.LC4, %edx
         movl $1, %esi
               $8, %edi
         movl
                %eax, %eax
         xorl
         call
               __syslog_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
         .loc 1 121 0
• asm(""); with gcc -g -00 produces:
          .loc 1 120 0
• asm(""); with gcc -g -01 produces:
          .loc 1 120 0
• asm(""); with gcc -g -03 produces:
          .loc 1 120 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
```

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;"); with gcc -g -03 produces:

```
# 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -O1 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
```

.loc 1 120 0

.loc 1 120 0

#NO_APP

movl \$4, -56(%rbp) .loc 1 121 0

• i = 4; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl \$4, 24(%rsp) .loc 1 121 0

• i = 4; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$4, 24(%rsp)
.loc 1 121 0

• gi = 11; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• gi = 11; with gcc - g - 01 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• gi = 11; with gcc - g - 03 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax movl %eax, -48(%rbp) .loc 1 121 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 120 0
#NO_APP
movss .LC4(%rip), %xmm0
movss %xmm0, 28(%rsp)
.loc 1 121 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 120 0 #NO_APP

movl \$0x41400000, 28(%rsp)
.loc 1 121 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 120 0
#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0

movss %xmm0, gf(%rip)

.loc 1 121 0

• gf = 12.0; with gcc -g -03 produces:

.loc 1 120 0 #NO_APP

movl \$0x41400000, gf(%rip)
.loc 1 121 0

• i++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl -56(%rbp), %eax
addl \$1, %eax
movl %eax, -56(%rbp)
.loc 1 121 0

• i++; with gcc -g -01 produces:

```
movl
                  24(%rsp), %eax
          addl
                  $1, %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• i++; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          addl
                  $1, %eax
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• gi++; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  gi(%rip), %eax
          movl
                  $1, %eax
          addl
                  %eax, gi(%rip)
          movl
          .loc 1 121 0
• gi++; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  gi(%rip), %eax
          movl
          addl
                  $1, %eax
          movl
                  %eax, gi(%rip)
          .loc 1 121 0
• gi++; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  gi(%rip), %eax
          movl
          addl
                  $1, %eax
                  %eax, gi(%rip)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -60(%rbp), %eax
          cltq
                  a(, %rax, 4), %eax
          movl
                  %eax, -56(%rbp)
          movl
          .loc 1 121 0
```

.loc 1 120 0

#NO_APP

```
• j = a[i]; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  20(%rsp), %eax
          movl
          cltq
          movl
                  a(,%rax,4), %eax
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movslq 20(%rsp), %rax
                  a(,\%rax,4), \%eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L5:
          .loc 1 121 0 is_stmt 1
• assert(i >= 10); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
                  $9, %eax
          cmpl
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
```

.loc 1 120 0 #NO_APP 24(%rsp), %eax movl cmpl \$9, %eax .L274 jle .L5: .loc 1 121 0 • BSD_assert(i >= 10); with gcc -g -00 produces: .loc 1 120 0 #NO_APP movl -56(%rbp), %eax \$9, %eax cmpl .L5 jg .loc 1 120 0 is_stmt 0 discriminator 1 movl \$.LC4, %edx \$120, %esi movl movl \$.LC5, %edi __BSD_assert call .L5: .loc 1 121 0 is_stmt 1 • BSD_assert(i >= 10); with gcc -g -01 produces: .loc 1 120 0 #NO_APP movl 24(%rsp), %eax .loc 1 121 0 • BSD_assert(i >= 10); with gcc -g -03 produces: .loc 1 120 0 #NO_APP movl 24(%rsp), %eax .loc 1 121 0 • TRAD_assert(i >= 10); with gcc -g -00 produces: .loc 1 120 0 #NO_APP movl -56(%rbp), %eax cmpl \$9, %eax .L5 jg .loc 1 120 0 is_stmt 0 discriminator 1 stderr(%rip), %rax movq movl \$.LC4, %r8d

• assert(i >= 10); with gcc -g -03 produces:

```
$120, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
          movq
                  %rax, %rdi
                  $0, %eax
          movl
                  fprintf
          call
          movl
                  $1, %edi
          call
                  exit
  .L5:
          .loc 1 121 0 is_stmt 1
• TRAD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
          movl
                  $.LC4, %r9d
                  $120, %r8d
          movl
          movl
                  $.LC5, %ecx
                  $.LC6, %edx
          movl
          movl
                  $1, %esi
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
          call
                  __fprintf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
          movl
                  $1, %edi
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
          jle
                  .L21
          .loc 1 121 0
```

• I(i >= 10); with gcc -g -00 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                   .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L5:
          .loc 1 121 0 is_stmt 1
• I(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          {\tt cmpl}
                  $9, %eax
                   .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(i \geq= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
          jle
                   .L274
  .L5:
          .loc 1 121 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
```

• DI(i >= 10); with gcc -g -01 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• DI(i \geq 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
 produces:
 #NO_APP
  .LBB2:
          .loc 1 120 0
          movl
                  $1, -3132(%rbp)
          movq
                  $str, -2096(%rbp)
          jmp
                  .L5
  .L8:
          .loc 1 120 0 is_stmt 0 discriminator 2
          call
                  __ctype_b_loc
          movq
                  (%rax), %rdx
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
          addq
                  %rax, %rax
                  %rdx, %rax
          addq
          movzwl (%rax), %eax
          movzwl %ax, %eax
          andl
                  $512, %eax
                  %eax, %eax
          testl
          jne
                  .L6
          .loc 1 120 0 discriminator 1
                  $0, -3132(%rbp)
          movl
          jmp
                  .L7
  .L6:
          .loc 1 120 0 discriminator 2
          addq
                  $1, -2096(%rbp)
  .L5:
          .loc 1 120 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          testb
                  %al, %al
                  .L8
          jne
  .L7:
          movl
                  -3132(%rbp), %eax
```

```
.LBE2:
                  %eax, %eax
          testl
          jne
                  .L9
          movl
                  $120, %edx
          movl
                  $.LC4, %esi
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L9:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -01
 produces:
  .LVL5:
 #NO_APP
  .LBB269:
          .loc 1 120 0
          movzbl str(%rip), %ebx
          testb
                  %bl, %bl
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
  .LVL7:
  .L7:
          .loc 1 120 0 is_stmt 0 discriminator 2
          movsbq %bl, %rbx
          testb
                  $2, 1(%rdx,%rbx,2)
          jе
                  .L6
          addq
                  $1, %rax
  .LVL8:
          movzbl
                 (%rax), %ebx
                  %bl, %bl
          testb
                  .L7
          jne
  .LVL9:
  .L5:
  .LBE269:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -03
 produces:
  .LVL5:
 #NO_APP
  .LBB271:
          .loc 1 120 0
          movsbq str(%rip), %rbx
          testb %bl, %bl
```

```
jе
                  .L5
                  __ctype_b_loc
          call
  .LVL6:
          movq
                  (%rax), %rdx
          movl
                  $str, %eax
                  .L7
          jmp
  .LVL7:
          .p2align 4,,10
          .p2align 3
  .L2074:
          .loc 1 120 0 is_stmt 0 discriminator 2
          addq
                  $1, %rax
  .LVL8:
          movsbq (%rax), %rbx
          testb
                  %bl, %bl
                  .L5
          jе
  .LVL9:
  .L7:
                  $2, 1(%rdx,%rbx,2)
          testb
          jne
                  .L2074
  .LVL10:
  .LBE271:
          .loc 1 120 0
                  $120, %edx
          movl
          movl
                  $.LC10, %esi
                  $.LC11, %edi
          movl
          call
                  _I_default_handler
  .LVL11:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
  with gcc -g -00 produces:
 #NO_APP
  .LBB2:
          .loc 1 120 0
                  $1, -5188(%rbp)
          movl
          movl
                  $0, -5184(%rbp)
                  .L5
          jmp
  .L11:
  .LBB3:
          .loc 1 120 0 is_stmt 0 discriminator 2
                  $0, -2096(%rbp)
          movq
          movl
                  $0, -5180(%rbp)
          jmp
                  .L6
  .L8:
                  -5184(%rbp), %eax
          movl
          cltq
          movl
                  a(, %rax, 4), %edx
```

```
-5180(%rbp), %eax
          movl
          cltq
          movl
                  a(, %rax, 4), %eax
          cmpl
                  %eax, %edx
          jne
                  .L7
          .loc 1 120 0 discriminator 1
                  $1, -2096(%rbp)
          addq
  .L7:
          .loc 1 120 0 discriminator 2
          addl
                  $1, -5180(%rbp)
  .L6:
          .loc 1 120 0 discriminator 1 \,
                  $9, -5180(%rbp)
          cmpl
          jle
                  .L8
          .loc 1 120 0 discriminator 3
                  -2096(%rbp), %rax
          movq
  .LBE3:
                  $1, %rax
          cmpq
          jе
                  .L9
          .loc 1 120 0 discriminator 1
                  $0, -5188(%rbp)
          movl
                  .L10
          jmp
  .L9:
          .loc 1 120 0 discriminator 2
          addl
                  $1, -5184(%rbp)
  .L5:
          .loc 1 120 0 discriminator 1
                  $9, -5184(%rbp)
          cmpl
                  .L11
          jle
  .L10:
                  -5188(%rbp), %eax
          movl
  .LBE2:
                  %eax, %eax
          testl
          jne
                  .L12
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L12:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
          .loc 1 120 0
                 $0, %ecx
          movl
  .LBE526:
```

```
.loc 1 119 0
                  $0, %eax
          movl
                  $0, %edx
          movl
          jmp
                  .L5
  .LVL6:
  .L1295:
  .LBB528:
          movl
                  $0, %eax
  .LVL7:
                  $0, %edx
          movl
  .LVL8:
  .L5:
  .LBB527:
          .loc 1 120 0 discriminator 2
          movslq %ecx, %rdi
          movslq %eax, %rsi
                  a(,%rsi,4), %ebx
          movl
                  %ebx, a(,%rdi,4)
          cmpl
          sete
                  %sil
          movzbl %sil, %esi
                  %rsi, %rdx
          addq
  .LVL9:
          addl
                  $1, %eax
  .LVL10:
          cmpl
                  $9, %eax
          jle
                  .L5
  .LBE527:
          .loc 1 120 0 is_stmt 0 discriminator 3
                  $1, %rdx
          cmpq
          jne
                  .L8
          .loc 1 120 0 discriminator 2
                  $1, %ecx
          addl
  .LVL11:
                  $9, %ecx
          cmpl
          jle
                  .L1295
  .LVL12:
  .L1038:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -03 produces:
  .LVL5:
 #NO_APP
  .LBB528:
  .LBB529:
          .loc 1 120 0
                  a(%rip), %r13d
          movl
          movl
                  a+4(%rip), %r12d
```

```
$a, %ecx
       movl
                a+8(%rip), %ebp
       movl
       movl
                a+12(%rip), %ebx
       movl
                a+16(%rip), %r11d
                a+20(%rip), %r10d
       movl
                a+24(%rip), %r9d
       movl
                a+28(%rip), %r8d
       movl
                a+32(%rip), %edi
       movl
                a+36(%rip), %esi
       movl
.LVL6:
.L5:
                (%rcx), %edx
       movl
                %eax, %eax
        xorl
        cmpl
                %r13d, %edx
        sete
                %al
.LVL7:
                %r14d, %r14d
       xorl
                %r12d, %edx
        cmpl
                %r14b
        sete
        addq
                %r14, %rax
.LVL8:
                %r14d, %r14d
        xorl
        cmpl
                %ebp, %edx
                %r14b
        sete
        addq
                %r14, %rax
.LVL9:
                %r14d, %r14d
        xorl
                %ebx, %edx
        cmpl
                %r14b
        sete
        addq
                %r14, %rax
.LVL10:
                %r14d, %r14d
        xorl
                %r11d, %edx
        cmpl
        sete
                %r14b
        addq
                %r14, %rax
.LVL11:
                %r14d, %r14d
        xorl
                %r10d, %edx
        cmpl
        sete
                %r14b
        addq
                %r14, %rax
.LVL12:
                %r14d, %r14d
        xorl
                %r9d, %edx
        cmpl
                %r14b
        sete
        addq
                %r14, %rax
.LVL13:
                %r14d, %r14d
        xorl
                %r8d, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
```

```
.LVL14:
                  %r14d, %r14d
          xorl
                  %edi, %edx
          cmpl
          sete
                  %r14b
          addq
                  %r14, %rax
  .LVL15:
                  %esi, %edx
          cmpl
          sete
                  %dl
          movzbl %dl, %edx
          addq
                  %rdx, %rax
  .LVL16:
  .LBE529:
                  $1, %rax
          cmpq
          jne
                  .L4164
          addq
                  $4, %rcx
          .loc 1 120 0 is_stmt 0 discriminator 2
                  $a+40, %rcx
          cmpq
                  .L5
          jne
  .LVL17:
  .L6:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• printf("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $.LC4, %edi
          movl
                  $0, %eax
          movl
          call
                  printf
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
                  $.LC4, %esi
          movl
                  $1, %edi
          movl
                  $0, %eax
          movl
          call
                  __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
```

• printf("helloworldn"); with gcc -g -03 produces:

```
.LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 104 0
         movl $.LC4, %esi
         movl $1, %edi
         xorl
                 %eax, %eax
         call
                  __printf_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movq
                  stdout(%rip), %rax
                  %rax, %rcx
         movq
                  $11, %edx
         movl
                  $1, %esi
         movl
                  $.LC4, %edi
         movl
         call
                  fwrite
          .loc 1 121 0
• L("helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
         movq stdout(%rip), %rcx
         movl
                 $11, %edx
                  $1, %esi
         movl
                  $.LC4, %edi
         movl
          call
                 fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
```

```
.loc 2 97 0
         movq stdout(%rip), %rcx
                 $11, %edx
         movl
         movl
                $1, %esi
                $.LC4, %edi
         movl
                  fwrite
         call
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl
                  $0, _dl_target(%rip)
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
         movl
               $0, _dl_target(%rip)
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _dl_target(%rip)
         movl
          .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl
                  $0, gi(%rip)
                  gi(%rip), %eax
         movl
                  $16, %eax
          andl
          testl %eax, %eax
                  .L5
          jе
          .loc 1 120 0 is_stmt 0 discriminator 1
                  stdout(%rip), %rax
                  %rax, %rcx
         movq
                  $11, %edx
         movl
                  $1, %esi
          movl
         movl
                  $.LC4, %edi
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
```

```
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
                  gi(%rip), %eax
          movl
                  $16, %al
          testb
          jе
                  .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
          movq stdout(%rip), %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
                  gi(%rip), %eax
          movl
          testb
                  $16, %al
                  .L1045
          jne
  .L5:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
                  $16, %eax
          andl
                  %eax, %eax
          testl
                  .L5
          jе
          .loc 1 120 0 is_stmt 0 discriminator 1
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
          movq
          movl
                  $11, %edx
          movl
                  $1, %esi
                  $.LC4, %edi
          movl
```

```
call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
                  .L5
          jе
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
          movq
                stdout(%rip), %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $-1, gi(%rip)
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
          jne
                  .L1045
  .L5:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -32(\%rbp), \%rax
          movq
          movq
                  %rax, %rcx
                  $11, %edx
          movl
          movl
                  $1, %esi
                  $.LC4, %edi
          movl
                  fwrite
          call
          .loc 1 121 0
```

```
• LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
         movq %rbx, %rcx
                $11, %edx
         movl
               $1, %esi
         movl
         movl $.LC4, %edi
               fwrite
          call
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 97 0
                 $11, %edx
         movl
                 %rbx, %rcx
         movq
                 $1, %esi
         movl
         movl
                 $.LC4, %edi
                 fwrite
          call
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -40(%rbp), %rax
         movq
         movl
                  $.LC4, %esi
                 %rax, %rdi
         movq
         movl
                  $0, %eax
                 L_buffer_printf
          call
          .loc 1 121 0
• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -01 produces:
         .loc 1 120 0
```

#NO_APP

```
$.LC4, %esi
          movl
                  %rbx, %rdi
          movq
          movl
                  $0, %eax
          call
                  L_buffer_printf
  .LVL6:
          .loc 1 121 0
• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O3 produces:
          .loc 1 120 0
 #NO_APP
                  $.LC4, %esi
          movl
                  %rbx, %rdi
          movq
                  %eax, %eax
          xorl
          call
                  L_buffer_printf
  .LVL279:
          .loc 1 121 0
• LHP(syslog,LOG_USER, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $.LC4, %esi
          movl
          movl
                  $8, %edi
                  $0, %eax
          movl
                  syslog
          call
          .loc 1 121 0
• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
          .loc 3 31 0
               $.LC4, %edx
          movl
                 $1, %esi
          movl
                $8, %edi
          movl
                  $0, %eax
          movl
          call
                __syslog_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
```

• LHP(syslog,LOG_USER, "helloworldn"); with gcc -g -O3 produces:

```
.LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 3 31 0
         movl $.LC4, %edx
         movl $1, %esi
         movl $8, %edi
         xorl %eax, %eax
         call __syslog_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
• asm(""); with gcc -g -00 produces:
          .loc 1 120 0
• asm(""); with gcc -g -01 produces:
          .loc 1 120 0
• asm(""); with gcc -g -03 produces:
          .loc 1 120 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
```

• asm("nop;nop;"); with gcc -g -00 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
        nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
         .loc 1 120 0
 #NO_APP
         movl $4, -56(%rbp)
         .loc 1 121 0
• i = 4; with gcc -g -01 produces:
         .loc 1 120 0
 #NO_APP
         movl $4, 24(%rsp)
         .loc 1 121 0
```

• i = 4; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$4, 24(%rsp) .loc 1 121 0

• gi = 11; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• gi = 11; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• gi = 11; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax movl %eax, -48(%rbp) .loc 1 121 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0 movss %xmm0, 28(%rsp) .loc 1 121 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, 28(%rsp)
.loc 1 121 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, gf(%rip) .loc 1 121 0

• i++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl -56(%rbp), %eax addl \$1, %eax movl %eax, -56(%rbp) .loc 1 121 0

• i++; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl 24(%rsp), %eax addl \$1, %eax movl %eax, 24(%rsp) .loc 1 121 0

• i++; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl 24(%rsp), %eax addl \$1, %eax movl %eax, 24(%rsp) .loc 1 121 0

• gi++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• gi++; with gcc -g -01 produces:

.loc 1 120 0
#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• gi++; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• j = a[i]; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl -60(%rbp), %eax
cltq
movl a(,%rax,4), %eax
movl %eax, -56(%rbp)
.loc 1 121 0

• j = a[i]; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl 20(%rsp), %eax
cltq
movl a(,%rax,4), %eax
movl %eax, 24(%rsp)
.loc 1 121 0

• j = a[i]; with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movslq 20(%rsp), %rax
          movl
                  a(,%rax,4), %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
                  $9, %eax
          cmpl
          jg
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
               $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L5:
          .loc 1 121 0 is_stmt 1
• assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
                  $9, %eax
          cmpl
          jg
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• assert(i \geq= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 121 0
```

• BSD_assert(i >= 10); with gcc -g -00 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
               $.LC4, %edx
          movl
                  $120, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  __BSD_assert
  .L5:
          .loc 1 121 0 is_stmt 1
• BSD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 121 0
• BSD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
                  $9, %eax
          cmpl
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movq
                  stderr(%rip), %rax
          movl
                  $.LC4, %r8d
                  $120, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
                  %rax, %rdi
          movq
                  $0, %eax
          movl
          call
                  fprintf
          movl
                  $1, %edi
          call
                  exit
  .L5:
          .loc 1 121 0 is_stmt 1
```

• TRAD_assert(i >= 10); with gcc -g -01 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  $.LC4, %r9d
          movl
                  $120, %r8d
          movl
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
          movl
                  $1, %esi
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
                  __fprintf_chk
          call
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
                  $1, %edi
          movl
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
                  $9, %eax
          cmpl
          jle
                  .L21
          .loc 1 121 0
• I(i \geq 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
          .loc 1 121 0 is_stmt 1
```

```
.loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
          jg
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(i \geq 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 121 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
• DI(i >= 10); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• DI(i \ge 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
```

• I(i \geq = 10); with gcc -g -01 produces:

produces:

```
#NO_APP
  .LBB2:
          .loc 1 120 0
                  $1, -3132(%rbp)
          movl
          movq
                  $str, -2096(%rbp)
                  .L5
          jmp
  .L8:
          .loc 1 120 0 is_stmt 0 discriminator 2
          call
                  __ctype_b_loc
                  (\%rax), \%rdx
          movq
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
          addq
                  %rax, %rax
          addq
                  %rdx, %rax
          movzwl (%rax), %eax
          movzwl %ax, %eax
                  $512, %eax
          andl
                  %eax, %eax
          testl
          jne
                  .L6
          .loc 1 120 0 discriminator 1
                  $0, -3132(%rbp)
          movl
                  .L7
          jmp
  .L6:
          .loc 1 120 0 discriminator 2
                  $1, -2096(%rbp)
          addq
  .L5:
          .loc 1 120 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl
                 (%rax), %eax
          testb
                  %al, %al
                  .L8
          jne
  .L7:
                  -3132(%rbp), %eax
          movl
  .LBE2:
          testl
                 %eax, %eax
                  .L9
          jne
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L9:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -01
 produces:
  .LVL5:
 #NO_APP
  .LBB269:
```

```
.loc 1 120 0
          movzbl str(%rip), %ebx
          testb
                  %bl, %bl
          jе
                  .L5
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
  .LVL7:
  .L7:
          .loc 1 120 0 is_stmt 0 discriminator 2
          movsbq %bl, %rbx
          testb
                  $2, 1(%rdx,%rbx,2)
          jе
                  .L6
          addq
                  $1, %rax
  .LVL8:
          movzbl (%rax), %ebx
                  %bl, %bl
          testb
                  .L7
          jne
  .LVL9:
  .L5:
  .LBE269:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -03
 produces:
  .LVL5:
 #NO_APP
  .LBB271:
          .loc 1 120 0
          movsbq str(%rip), %rbx
                  %bl, %bl
          testb
          jе
                  .L5
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
                  .L7
          jmp
  .LVL7:
          .p2align 4,,10
          .p2align 3
  .L2074:
          .loc 1 120 0 is_stmt 0 discriminator 2
          addq
                  $1, %rax
  .LVL8:
          movsbq (%rax), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
  .LVL9:
```

```
.L7:
                  $2, 1(%rdx,%rbx,2)
          testb
          jne
                  .L2074
  .LVL10:
  .LBE271:
          .loc 1 120 0
          movl
                  $120, %edx
                  $.LC10, %esi
          movl
                  $.LC11, %edi
          movl
                  _I_default_handler
          call
  .LVL11:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -00 produces:
 #NO_APP
  .LBB2:
          .loc 1 120 0
          movl
                  $1, -5188(%rbp)
                  $0, -5184(%rbp)
          movl
                  .L5
          jmp
  .L11:
  .LBB3:
          .loc 1 120 0 is_stmt 0 discriminator 2
                  $0, -2096(%rbp)
          movq
                  $0, -5180(%rbp)
          movl
                  .L6
          jmp
  .L8:
                  -5184(%rbp), %eax
          movl
          cltq
                  a(, %rax, 4), %edx
          movl
          movl
                  -5180(%rbp), %eax
          cltq
          movl
                  a(, %rax, 4), %eax
                  %eax, %edx
          cmpl
          jne
                  .L7
          .loc 1 120 0 discriminator 1
                  $1, -2096(%rbp)
          addq
  .L7:
          .loc 1 120 0 discriminator 2
                  $1, -5180(%rbp)
          addl
  .L6:
          .loc 1 120 0 discriminator 1
          cmpl
                  $9, -5180(%rbp)
          jle
                  .L8
          .loc 1 120 0 discriminator 3
                  -2096(%rbp), %rax
          movq
  .LBE3:
```

```
$1, %rax
          cmpq
                  .L9
          jе
          .loc 1 120 0 discriminator 1
          movl
                  $0, -5188(%rbp)
                  .L10
          jmp
  .L9:
          .loc 1 120 0 discriminator 2
          addl
                  $1, -5184(%rbp)
  .L5:
          .loc 1 120 0 discriminator 1
                  $9, -5184(%rbp)
          cmpl
          jle
                  .L11
  .L10:
          movl
                  -5188(%rbp), %eax
  .LBE2:
                  %eax, %eax
          testl
                  .L12
          jne
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
                  _I_default_handler
          call
  .L12:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
          .loc 1 120 0
                 $0, %ecx
          movl
  .LBE526:
          .loc 1 119 0
          movl
                  $0, %eax
          movl
                  $0, %edx
                  .L5
          jmp
  .LVL6:
  .L1295:
  .LBB528:
                  $0, %eax
          movl
  .LVL7:
                  $0, %edx
          movl
  .LVL8:
  .L5:
  .LBB527:
          .loc 1 120 0 discriminator 2
          movslq %ecx, %rdi
          movslq %eax, %rsi
                  a(,%rsi,4), %ebx
          movl
```

```
%ebx, a(,%rdi,4)
          cmpl
                  %sil
          sete
          movzbl %sil, %esi
          addq
                  %rsi, %rdx
  .LVL9:
          addl
                  $1, %eax
  .LVL10:
                  $9, %eax
          cmpl
          jle
                  .L5
  .LBE527:
          .loc 1 120 0 is_stmt 0 discriminator 3
                  $1, %rdx
          cmpq
          jne
                  .L8
          .loc 1 120 0 discriminator 2
          addl
                  $1, %ecx
  .LVL11:
                  $9, %ecx
          cmpl
                  .L1295
          jle
  .LVL12:
  .L1038:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -03 produces:
  .LVL5:
 #NO_APP
  .LBB528:
  .LBB529:
          .loc 1 120 0
                  a(%rip), %r13d
          movl
                  a+4(%rip), %r12d
          movl
                  $a, %ecx
          movl
          movl
                  a+8(%rip), %ebp
                  a+12(%rip), %ebx
          movl
                  a+16(%rip), %r11d
          movl
                  a+20(%rip), %r10d
          movl
                  a+24(%rip), %r9d
          movl
                  a+28(%rip), %r8d
          movl
                  a+32(%rip), %edi
          movl
                  a+36(%rip), %esi
          movl
  .LVL6:
  .L5:
          movl
                  (%rcx), %edx
          xorl
                  %eax, %eax
          {\tt cmpl}
                  %r13d, %edx
                  %al
          sete
  .LVL7:
          xorl
                  %r14d, %r14d
```

```
%r12d, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
.LVL8:
                %r14d, %r14d
       xorl
                %ebp, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
.LVL9:
                %r14d, %r14d
        xorl
                %ebx, %edx
        cmpl
        sete
                %r14b
                %r14, %rax
        addq
.LVL10:
        xorl
                %r14d, %r14d
                %r11d, %edx
        cmpl
        sete
                %r14b
                %r14, %rax
        addq
.LVL11:
        xorl
                %r14d, %r14d
                %r10d, %edx
        cmpl
                %r14b
        sete
        addq
                %r14, %rax
.LVL12:
        xorl
                %r14d, %r14d
                %r9d, %edx
        cmpl
        sete
                %r14b
                %r14, %rax
        addq
.LVL13:
                %r14d, %r14d
        xorl
        cmpl
                %r8d, %edx
                %r14b
        sete
                %r14, %rax
        addq
.LVL14:
        xorl
                %r14d, %r14d
                %edi, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
.LVL15:
                %esi, %edx
        cmpl
        sete
                %dl
                %dl, %edx
       movzbl
                %rdx, %rax
        addq
.LVL16:
.LBE529:
        cmpq
                $1, %rax
        jne
                .L4164
                $4, %rcx
        addq
        .loc 1 120 0 is_stmt 0 discriminator 2
        cmpq
                $a+40, %rcx
```

```
jne
                .L5
  .LVL17:
  .L6:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• printf("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $.LC4, %edi
         movl
                  $0, %eax
         movl
          call
                  printf
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
               $.LC4, %esi
         movl
                  $1, %edi
         movl
               $0, %eax
         movl
          call
                  __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 104 0
         movl $.LC4, %esi
                 $1, %edi
         movl
                 %eax, %eax
         xorl
         call
                __printf_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
```

 \bullet L("helloworldn"); with gcc -g -00 produces:

```
.loc 1 120 0
 #NO_APP
               stdout(%rip), %rax
         movq
         movq %rax, %rcx
         movl
              $11, %edx
              $1, %esi
         movl
               $.LC4, %edi
         movl
         call fwrite
         .loc 1 121 0
• L("helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
         .loc 2 97 0
         movq stdout(%rip), %rcx
               $11, %edx
         movl
         movl $1, %esi
         movl $.LC4, %edi
         call fwrite
  .LVL6:
  .LBE527:
  .LBE526:
         .loc 1 121 0
• L("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 2 97 0
         movq stdout(%rip), %rcx
         movl $11, %edx
         movl $1, %esi
         movl $.LC4, %edi
         call
                fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
         .loc 1 121 0
• DL("helloworldn"); with gcc -g -00 produces:
         .loc 1 120 0
 #NO_APP
                 $0, _dl_target(%rip)
         movl
         .loc 1 121 0
```

```
• DL("helloworldn"); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
         movl
                  $0, _dl_target(%rip)
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
         movl
                  $0, _dl_target(%rip)
          .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl
                  $0, gi(%rip)
                  gi(%rip), %eax
         movl
                  $16, %eax
          andl
          testl %eax, %eax
          jе
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
          movq stdout(%rip), %rax
                %rax, %rcx
         movq
         movl
                 $11, %edx
                 $1, %esi
         movl
                  $.LC4, %edi
         movl
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
                  $0, gi(%rip)
         movl
                  gi(%rip), %eax
         movl
          testb
                  $16, %al
         jе
                  .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
         movq
                  $11, %edx
         movl
                  $1, %esi
         movl
                  $.LC4, %edi
         movl
```

```
call
                   fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
           .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
           .loc 1 120 0
  #NO_APP
                   $0, gi(%rip)
          movl
          movl
                   gi(%rip), %eax
                   $16, %al
          testb
                   .L1045
          jne
  .L5:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
           .loc 1 120 0
  #NO_APP
                   $-1, gi(%rip)
          movl
          movl
                   gi(%rip), %eax
          \quad \text{andl} \quad
                   $16, %eax
                   %eax, %eax
          testl
          jе
                   .L5
          .loc 1 120 0 is_stmt 0 discriminator 1 \,
                   stdout(%rip), %rax
          movq
                   %rax, %rcx
          movq
                   $11, %edx
          movl
                   $1, %esi
          movl
                   $.LC4, %edi
          movl
          call
                   fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
           .loc 1 120 0
  #NO_APP
                   $-1, gi(%rip)
          movl
          movl
                   gi(%rip), %eax
          testb
                   $16, %al
          jе
                   .L5
  .LVL5:
  .LBB526:
  .LBB527:
           .loc 2 97 0 discriminator 1
```

```
stdout(%rip), %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
          movl
                  $.LC4, %edi
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
                  $16, %al
          testb
          jne
                  .L1045
  .L5:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -32(%rbp), %rax
          movq
                  %rax, %rcx
          movq
                  $11, %edx
          movl
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          call
                  fwrite
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
                %rbx, %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
          movl
                  $.LC4, %edi
          call
                 fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
```

• LHP(fprintf,log, "helloworldn"); with gcc -g -03 produces: .LVL277: #NO_APP .LBB1050: .LBB1051: .loc 2 97 0 movl \$11, %edx movq %rbx, %rcx \$1, %esi movl \$.LC4, %edi movl call fwrite .LVL278: .LBE1051: .LBE1050: .loc 1 121 0 • LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -OO produces: .loc 1 120 0 #NO_APP -40(%rbp), %rax movq movl \$.LC4, %esi %rax, %rdi movq **\$0,** %eax movl L_buffer_printf call .loc 1 121 0 \bullet LHP(L_buffer_printf,buf,"helloworldn"); with gcc $\,$ -g $\,$ -O1 produces: .loc 1 120 0 #NO_APP \$.LC4, %esi movl movq %rbx, %rdi \$0, %eax movl L_buffer_printf call .LVL6: .loc 1 121 0 \bullet LHP(L_buffer_printf,buf,"helloworldn"); with gcc $\,$ -g $\,$ -03 produces: .loc 1 120 0 #NO_APP \$.LC4, %esi movl movq %rbx, %rdi xorl %eax, %eax L_buffer_printf call .LVL279: .loc 1 121 0

• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -OO produces: .loc 1 120 0 #NO_APP \$.LC4, %esi movl \$8, %edi movl **\$0,** %eax movl call syslog .loc 1 121 0 \bullet LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces: .LVL5: #NO_APP .LBB526: .LBB527: .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h" .loc 3 31 0 \$.LC4, %edx movl movl \$1, %esi movl \$8, %edi movl \$0, %eax call __syslog_chk .LVL6: .LBE527: .LBE526: .loc 1 121 0 • LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O3 produces: .LVL277: #NO_APP .LBB1050: .LBB1051: .loc 3 31 0 movl \$.LC4, %edx movl \$1, %esi \$8, %edi movl xorl %eax, %eax call __syslog_chk .LVL278: .LBE1051: .LBE1050: .loc 1 121 0

• asm(""); with gcc -g -00 produces:

.loc 1 120 0

```
.loc 1 120 0
• asm(""); with gcc -g -03 produces:
          .loc 1 120 0
• asm("nop"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
```

 \bullet asm(""); with gcc -g -01 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;"); with gcc -g -00 produces:

```
• asm("nop;nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl
               $4, -56(%rbp)
          .loc 1 121 0
• i = 4; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
         movl
               $4, 24(%rsp)
         .loc 1 121 0
• i = 4; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
         movl $4, 24(%rsp)
         .loc 1 121 0
• gi = 11; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl $11, gi(%rip)
          .loc 1 121 0
• gi = 11; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
```

\$11, gi(%rip)

movl

.loc 1 121 0

• gi = 11; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, -48(%rbp)
.loc 1 121 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, 28(%rsp)
.loc 1 121 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, 28(%rsp) .loc 1 121 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
                  $0x41400000, gf(%rip)
          movl
          .loc 1 121 0
• i++; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          addl
                  $1, %eax
                  %eax, -56(%rbp)
          .loc 1 121 0
• i++; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          addl
                  $1, %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• i++; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          addl
                  $1, %eax
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• gi++; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  gi(%rip), %eax
          addl
                  $1, %eax
                  %eax, gi(%rip)
          movl
          .loc 1 121 0
• gi++; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  gi(%rip), %eax
                  $1, %eax
          addl
```

%eax, gi(%rip)

movl

.loc 1 121 0

```
• gi++; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  gi(%rip), %eax
          movl
          addl
                  $1, %eax
          movl
                  %eax, gi(%rip)
          .loc 1 121 0
• j = a[i]; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -60(%rbp), %eax
          movl
          cltq
          movl
                  a(, %rax, 4), %eax
                  %eax, -56(%rbp)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  20(%rsp), %eax
          movl
          cltq
                  a(, %rax, 4), %eax
          movl
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movslq 20(%rsp), %rax
                  a(, %rax, 4), %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          cmpl
                  $9, %eax
                   .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
```

.loc 1 121 0 is_stmt 1

.loc 1 120 0 #NO_APP 24(%rsp), %eax movl \$9, %eax cmpljg .L5 .loc 1 120 0 is_stmt 0 discriminator 1 \$120, %edx movl \$.LC4, %esi movl \$.LC5, %edi movl call _I_default_handler .LVL5: .L5: .loc 1 121 0 is_stmt 1 • assert(i >= 10); with gcc -g -03 produces: .loc 1 120 0 #NO_APP movl 24(%rsp), %eax cmpl\$9, %eax jle .L274 .L5: .loc 1 121 0 • BSD_assert(i >= 10); with gcc -g -00 produces: .loc 1 120 0 #NO_APP -56(%rbp), %eax movl cmpl \$9, %eax .L5 jg .loc 1 120 0 is_stmt 0 discriminator 1 movl \$.LC4, %edx movl \$120, %esi \$.LC5, %edi movl call __BSD_assert .L5: .loc 1 121 0 is_stmt 1 • BSD_assert(i >= 10); with gcc -g -01 produces: .loc 1 120 0 #NO_APP movl24(%rsp), %eax .loc 1 121 0

• assert(i >= 10); with gcc -g -O1 produces:

• BSD_assert(i >= 10); with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movq
                  stderr(%rip), %rax
          movl
                  $.LC4, %r8d
                  $120, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
                  %rax, %rdi
          movq
                  $0, %eax
          movl
          call
                  fprintf
                  $1, %edi
          movl
          call
                  exit
  .L5:
          .loc 1 121 0 is_stmt 1
• TRAD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
                  .L5
          jg
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                 $.LC4, %r9d
          movl
          movl
                  $120, %r8d
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
                  $1, %esi
          movl
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
          call
                  __fprintf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
```

```
$1, %edi
          movl
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
                  $9, %eax
          cmpl
          jle
                  .L21
          .loc 1 121 0
• I(i \geq 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
          .loc 1 121 0 is_stmt 1
• I(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
```

• I(i \geq 10); with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 121 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
• DI(i \ge 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• DI(i \ge 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
 produces:
 #NO_APP
  .LBB2:
          .loc 1 120 0
          movl $1, -3132(%rbp)
                  $str, -2096(%rbp)
          movq
          jmp
                  .L5
  .L8:
          .loc 1 120 0 is_stmt 0 discriminator 2
                  __ctype_b_loc
          call
                  (\%rax), \%rdx
          movq
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
          addq
                  %rax, %rax
                  %rdx, %rax
          addq
          movzwl (%rax), %eax
          movzwl %ax, %eax
```

```
$512, %eax
          andl
                  %eax, %eax
          testl
          jne
                  .L6
          .loc 1 120 0 discriminator 1
          movl
                  $0, -3132(%rbp)
                  .L7
          jmp
  .L6:
          .loc 1 120 0 discriminator 2
                  $1, -2096(%rbp)
          addq
  .L5:
          .loc 1 120 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl
                  (%rax), %eax
          testb
                  %al, %al
          jne
                  .L8
  .L7:
                  -3132(%rbp), %eax
          movl
  .LBE2:
                  %eax, %eax
          testl
          jne
                  .L9
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L9:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -01
 produces:
  .LVL5:
 #NO_APP
  .LBB269:
          .loc 1 120 0
          movzbl str(%rip), %ebx
          testb
                  %bl, %bl
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
                  $str, %eax
          movl
  .LVL7:
  .L7:
          .loc 1 120 0 is_stmt 0 discriminator 2
          movsbq %bl, %rbx
          testb
                  $2, 1(%rdx,%rbx,2)
          jе
                  .L6
                  $1, %rax
          addq
  .LVL8:
          movzbl (%rax), %ebx
```

```
%bl, %bl
          testb
                  .L7
          jne
  .LVL9:
  .L5:
  .LBE269:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -03
 produces:
  .LVL5:
 #NO_APP
  .LBB271:
          .loc 1 120 0
          movsbq str(%rip), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
                  .L7
          jmp
  .LVL7:
          .p2align 4,,10
          .p2align 3
  .L2074:
          .loc 1 120 0 is_stmt 0 discriminator 2
          addq
                  $1, %rax
  .LVL8:
          movsbq (%rax), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
  .LVL9:
  .L7:
                  $2, 1(%rdx,%rbx,2)
          testb
          jne
                  .L2074
  .LVL10:
  .LBE271:
          .loc 1 120 0
          movl
                  $120, %edx
                  $.LC10, %esi
          movl
                  $.LC11, %edi
          movl
                  _I_default_handler
          call
  .LVL11:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -00 produces:
```

```
#NO_APP
.LBB2:
        .loc 1 120 0
        movl
                $1, -5188(%rbp)
        movl
                $0, -5184(%rbp)
                .L5
        jmp
.L11:
.LBB3:
        .loc 1 120 0 is_stmt 0 discriminator 2
                $0, -2096(%rbp)
        movq
                $0, -5180(%rbp)
        movl
                .L6
        jmp
.L8:
        movl
                -5184(%rbp), %eax
        cltq
                a(, %rax, 4), %edx
        movl
                -5180(%rbp), %eax
        movl
        cltq
                a(, %rax, 4), %eax
        movl
        cmpl
                %eax, %edx
        jne
                .L7
        .loc 1 120 0 discriminator 1
        addq
                $1, -2096(%rbp)
.L7:
        .loc 1 120 0 discriminator 2
        addl
                $1, -5180(%rbp)
.L6:
        .loc 1 120 0 discriminator 1
                $9, -5180(%rbp)
        cmpl
        jle
                .L8
        .loc 1 120 0 discriminator 3
                -2096(%rbp), %rax
        movq
.LBE3:
                $1, %rax
        cmpq
        jе
                .L9
        .loc 1 120 0 discriminator 1
                $0, -5188(%rbp)
        movl
        jmp
                .L10
.L9:
        .loc 1 120 0 discriminator 2
        addl
                $1, -5184(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
                $9, -5184(%rbp)
        cmpl
        jle
                .L11
.L10:
        movl
                -5188(%rbp), %eax
.LBE2:
                %eax, %eax
        testl
        jne
                .L12
```

```
$120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L12:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
          .loc 1 120 0
               $0, %ecx
  .LBE526:
          .loc 1 119 0
                  $0, %eax
          movl
                  $0, %edx
          movl
          jmp
                  .L5
  .LVL6:
  .L1295:
  .LBB528:
          movl
                  $0, %eax
  .LVL7:
                  $0, %edx
          movl
  .LVL8:
  .L5:
  .LBB527:
          .loc 1 120 0 discriminator 2
          movslq %ecx, %rdi
          movslq %eax, %rsi
                  a(, %rsi, 4), %ebx
          movl
          cmpl
                  %ebx, a(, %rdi, 4)
          sete
                  %sil
          movzbl %sil, %esi
                  %rsi, %rdx
          addq
  .LVL9:
          addl
                  $1, %eax
  .LVL10:
          cmpl
                  $9, %eax
          jle
                  .L5
  .LBE527:
          .loc 1 120 0 is_stmt 0 discriminator 3
          cmpq
                  $1, %rdx
          jne
                  .L8
          .loc 1 120 0 discriminator 2
                  $1, %ecx
          addl
  .LVL11:
          cmpl
                  $9, %ecx
```

```
.L1295
          jle
  .LVL12:
  .L1038:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -03 produces:
  .LVL5:
 #NO_APP
  .LBB528:
  .LBB529:
          .loc 1 120 0
          movl
                  a(%rip), %r13d
                  a+4(%rip), %r12d
          movl
                  $a, %ecx
          movl
                  a+8(%rip), %ebp
          movl
                  a+12(%rip), %ebx
          movl
          movl
                  a+16(%rip), %r11d
                  a+20(%rip), %r10d
          movl
                  a+24(%rip), %r9d
          movl
                  a+28(%rip), %r8d
          movl
                  a+32(%rip), %edi
          movl
          movl
                  a+36(%rip), %esi
  .LVL6:
  .L5:
                  (%rcx), %edx
          movl
                  %eax, %eax
          xorl
          cmpl
                  %r13d, %edx
          sete
                  %al
  .LVL7:
                  %r14d, %r14d
          xorl
                  %r12d, %edx
          cmpl
          sete
                  %r14b
          addq
                  %r14, %rax
  .LVL8:
                  %r14d, %r14d
          xorl
                  %ebp, %edx
          cmpl
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL9:
                  %r14d, %r14d
          xorl
                  %ebx, %edx
          cmpl
          sete
                  %r14b
          addq
                  %r14, %rax
  .LVL10:
                  %r14d, %r14d
          xorl
                  %r11d, %edx
          cmpl
```

sete

%r14b

```
%r14, %rax
          addq
  .LVL11:
                  %r14d, %r14d
          xorl
          cmpl
                  %r10d, %edx
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL12:
                  %r14d, %r14d
          xorl
                  %r9d, %edx
          cmpl
                  %r14b
          sete
                  %r14, %rax
          addq
  .LVL13:
                  %r14d, %r14d
          xorl
          cmpl
                  %r8d, %edx
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL14:
                  %r14d, %r14d
          xorl
                  %edi, %edx
          cmpl
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL15:
          cmpl
                  %esi, %edx
                  %dl
          sete
          movzbl %dl, %edx
          addq
                  %rdx, %rax
  .LVL16:
  .LBE529:
                  $1, %rax
          cmpq
          jne
                  .L4164
          addq
                  $4, %rcx
          .loc 1 120 0 is_stmt 0 discriminator 2 \,
                  $a+40, %rcx
          cmpq
          jne
  .LVL17:
  .L6:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• printf("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $.LC4, %edi
          movl
          movl
                  $0, %eax
          call
                  printf
```

• printf("helloworldn"); with gcc -g -01 produces:

.loc 1 121 0

```
.LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
         movl $.LC4, %esi
         movl $1, %edi
         movl
               $0, %eax
         call
                __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 104 0
         movl $.LC4, %esi
                 $1, %edi
         movl
         xorl %eax, %eax
          call
                 __printf_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                 stdout(%rip), %rax
         movq
                 %rax, %rcx
         movq
                 $11, %edx
         movl
                 $1, %esi
         movl
         movl
                 $.LC4, %edi
          call
                 fwrite
          .loc 1 121 0
• L("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
```

```
stdout(%rip), %rcx
         movq
                 $11, %edx
         movl
                 $1, %esi
         movl
         movl
               $.LC4, %edi
         call fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 2 97 0
         movq stdout(%rip), %rcx
                $11, %edx
         movl
         movl $1, %esi
         movl
               $.LC4, %edi
         call
                 fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
         .loc 1 121 0
• DL("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl $0, _dl_target(%rip)
         .loc 1 121 0
• DL("helloworldn"); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
         movl
                $0, _dl_target(%rip)
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                 $0, _dl_target(%rip)
         .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
```

```
.loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
          andl
                  $16, %eax
                  %eax, %eax
          testl
          jе
          .loc 1 120 0 is_stmt 0 discriminator 1
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  $0, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
                  $16, %al
          testb
                  .L5
          jе
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
                  .L1045
          jne
  .L5:
          .loc 1 121 0
```

```
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $-1, gi(%rip)
                  gi(%rip), %eax
          movl
                  $16, %eax
          andl
                  %eax, %eax
          testl
                   .L5
          jе
          .loc 1 120 0 is_stmt 0 discriminator 1 \,
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  $-1, gi(%rip)
          movl
                  gi(%rip), %eax
          movl
                  $16, %al
          testb
          jе
                   .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
                  fwrite
          call
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $-1, gi(%rip)
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
```

```
.L1045
          jne
  .L5:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -OO produces:
          .loc 1 120 0
 #NO_APP
                  -32(\%rbp), \%rax
          movq
          movq
                  %rax, %rcx
          movl
                  $11, %edx
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
          movq %rbx, %rcx
          movl
                  $11, %edx
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 97 0
                 $11, %edx
          movl
                  %rbx, %rcx
          movq
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
```

 \bullet LHP(L_buffer_printf,buf,"helloworldn"); with gcc $\,$ -g $\,$ -00 produces:

```
.loc 1 120 0

#NO_APP

movq -40(%rbp), %rax
movl $.LC4, %esi
movq %rax, %rdi
movl $0, %eax
call L_buffer_printf
.loc 1 121 0
```

• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -O1 produces:

```
.loc 1 120 0

#NO_APP

movl $.LC4, %esi
movq %rbx, %rdi
movl $0, %eax
call L_buffer_printf
.LVL6:
.loc 1 121 0
```

• LHP(L_buffer_printf,buf, "helloworldn"); with gcc -g -03 produces:

```
.loc 1 120 0

#NO_APP

movl $.LC4, %esi
movq %rbx, %rdi
xorl %eax, %eax
call L_buffer_printf
.LVL279:
.loc 1 121 0
```

• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -OO produces:

```
.loc 1 120 0
#NO_APP

movl $.LC4, %esi
movl $8, %edi
movl $0, %eax
call syslog
.loc 1 121 0
```

• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:

```
.LVL5:
#NO_APP
.LBB526:
.LBB527:
```

```
.file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
          .loc 3 31 0
                  $.LC4, %edx
          movl
          movl
                 $1, %esi
          movl
                $8, %edi
                $0, %eax
          movl
          call
                  __syslog_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
\bullet LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 3 31 0
               $.LC4, %edx
          movl
          movl
                 $1, %esi
                 $8, %edi
          movl
                 %eax, %eax
          xorl
          call
                  __syslog_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
• assert(i >= 2); with gcc -g -O produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $1, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1 \,
          movl
                  $120, %edx
          movl
                  $.LC4, %esi
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• TRAD_assert(i >= 2); with gcc -g -O produces:
         .loc 1 120 0
 #NO_APP
```

```
movl
                  24(%rsp), %eax
                  $1, %eax
          cmpl
          jg
                  .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  $.LC4, %r9d
          movl
                  $120, %r8d
          movl
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
                  $1, %esi
          movl
                  stderr(%rip), %rdi
          movq
          movl
                  $0, %eax
          call
                  __fprintf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
          movl
                  $1, %edi
                  exit
          call
  .LVL7:
  .L5:
          .loc 1 121 0
• I(i >= 2); with gcc -g -O produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
                  $1, %eax
          cmpl
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• DI(i >= 2); with gcc -g -O produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
```

• I(A(int i=0, i!=10, i++, a[i]>=0)); with gcc -g -0 produces:

```
.LVL5:
 #NO_APP
  .LBB269:
         .loc 1 120 0
         .L5
                 $a+4, %eax
         movl
         movl
                 $a+40, %edx
  .LVL6:
  .L7:
          .loc 1 120 0 is_stmt 0 discriminator 2
         cmpl
                 $0, (%rax)
                  .L5
         js
         addq
                 $4, %rax
         cmpq
                 %rdx, %rax
         jne
                 .L7
  .LVL7:
  .L6:
  .LBE269:
          .loc 1 121 0 is_stmt 1
• d = now(); with gcc -g -O produces:
          .loc 1 120 0
 #NO_APP
         call
                 now
  .LVL5:
         movsd %xmm0, 24(%rsp)
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -O produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
                 $.LC4, %esi
         movl
                 $1, %edi
         movl
                $0, %eax
         movl
                 __printf_chk
         call
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -O produces:
```

.LVL5:

```
#NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
         movq stdout(%rip), %rcx
               $11, %edx
         movl
               $1, %esi
         movl
         movl
               $.LC4, %edi
          call
                 fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -O produces:
          .loc 1 120 0
 #NO_APP
          movl $0, _dl_target(%rip)
          .loc 1 121 0
• assert(i >= 2); with gcc -g -O produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $1, %eax
          jg
                 .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
                 $120, %edx
         movl
                 $.LC4, %esi
         movl
                  $.LC5, %edi
         movl
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• TRAD_assert(i >= 2); with gcc -g -O produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
         movl
                  $1, %eax
          cmpl
                  .L5
          jg
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
         movl $.LC4, %r9d
```

```
$120, %r8d
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
          movl
                  $1, %esi
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
          call
                  __fprintf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
                  $1, %edi
          movl
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• I(i \geq= 2); with gcc -g -0 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $1, %eax
                   .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• DI(i >= 2); with gcc -g -O produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• I(A(int i=0, i!=10, i++, a[i]>=0)); with gcc -g -0 produces:
  .LVL5:
 #NO_APP
  .LBB269:
          .loc 1 120 0
                  $0, a(%rip)
          cmpl
                   .L5
                  $a+4, %eax
          movl
```

movl

```
$a+40, %edx
          movl
  .LVL6:
  .L7:
          .loc 1 120 0 is_stmt 0 discriminator 2
          cmpl
                  $0, (%rax)
                  .L5
          js
                  $4, %rax
          addq
          cmpq
                  %rdx, %rax
                  .L7
          jne
  .LVL7:
  .L6:
  .LBE269:
          .loc 1 121 0 is_stmt 1
• d = now(); with gcc -g -O produces:
          .loc 1 120 0
 #NO_APP
          call
                  now
  .LVL5:
          movsd
                %xmm0, 24(%rsp)
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -O produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
          movl $.LC4, %esi
          movl
                 $1, %edi
          movl
                $0, %eax
          call
                 __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -0 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
          movq
                  stdout(%rip), %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
```

```
movl $.LC4, %edi
          call fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -O produces:
          .loc 1 120 0
 #NO_APP
          movl $0, _dl_target(%rip)
          .loc 1 121 0
• asm(""); with gcc -g -00 produces:
          .loc 1 120 0
• asm(""); with gcc -g -01 produces:
          .loc 1 120 0
• asm(""); with gcc -g -03 produces:
          .loc 1 120 0
• asm("nop"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
```

• asm("nop;nop;"); with gcc -g -00 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
        nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
         .loc 1 120 0
 #NO_APP
         movl $4, -56(%rbp)
         .loc 1 121 0
• i = 4; with gcc -g -01 produces:
         .loc 1 120 0
 #NO_APP
         movl $4, 24(%rsp)
         .loc 1 121 0
• i = 4; with gcc -g -03 produces:
```

.loc 1 120 0

#NO_APP

movl \$4, 24(%rsp) .loc 1 121 0

• gi = 11; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip) .loc 1 121 0

• gi = 11; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• gi = 11; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax movl %eax, -48(%rbp) .loc 1 121 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0 movss %xmm0, 28(%rsp) .loc 1 121 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, 28(%rsp)
.loc 1 121 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, gf(%rip) .loc 1 121 0

• i++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl -56(%rbp), %eax addl \$1, %eax movl %eax, -56(%rbp) .loc 1 121 0

• i++; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl 24(%rsp), %eax addl \$1, %eax movl %eax, 24(%rsp) .loc 1 121 0

• i++; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl 24(%rsp), %eax addl \$1, %eax movl %eax, 24(%rsp) .loc 1 121 0 • gi++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• gi++; with gcc -g -01 produces:

.loc 1 120 0
#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• gi++; with gcc -g -03 produces:

.loc 1 120 0
#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• j = a[i]; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl -60(%rbp), %eax
cltq
movl a(,%rax,4), %eax
movl %eax, -56(%rbp)
.loc 1 121 0

• j = a[i]; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl 20(%rsp), %eax
cltq
movl a(,%rax,4), %eax
movl %eax, 24(%rsp)
.loc 1 121 0

• j = a[i]; with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movslq 20(%rsp), %rax
          movl
                  a(,%rax,4), %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
                  $9, %eax
          cmpl
          jg
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
               $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .L5:
          .loc 1 121 0 is_stmt 1
• assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
                  $9, %eax
          cmpl
          jg
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• assert(i \geq= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 121 0
```

• BSD_assert(i >= 10); with gcc -g -00 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
               $.LC4, %edx
          movl
                  $120, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  __BSD_assert
  .L5:
          .loc 1 121 0 is_stmt 1
• BSD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 121 0
• BSD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
                  $9, %eax
          cmpl
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movq stderr(%rip), %rax
          movl
                  $.LC4, %r8d
                  $120, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
                  %rax, %rdi
          movq
                  $0, %eax
          movl
          call
                  fprintf
          movl
                  $1, %edi
          call
                  exit
  .L5:
          .loc 1 121 0 is_stmt 1
```

• TRAD_assert(i >= 10); with gcc -g -01 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  $.LC4, %r9d
          movl
                  $120, %r8d
          movl
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
          movl
                  $1, %esi
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
                  __fprintf_chk
          call
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
                  $1, %edi
          movl
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
                  $9, %eax
          cmpl
          jle
                  .L21
          .loc 1 121 0
• I(i \geq 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
          .loc 1 121 0 is_stmt 1
```

```
.loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
          jg
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(i \geq 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 121 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
• DI(i >= 10); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• DI(i \ge 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
```

• I(i \geq = 10); with gcc -g -01 produces:

produces:

```
#NO_APP
  .LBB2:
          .loc 1 120 0
                  $1, -3132(%rbp)
          movl
          movq
                  $str, -2096(%rbp)
                  .L5
          jmp
  .L8:
          .loc 1 120 0 is_stmt 0 discriminator 2
          call
                  __ctype_b_loc
                  (%rax), %rdx
          movq
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
          addq
                  %rax, %rax
          addq
                  %rdx, %rax
          movzwl (%rax), %eax
          movzwl %ax, %eax
                  $512, %eax
          andl
                  %eax, %eax
          testl
          jne
                  .L6
          .loc 1 120 0 discriminator 1
                  $0, -3132(%rbp)
          movl
                  .L7
          jmp
  .L6:
          .loc 1 120 0 discriminator 2
                  $1, -2096(%rbp)
          addq
  .L5:
          .loc 1 120 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl
                 (%rax), %eax
          testb
                  %al, %al
                  .L8
          jne
  .L7:
                  -3132(%rbp), %eax
          movl
  .LBE2:
          testl
                  %eax, %eax
                  .L9
          jne
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L9:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -01
 produces:
  .LVL5:
 #NO_APP
  .LBB269:
```

```
.loc 1 120 0
          movzbl str(%rip), %ebx
          testb
                  %bl, %bl
          jе
                  .L5
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
  .LVL7:
  .L7:
          .loc 1 120 0 is_stmt 0 discriminator 2
          movsbq %bl, %rbx
          testb
                  $2, 1(%rdx,%rbx,2)
          jе
                  .L6
          addq
                  $1, %rax
  .LVL8:
          movzbl (%rax), %ebx
                  %bl, %bl
          testb
                  .L7
          jne
  .LVL9:
  .L5:
  .LBE269:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -03
 produces:
  .LVL5:
 #NO_APP
  .LBB271:
          .loc 1 120 0
          movsbq str(%rip), %rbx
                  %bl, %bl
          testb
          jе
                  .L5
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
                  .L7
          jmp
  .LVL7:
          .p2align 4,,10
          .p2align 3
  .L2074:
          .loc 1 120 0 is_stmt 0 discriminator 2
          addq
                  $1, %rax
  .LVL8:
          movsbq (%rax), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
  .LVL9:
```

```
.L7:
                  $2, 1(%rdx,%rbx,2)
          testb
          jne
                  .L2074
  .LVL10:
  .LBE271:
          .loc 1 120 0
          movl
                  $120, %edx
                  $.LC10, %esi
          movl
                  $.LC11, %edi
          movl
                  _I_default_handler
          call
  .LVL11:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -00 produces:
 #NO_APP
  .LBB2:
          .loc 1 120 0
          movl
                  $1, -5188(%rbp)
                  $0, -5184(%rbp)
          movl
                  .L5
          jmp
  .L11:
  .LBB3:
          .loc 1 120 0 is_stmt 0 discriminator 2
                  $0, -2096(%rbp)
          movq
                  $0, -5180(%rbp)
          movl
                  .L6
          jmp
  .L8:
                  -5184(%rbp), %eax
          movl
          cltq
                  a(, %rax, 4), %edx
          movl
          movl
                  -5180(%rbp), %eax
          cltq
          movl
                  a(, %rax, 4), %eax
                  %eax, %edx
          cmpl
          jne
                  .L7
          .loc 1 120 0 discriminator 1
                  $1, -2096(%rbp)
          addq
  .L7:
          .loc 1 120 0 discriminator 2
                  $1, -5180(%rbp)
          addl
  .L6:
          .loc 1 120 0 discriminator 1
          cmpl
                  $9, -5180(%rbp)
          jle
                  .L8
          .loc 1 120 0 discriminator 3
                  -2096(%rbp), %rax
          movq
  .LBE3:
```

```
$1, %rax
          cmpq
                  .L9
          jе
          .loc 1 120 0 discriminator 1
          movl
                  $0, -5188(%rbp)
                  .L10
          jmp
  .L9:
          .loc 1 120 0 discriminator 2
          addl
                  $1, -5184(%rbp)
  .L5:
          .loc 1 120 0 discriminator 1
                  $9, -5184(%rbp)
          cmpl
          jle
                  .L11
  .L10:
          movl
                  -5188(%rbp), %eax
  .LBE2:
                  %eax, %eax
          testl
                  .L12
          jne
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
                  _I_default_handler
          call
  .L12:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
          .loc 1 120 0
                 $0, %ecx
          movl
  .LBE526:
          .loc 1 119 0
          movl
                  $0, %eax
          movl
                  $0, %edx
                  .L5
          jmp
  .LVL6:
  .L1295:
  .LBB528:
                  $0, %eax
          movl
  .LVL7:
                  $0, %edx
          movl
  .LVL8:
  .L5:
  .LBB527:
          .loc 1 120 0 discriminator 2
          movslq %ecx, %rdi
          movslq %eax, %rsi
          movl
                  a(, %rsi, 4), %ebx
```

```
%ebx, a(,%rdi,4)
          cmpl
                  %sil
          sete
          movzbl %sil, %esi
          addq
                  %rsi, %rdx
  .LVL9:
          addl
                  $1, %eax
  .LVL10:
                  $9, %eax
          cmpl
          jle
                  .L5
  .LBE527:
          .loc 1 120 0 is_stmt 0 discriminator 3
                  $1, %rdx
          cmpq
                  .L8
          jne
          .loc 1 120 0 discriminator 2
          addl
                  $1, %ecx
  .LVL11:
                  $9, %ecx
          cmpl
                  .L1295
          jle
  .LVL12:
  .L1038:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -03 produces:
  .LVL5:
 #NO_APP
  .LBB528:
  .LBB529:
          .loc 1 120 0
                  a(%rip), %r13d
          movl
                  a+4(%rip), %r12d
          movl
                  $a, %ecx
          movl
          movl
                  a+8(%rip), %ebp
                  a+12(%rip), %ebx
          movl
                  a+16(%rip), %r11d
          movl
                  a+20(%rip), %r10d
          movl
                  a+24(%rip), %r9d
          movl
                  a+28(%rip), %r8d
          movl
                  a+32(%rip), %edi
          movl
                  a+36(%rip), %esi
          movl
  .LVL6:
  .L5:
          movl
                  (%rcx), %edx
          xorl
                  %eax, %eax
          {\tt cmpl}
                  %r13d, %edx
                  %al
          sete
  .LVL7:
          xorl
                  %r14d, %r14d
```

```
%r12d, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
.LVL8:
                %r14d, %r14d
       xorl
                %ebp, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
.LVL9:
                %r14d, %r14d
        xorl
                %ebx, %edx
        cmpl
        sete
                %r14b
                %r14, %rax
        addq
.LVL10:
        xorl
                %r14d, %r14d
                %r11d, %edx
        cmpl
        sete
                %r14b
                %r14, %rax
        addq
.LVL11:
        xorl
                %r14d, %r14d
                %r10d, %edx
        cmpl
                %r14b
        sete
        addq
                %r14, %rax
.LVL12:
        xorl
                %r14d, %r14d
                %r9d, %edx
        cmpl
        sete
                %r14b
                %r14, %rax
        addq
.LVL13:
                %r14d, %r14d
        xorl
        cmpl
                %r8d, %edx
                %r14b
        sete
                %r14, %rax
        addq
.LVL14:
        xorl
                %r14d, %r14d
                %edi, %edx
        cmpl
                %r14b
        sete
                %r14, %rax
        addq
.LVL15:
                %esi, %edx
        cmpl
        sete
                %dl
                %dl, %edx
       movzbl
                %rdx, %rax
        addq
.LVL16:
.LBE529:
        cmpq
                $1, %rax
        jne
                .L4164
                $4, %rcx
        addq
        .loc 1 120 0 is_stmt 0 discriminator 2
        cmpq
                $a+40, %rcx
```

```
jne
                .L5
  .LVL17:
  .L6:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• printf("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $.LC4, %edi
         movl
                  $0, %eax
         movl
          call
                  printf
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
               $.LC4, %esi
         movl
                  $1, %edi
         movl
                $0, %eax
         movl
          call
                  __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 104 0
         movl $.LC4, %esi
                 $1, %edi
         movl
                 %eax, %eax
         xorl
         call
                __printf_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
```

 \bullet L("helloworldn"); with gcc -g -00 produces:

```
.loc 1 120 0
 #NO_APP
               stdout(%rip), %rax
         movq
         movq %rax, %rcx
         movl
              $11, %edx
              $1, %esi
         movl
               $.LC4, %edi
         movl
         call fwrite
         .loc 1 121 0
• L("helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
         .loc 2 97 0
         movq stdout(%rip), %rcx
               $11, %edx
         movl
         movl $1, %esi
         movl $.LC4, %edi
         call fwrite
  .LVL6:
  .LBE527:
  .LBE526:
         .loc 1 121 0
• L("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 2 97 0
         movq stdout(%rip), %rcx
         movl $11, %edx
         movl $1, %esi
         movl $.LC4, %edi
         call
                fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
         .loc 1 121 0
• DL("helloworldn"); with gcc -g -00 produces:
         .loc 1 120 0
 #NO_APP
                 $0, _dl_target(%rip)
         movl
         .loc 1 121 0
```

```
• DL("helloworldn"); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _dl_target(%rip)
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _dl_target(%rip)
          .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
                  gi(%rip), %eax
          movl
                  $16, %eax
          andl
          testl %eax, %eax
          jе
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
          movq stdout(%rip), %rax
                %rax, %rcx
          movq
          movl
                 $11, %edx
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
          movl
          testb
                  $16, %al
          jе
                  .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
```

```
call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  $0, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
                  $16, %al
          testb
                  .L1045
          jne
  .L5:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
          andl
                  $16, %eax
                  %eax, %eax
          testl
          jе
                   .L5
          .loc 1 120 0 is_stmt 0 discriminator 1 \,
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
          jе
                   .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
```

```
stdout(%rip), %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
          movl
                  $.LC4, %edi
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  $-1, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
                  $16, %al
          testb
          jne
                  .L1045
  .L5:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -32(%rbp), %rax
          movq
                  %rax, %rcx
          movq
                  $11, %edx
          movl
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          call
                  fwrite
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
                %rbx, %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
          movl
                  $.LC4, %edi
          call
                  fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
```

• LHP(fprintf,log,"helloworldn"); with gcc -g -03 produces: .LVL277: #NO_APP .LBB1050: .LBB1051: .loc 2 97 0 movl \$11, %edx movq %rbx, %rcx \$1, %esi movl \$.LC4, %edi movl call fwrite .LVL278: .LBE1051: .LBE1050: .loc 1 121 0 • LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -OO produces: .loc 1 120 0 #NO_APP -40(%rbp), %rax movq movl \$.LC4, %esi %rax, %rdi movq **\$0,** %eax movl L_buffer_printf call .loc 1 121 0 \bullet LHP(L_buffer_printf,buf,"helloworldn"); with gcc $\,$ -g $\,$ -O1 produces: .loc 1 120 0 #NO_APP \$.LC4, %esi movl movq %rbx, %rdi \$0, %eax movl L_buffer_printf call .LVL6: .loc 1 121 0 • LHP(L_buffer_printf,buf, "helloworldn"); with gcc -g -03 produces: .loc 1 120 0 #NO_APP \$.LC4, %esi movl movq %rbx, %rdi xorl %eax, %eax L_buffer_printf call .LVL279: .loc 1 121 0

• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -OO produces: .loc 1 120 0 #NO_APP \$.LC4, %esi movl \$8, %edi movl **\$0,** %eax movl call syslog .loc 1 121 0 \bullet LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces: .LVL5: #NO_APP .LBB526: .LBB527: .file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h" .loc 3 31 0 \$.LC4, %edx movl movl \$1, %esi movl \$8, %edi movl \$0, %eax call __syslog_chk .LVL6: .LBE527: .LBE526: .loc 1 121 0 • LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O3 produces: .LVL277: #NO_APP .LBB1050: .LBB1051: .loc 3 31 0 movl \$.LC4, %edx movl \$1, %esi \$8, %edi movl xorl %eax, %eax call __syslog_chk .LVL278: .LBE1051: .LBE1050: .loc 1 121 0

• asm(""); with gcc -g -00 produces:

.loc 1 120 0

```
.loc 1 120 0
• asm(""); with gcc -g -03 produces:
          .loc 1 120 0
• asm("nop"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop
 # 0 "" 2
• asm("nop"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
```

 \bullet asm(""); with gcc -g -01 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
        nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;"); with gcc -g -00 produces:

```
• asm("nop;nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl
               $4, -56(%rbp)
          .loc 1 121 0
• i = 4; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
         movl
               $4, 24(%rsp)
         .loc 1 121 0
• i = 4; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
         movl $4, 24(%rsp)
         .loc 1 121 0
• gi = 11; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl $11, gi(%rip)
          .loc 1 121 0
• gi = 11; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
```

\$11, gi(%rip)

movl

.loc 1 121 0

• gi = 11; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, -48(%rbp)
.loc 1 121 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0 movss %xmm0, 28(%rsp) .loc 1 121 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, 28(%rsp) .loc 1 121 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
                  $0x41400000, gf(%rip)
          movl
          .loc 1 121 0
• i++; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          addl
                  $1, %eax
                  %eax, -56(%rbp)
          .loc 1 121 0
• i++; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          addl
                  $1, %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• i++; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          addl
                  $1, %eax
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• gi++; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  gi(%rip), %eax
          addl
                  $1, %eax
                  %eax, gi(%rip)
          movl
          .loc 1 121 0
• gi++; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  gi(%rip), %eax
                  $1, %eax
          addl
```

%eax, gi(%rip)

movl

.loc 1 121 0

```
• gi++; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  gi(%rip), %eax
          movl
          addl
                  $1, %eax
          movl
                  %eax, gi(%rip)
          .loc 1 121 0
• j = a[i]; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -60(%rbp), %eax
          movl
          cltq
          movl
                  a(, %rax, 4), %eax
                  %eax, -56(%rbp)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  20(%rsp), %eax
          movl
          cltq
                  a(, %rax, 4), %eax
          movl
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movslq 20(%rsp), %rax
                  a(, %rax, 4), %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          cmpl
                  $9, %eax
                   .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
```

.loc 1 121 0 is_stmt 1

• assert(i >= 10); with gcc -g -O1 produces: .loc 1 120 0 #NO_APP 24(%rsp), %eax movl \$9, %eax cmpljg .L5 .loc 1 120 0 is_stmt 0 discriminator 1 \$120, %edx movl \$.LC4, %esi movl \$.LC5, %edi movl call _I_default_handler .LVL5: .L5: .loc 1 121 0 is_stmt 1 • assert(i >= 10); with gcc -g -03 produces: .loc 1 120 0 #NO_APP movl 24(%rsp), %eax cmpl\$9, %eax jle .L274 .L5: .loc 1 121 0 • BSD_assert(i >= 10); with gcc -g -00 produces: .loc 1 120 0 #NO_APP -56(%rbp), %eax movl cmpl \$9, %eax .L5 jg .loc 1 120 0 is_stmt 0 discriminator 1 movl \$.LC4, %edx movl \$120, %esi \$.LC5, %edi movl call __BSD_assert .L5: .loc 1 121 0 is_stmt 1 • BSD_assert(i >= 10); with gcc -g -01 produces: .loc 1 120 0 #NO_APP movl24(%rsp), %eax .loc 1 121 0

• BSD_assert(i >= 10); with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movq
                  stderr(%rip), %rax
          movl
                  $.LC4, %r8d
                  $120, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
                  %rax, %rdi
          movq
                  $0, %eax
          movl
          call
                  fprintf
                  $1, %edi
          movl
          call
                  exit
  .L5:
          .loc 1 121 0 is_stmt 1
• TRAD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
                  .L5
          jg
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1 \,
                 $.LC4, %r9d
          movl
          movl
                  $120, %r8d
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
                  $1, %esi
          movl
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
          call
                  __fprintf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
```

```
$1, %edi
          movl
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
                  $9, %eax
          cmpl
          jle
                  .L21
          .loc 1 121 0
• I(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L5:
          .loc 1 121 0 is_stmt 1
• I(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
```

• I(i \geq 10); with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 121 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                 $0, _di_target(%rip)
          .loc 1 121 0
• DI(i \geq= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• assert(i >= 10); with gcc -g -0 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
          jg
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• BSD_assert(i >= 10); with gcc -g -O produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          .loc 1 121 0
```

• TRAD_assert(i \geq 10); with gcc -g -0 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  $.LC4, %r9d
          movl
                  $120, %r8d
          movl
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
          movl
                  $1, %esi
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
                  __fprintf_chk
          call
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
                $1, %edi
          movl
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• I(i \geq 10); with gcc -g -0 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
          jg
                  .L5
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
                  _I_default_handler
          call
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
• DI(i >= 10); with gcc -g -O produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
```

```
asm(""); with gcc -g -00 produces:
.loc 1 120 0
asm(""); with gcc -g -01 produces:
.loc 1 120 0
asm(""); with gcc -g -03 produces:
.loc 1 120 0
```

• asm("nop"); with gcc -g -00 produces:

• asm("nop"); with gcc -g -01 produces:

• asm("nop"); with gcc -g -03 produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
nop
# 0 "" 2
```

• asm("nop;nop;"); with gcc -g -00 produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
nop;nop;
# 0 "" 2
```

• asm("nop;nop;"); with gcc -g -01 produces:

```
.loc 1 120 0
# 120 "tmp.c" 1
nop;nop;
# 0 "" 2
```

• asm("nop;nop;"); with gcc -g -03 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -00 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -01 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
```

• asm("nop;nop;nop;nop;nop;"); with gcc -g -00 produces:

```
.loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -01 produces:
         .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• asm("nop;nop;nop;nop;nop;"); with gcc -g -03 produces:
          .loc 1 120 0
 # 120 "tmp.c" 1
         nop;nop;nop;nop;
 # 0 "" 2
• i = 4; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl $4, -56(%rbp)
         .loc 1 121 0
• i = 4; with gcc -g -01 produces:
         .loc 1 120 0
 #NO_APP
         movl
               $4, 24(%rsp)
          .loc 1 121 0
• i = 4; with gcc -g -03 produces:
         .loc 1 120 0
 #NO_APP
               $4, 24(%rsp)
         movl
         .loc 1 121 0
• gi = 11; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl $11, gi(%rip)
         .loc 1 121 0
```

• gi = 11; with gcc - g - 01 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• gi = 11; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$11, gi(%rip)
.loc 1 121 0

• f = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, -48(%rbp)
.loc 1 121 0

• f = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0 movss %xmm0, 28(%rsp) .loc 1 121 0

• f = 12.0; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl \$0x41400000, 28(%rsp) .loc 1 121 0

• gf = 12.0; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl .LC4(%rip), %eax
movl %eax, gf(%rip)
.loc 1 121 0

• gf = 12.0; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movss .LC4(%rip), %xmm0
movss %xmm0, gf(%rip)
.loc 1 121 0

gf = 12.0; with gcc -g -03 produces:
.loc 1 120 0
#NO_APP

movl \$0x41400000, gf(%rip)
.loc 1 121 0
i++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl -56(%rbp), %eax
addl \$1, %eax
movl %eax, -56(%rbp)
.loc 1 121 0

• i++; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl 24(%rsp), %eax
addl \$1, %eax
movl %eax, 24(%rsp)
.loc 1 121 0

• i++; with gcc -g -03 produces:

.loc 1 120 0

#NO_APP

movl 24(%rsp), %eax
addl \$1, %eax
movl %eax, 24(%rsp)
.loc 1 121 0

• gi++; with gcc -g -00 produces:

.loc 1 120 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

• gi++; with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl gi(%rip), %eax
addl \$1, %eax
movl %eax, gi(%rip)
.loc 1 121 0

```
• gi++; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  gi(%rip), %eax
          movl
          addl
                  $1, %eax
          movl
                  %eax, gi(%rip)
          .loc 1 121 0
• j = a[i]; with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -60(%rbp), %eax
          movl
          cltq
          movl
                  a(, %rax, 4), %eax
                  %eax, -56(%rbp)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  20(%rsp), %eax
          movl
          cltq
                  a(, %rax, 4), %eax
          movl
                  %eax, 24(%rsp)
          movl
          .loc 1 121 0
• j = a[i]; with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movslq 20(%rsp), %rax
                  a(, %rax, 4), %eax
          movl
                  %eax, 24(%rsp)
          .loc 1 121 0
• assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  -56(%rbp), %eax
          movl
          cmpl
                  $9, %eax
                   .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
```

.loc 1 121 0 is_stmt 1

.loc 1 120 0 #NO_APP 24(%rsp), %eax movl \$9, %eax cmpljg .L5 .loc 1 120 0 is_stmt 0 discriminator 1 \$120, %edx movl \$.LC4, %esi movl \$.LC5, %edi movl call _I_default_handler .LVL5: .L5: .loc 1 121 0 is_stmt 1 • assert(i >= 10); with gcc -g -03 produces: .loc 1 120 0 #NO_APP movl 24(%rsp), %eax cmpl\$9, %eax jle .L274 .L5: .loc 1 121 0 • BSD_assert(i >= 10); with gcc -g -00 produces: .loc 1 120 0 #NO_APP -56(%rbp), %eax movl cmpl \$9, %eax .L5 jg .loc 1 120 0 is_stmt 0 discriminator 1 movl \$.LC4, %edx movl \$120, %esi \$.LC5, %edi movl call __BSD_assert .L5: .loc 1 121 0 is_stmt 1 • BSD_assert(i >= 10); with gcc -g -01 produces: .loc 1 120 0 #NO_APP movl24(%rsp), %eax .loc 1 121 0

• assert(i >= 10); with gcc -g -O1 produces:

• BSD_assert(i >= 10); with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movq
                  stderr(%rip), %rax
          movl
                  $.LC4, %r8d
                  $120, %ecx
          movl
                  $.LC5, %edx
          movl
                  $.LC6, %esi
          movl
                  %rax, %rdi
          movq
                  $0, %eax
          movl
          call
                  fprintf
                  $1, %edi
          movl
          call
                  exit
  .L5:
          .loc 1 121 0 is_stmt 1
• TRAD_assert(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
          cmpl
                  $9, %eax
                  .L5
          jg
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1 \,
                 $.LC4, %r9d
          movl
          movl
                  $120, %r8d
                  $.LC5, %ecx
          movl
                  $.LC6, %edx
          movl
                  $1, %esi
          movl
                  stderr(%rip), %rdi
          movq
                  $0, %eax
          movl
          call
                  __fprintf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 120 0 discriminator 1
```

```
$1, %edi
          movl
          call
                  exit
  .LVL7:
  .L5:
          .loc 1 121 0
• TRAD_assert(i >= 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                  24(%rsp), %eax
          movl
                  $9, %eax
          cmpl
          jle
                  .L21
          .loc 1 121 0
• I(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  -56(%rbp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
          movl
                  $120, %edx
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L5:
          .loc 1 121 0 is_stmt 1
• I(i >= 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L5
          jg
          .loc 1 120 0 is_stmt 0 discriminator 1
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
                  _I_default_handler
          call
  .LVL5:
  .L5:
          .loc 1 121 0 is_stmt 1
```

• I(i \geq 10); with gcc -g -03 produces:

```
.loc 1 120 0
 #NO_APP
          movl
                  24(%rsp), %eax
          cmpl
                  $9, %eax
                  .L274
          jle
  .L5:
          .loc 1 121 0
• DI(i >= 10); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $0, _di_target(%rip)
          movl
          .loc 1 121 0
• DI(i \ge 10); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• DI(i \ge 10); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, _di_target(%rip)
          .loc 1 121 0
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -00
 produces:
 #NO_APP
  .LBB2:
          .loc 1 120 0
          movl $1, -3132(%rbp)
                  $str, -2096(%rbp)
          movq
          jmp
                  .L5
  .L8:
          .loc 1 120 0 is_stmt 0 discriminator 2
                  __ctype_b_loc
          call
                  (\%rax), \%rdx
          movq
                  -2096(%rbp), %rax
          movq
          movzbl (%rax), %eax
          movsbq %al, %rax
          addq
                  %rax, %rax
          addq
                  %rdx, %rax
          movzwl (%rax), %eax
          movzwl %ax, %eax
```

```
$512, %eax
          andl
                  %eax, %eax
          testl
          jne
                  .L6
          .loc 1 120 0 discriminator 1
          movl
                  $0, -3132(%rbp)
                  .L7
          jmp
  .L6:
          .loc 1 120 0 discriminator 2
                  $1, -2096(%rbp)
          addq
  .L5:
          .loc 1 120 0 discriminator 1
                  -2096(%rbp), %rax
          movq
          movzbl
                  (%rax), %eax
          testb
                  %al, %al
          jne
                  .L8
  .L7:
                  -3132(%rbp), %eax
          movl
  .LBE2:
                  %eax, %eax
          testl
          jne
                  .L9
                  $120, %edx
          movl
                  $.LC4, %esi
          movl
          movl
                  $.LC5, %edi
          call
                  _I_default_handler
  .L9:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -01
 produces:
  .LVL5:
 #NO_APP
  .LBB269:
          .loc 1 120 0
          movzbl str(%rip), %ebx
          testb
                  %bl, %bl
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
                  $str, %eax
          movl
  .LVL7:
  .L7:
          .loc 1 120 0 is_stmt 0 discriminator 2
          movsbq %bl, %rbx
          testb
                  $2, 1(%rdx,%rbx,2)
          jе
                  .L6
                  $1, %rax
          addq
  .LVL8:
          movzbl (%rax), %ebx
```

```
.LVL9:
  .L5:
  .LBE269:
          .loc 1 121 0 is_stmt 1
• I(A(char *p = str, *p != '\0', p++, islower(*p))); with gcc -g -03
 produces:
  .LVL5:
 #NO_APP
  .LBB271:
          .loc 1 120 0
          movsbq str(%rip), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
          call
                  __ctype_b_loc
  .LVL6:
                  (%rax), %rdx
          movq
          movl
                  $str, %eax
                  .L7
          jmp
  .LVL7:
          .p2align 4,,10
          .p2align 3
  .L2074:
          .loc 1 120 0 is_stmt 0 discriminator 2
          addq
                  $1, %rax
  .LVL8:
          movsbq (%rax), %rbx
                  %bl, %bl
          testb
                  .L5
          jе
  .LVL9:
  .L7:
                  $2, 1(%rdx,%rbx,2)
          testb
          jne
                  .L2074
  .LVL10:
  .LBE271:
          .loc 1 120 0
          movl
                  $120, %edx
                  $.LC10, %esi
          movl
                  $.LC11, %edi
          movl
                  _I_default_handler
          call
  .LVL11:
  .L5:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -00 produces:
```

%bl, %bl

.L7

testb jne

```
#NO_APP
.LBB2:
        .loc 1 120 0
        movl
                $1, -5188(%rbp)
        movl
                $0, -5184(%rbp)
                .L5
        jmp
.L11:
.LBB3:
        .loc 1 120 0 is_stmt 0 discriminator 2
                $0, -2096(%rbp)
        movq
                $0, -5180(%rbp)
        movl
                .L6
        jmp
.L8:
        movl
                -5184(%rbp), %eax
        cltq
                a(, %rax, 4), %edx
        movl
                -5180(%rbp), %eax
        movl
        cltq
                a(, %rax, 4), %eax
        movl
        cmpl
                %eax, %edx
                .L7
        jne
        .loc 1 120 0 discriminator 1
        addq
                $1, -2096(%rbp)
.L7:
        .loc 1 120 0 discriminator 2
        addl
                $1, -5180(%rbp)
.L6:
        .loc 1 120 0 discriminator 1
                $9, -5180(%rbp)
        cmpl
        jle
                .L8
        .loc 1 120 0 discriminator 3
                -2096(%rbp), %rax
        movq
.LBE3:
                $1, %rax
        cmpq
        jе
                .L9
        .loc 1 120 0 discriminator 1
                $0, -5188(%rbp)
        movl
        jmp
                .L10
.L9:
        .loc 1 120 0 discriminator 2
        addl
                $1, -5184(%rbp)
.L5:
        .loc 1 120 0 discriminator 1
                $9, -5184(%rbp)
        cmpl
        jle
                .L11
.L10:
        movl
                -5188(%rbp), %eax
.LBE2:
                %eax, %eax
        testl
        jne
                .L12
```

```
$120, %edx
          movl
                  $.LC4, %esi
          movl
                  $.LC5, %edi
          movl
          call
                  _I_default_handler
  .L12:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
          .loc 1 120 0
               $0, %ecx
  .LBE526:
          .loc 1 119 0
                  $0, %eax
          movl
                  $0, %edx
          movl
          jmp
                  .L5
  .LVL6:
  .L1295:
  .LBB528:
          movl
                  $0, %eax
  .LVL7:
                  $0, %edx
          movl
  .LVL8:
  .L5:
  .LBB527:
          .loc 1 120 0 discriminator 2
          movslq %ecx, %rdi
          movslq %eax, %rsi
                  a(, %rsi, 4), %ebx
          movl
          cmpl
                  %ebx, a(, %rdi, 4)
          sete
                  %sil
          movzbl %sil, %esi
                  %rsi, %rdx
          addq
  .LVL9:
          addl
                  $1, %eax
  .LVL10:
          cmpl
                  $9, %eax
          jle
                  .L5
  .LBE527:
          .loc 1 120 0 is_stmt 0 discriminator 3
          cmpq
                  $1, %rdx
          jne
                  .L8
          .loc 1 120 0 discriminator 2
                  $1, %ecx
          addl
  .LVL11:
          cmpl
                  $9, %ecx
```

```
.L1295
          jle
  .LVL12:
  .L1038:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• I(A(int i = 0, i < 10, i++, E1(int j = 0, j < 10, j++, a[i] == a[j])));
 with gcc -g -03 produces:
  .LVL5:
 #NO_APP
  .LBB528:
  .LBB529:
          .loc 1 120 0
          movl
                  a(%rip), %r13d
                  a+4(%rip), %r12d
          movl
                  $a, %ecx
          movl
                  a+8(%rip), %ebp
          movl
                  a+12(%rip), %ebx
          movl
          movl
                  a+16(%rip), %r11d
                  a+20(%rip), %r10d
          movl
                  a+24(%rip), %r9d
          movl
                  a+28(%rip), %r8d
          movl
                  a+32(%rip), %edi
          movl
          movl
                  a+36(%rip), %esi
  .LVL6:
  .L5:
                  (%rcx), %edx
          movl
                  %eax, %eax
          xorl
          cmpl
                  %r13d, %edx
          sete
                  %al
  .LVL7:
                  %r14d, %r14d
          xorl
                  %r12d, %edx
          cmpl
          sete
                  %r14b
          addq
                  %r14, %rax
  .LVL8:
                  %r14d, %r14d
          xorl
                  %ebp, %edx
          cmpl
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL9:
                  %r14d, %r14d
          xorl
                  %ebx, %edx
          cmpl
          sete
                  %r14b
          addq
                  %r14, %rax
  .LVL10:
                  %r14d, %r14d
          xorl
                  %r11d, %edx
          cmpl
```

sete

%r14b

```
%r14, %rax
          addq
  .LVL11:
                  %r14d, %r14d
          xorl
          cmpl
                  %r10d, %edx
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL12:
                  %r14d, %r14d
          xorl
                  %r9d, %edx
          cmpl
                  %r14b
          sete
                  %r14, %rax
          addq
  .LVL13:
                  %r14d, %r14d
          xorl
          cmpl
                  %r8d, %edx
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL14:
                  %r14d, %r14d
          xorl
                  %edi, %edx
          cmpl
          sete
                  %r14b
                  %r14, %rax
          addq
  .LVL15:
          cmpl
                  %esi, %edx
                  %dl
          sete
          movzbl %dl, %edx
          addq
                  %rdx, %rax
  .LVL16:
  .LBE529:
                  $1, %rax
          cmpq
          jne
                  .L4164
          addq
                  $4, %rcx
          .loc 1 120 0 is_stmt 0 discriminator 2 \,
                  $a+40, %rcx
          cmpq
          jne
  .LVL17:
  .L6:
  .LBE528:
          .loc 1 121 0 is_stmt 1
• printf("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                  $.LC4, %edi
          movl
          movl
                  $0, %eax
          call
                  printf
```

• printf("helloworldn"); with gcc -g -01 produces:

.loc 1 121 0

```
.LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 104 0
         movl $.LC4, %esi
         movl $1, %edi
         movl
               $0, %eax
         call
                __printf_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• printf("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 104 0
         movl $.LC4, %esi
                 $1, %edi
         movl
         xorl %eax, %eax
          call
                 __printf_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
                 stdout(%rip), %rax
         movq
                 %rax, %rcx
         movq
                 $11, %edx
         movl
                  $1, %esi
         movl
         movl
                 $.LC4, %edi
          call
                 fwrite
          .loc 1 121 0
• L("helloworldn"); with gcc -g -01 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
```

```
stdout(%rip), %rcx
         movq
                 $11, %edx
         movl
                 $1, %esi
         movl
         movl
               $.LC4, %edi
         call fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• L("helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
         .loc 2 97 0
         movq stdout(%rip), %rcx
                $11, %edx
         movl
         movl $1, %esi
         movl
               $.LC4, %edi
         call
                 fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
         .loc 1 121 0
• DL("helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
         movl $0, _dl_target(%rip)
         .loc 1 121 0
• DL("helloworldn"); with gcc -g -O1 produces:
          .loc 1 120 0
 #NO_APP
         movl
                $0, _dl_target(%rip)
          .loc 1 121 0
• DL("helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
                 $0, _dl_target(%rip)
         .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
```

```
.loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
          andl
                  $16, %eax
                  %eax, %eax
          testl
          jе
          .loc 1 120 0 is_stmt 0 discriminator 1
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  $0, gi(%rip)
          movl
          movl
                  gi(%rip), %eax
                  $16, %al
          testb
                  .L5
          jе
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = 0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $0, gi(%rip)
          movl
                  gi(%rip), %eax
          testb
                  $16, %al
                  .L1045
          jne
  .L5:
          .loc 1 121 0
```

```
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -00 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $-1, gi(%rip)
                  gi(%rip), %eax
          movl
                  $16, %eax
          andl
                  %eax, %eax
          testl
                   .L5
          jе
          .loc 1 120 0 is_stmt 0 discriminator 1 \,
                  stdout(%rip), %rax
          movq
                  %rax, %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .L5:
          .loc 1 121 0 is_stmt 1
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -01 produces:
          .loc 1 120 0
 #NO_APP
                  $-1, gi(%rip)
          movl
                  gi(%rip), %eax
          movl
                  $16, %al
          testb
          jе
                   .L5
  .LVL5:
  .LBB526:
  .LBB527:
          .loc 2 97 0 discriminator 1
                  stdout(%rip), %rcx
          movq
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
                  fwrite
          call
  .LVL6:
  .L5:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• gi = ~0; LG(gi & 0x10, "helloworldn"); with gcc -g -03 produces:
          .loc 1 120 0
 #NO_APP
          movl
                  $-1, gi(%rip)
          movl
                  gi(%rip), %eax
                  $16, %al
```

testb

```
.L1045
          jne
  .L5:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -OO produces:
          .loc 1 120 0
 #NO_APP
                  -32(\%rbp), \%rax
          movq
          movq
                  %rax, %rcx
          movl
                  $11, %edx
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -O1 produces:
  .LVL5:
 #NO_APP
  .LBB526:
  .LBB527:
          .loc 2 97 0
          movq %rbx, %rcx
                  $11, %edx
          movl
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
• LHP(fprintf,log,"helloworldn"); with gcc -g -O3 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 2 97 0
                 $11, %edx
          movl
                  %rbx, %rcx
          movq
                  $1, %esi
          movl
                  $.LC4, %edi
          movl
          call
                  fwrite
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
```

• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -00 produces:

.loc 1 120 0
#NO_APP

MPP

movq -40(%rbp), %rax
movl \$.LC4, %esi
movq %rax, %rdi
movl \$0, %eax
call L_buffer_printf
.loc 1 121 0

 \bullet LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -01 produces:

.loc 1 120 0

#NO_APP

movl \$.LC4, %esi
movq %rbx, %rdi
movl \$0, %eax
call L_buffer_printf
.LVL6:
.loc 1 121 0

• LHP(L_buffer_printf,buf,"helloworldn"); with gcc -g -03 produces:

```
.loc 1 120 0

#NO_APP

movl $.LC4, %esi
movq %rbx, %rdi
xorl %eax, %eax
call L_buffer_printf
.LVL279:
.loc 1 121 0
```

• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -OO produces:

```
.loc 1 120 0
#NO_APP

movl $.LC4, %esi
movl $8, %edi
movl $0, %eax
call syslog
.loc 1 121 0
```

• LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -O1 produces:

.LVL5: #NO_APP .LBB526: .LBB527:

```
.file 3 "/usr/include/x86_64-linux-gnu/bits/syslog.h"
          .loc 3 31 0
                  $.LC4, %edx
          movl
          movl
                  $1, %esi
          movl
                  $8, %edi
                  $0, %eax
          movl
          call
                  __syslog_chk
  .LVL6:
  .LBE527:
  .LBE526:
          .loc 1 121 0
\bullet LHP(syslog,LOG_USER,"helloworldn"); with gcc -g -03 produces:
  .LVL277:
 #NO_APP
  .LBB1050:
  .LBB1051:
          .loc 3 31 0
                  $.LC4, %edx
          movl
          movl
                  $1, %esi
                  $8, %edi
          movl
                  %eax, %eax
          xorl
          call
                  __syslog_chk
  .LVL278:
  .LBE1051:
  .LBE1050:
          .loc 1 121 0
```

4 Conclusion

Finally, if you have used this package on an interesting (or uninteresting) architecture please mail me a copy of the results for the nana home page.