(Ab)using compiler tools

Reka Kovacs / rekanikolett@gmail.com

CppCon 2019

(Ab)using compiler tools

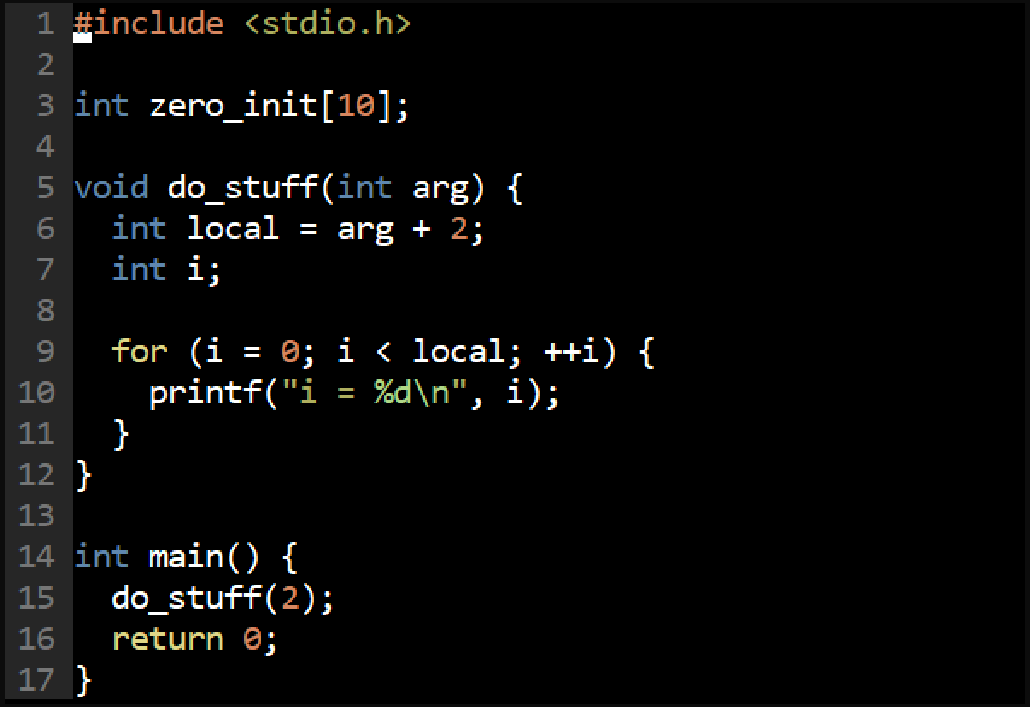
Two binary analysis tools
Only one of them is strictly connected to compilers
No abuse, no fun

Reka Kovacs / rekanikolett@gmail.com

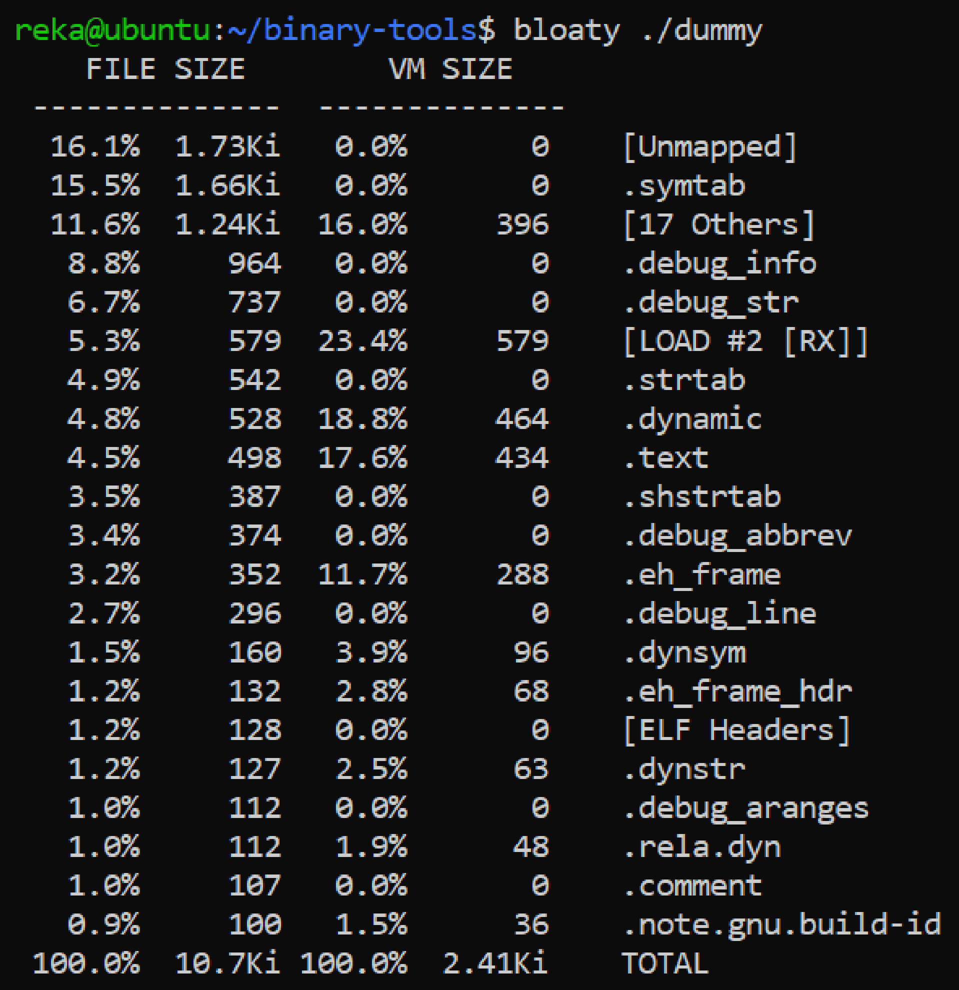
CppCon 2019

Bloaty McBloatface

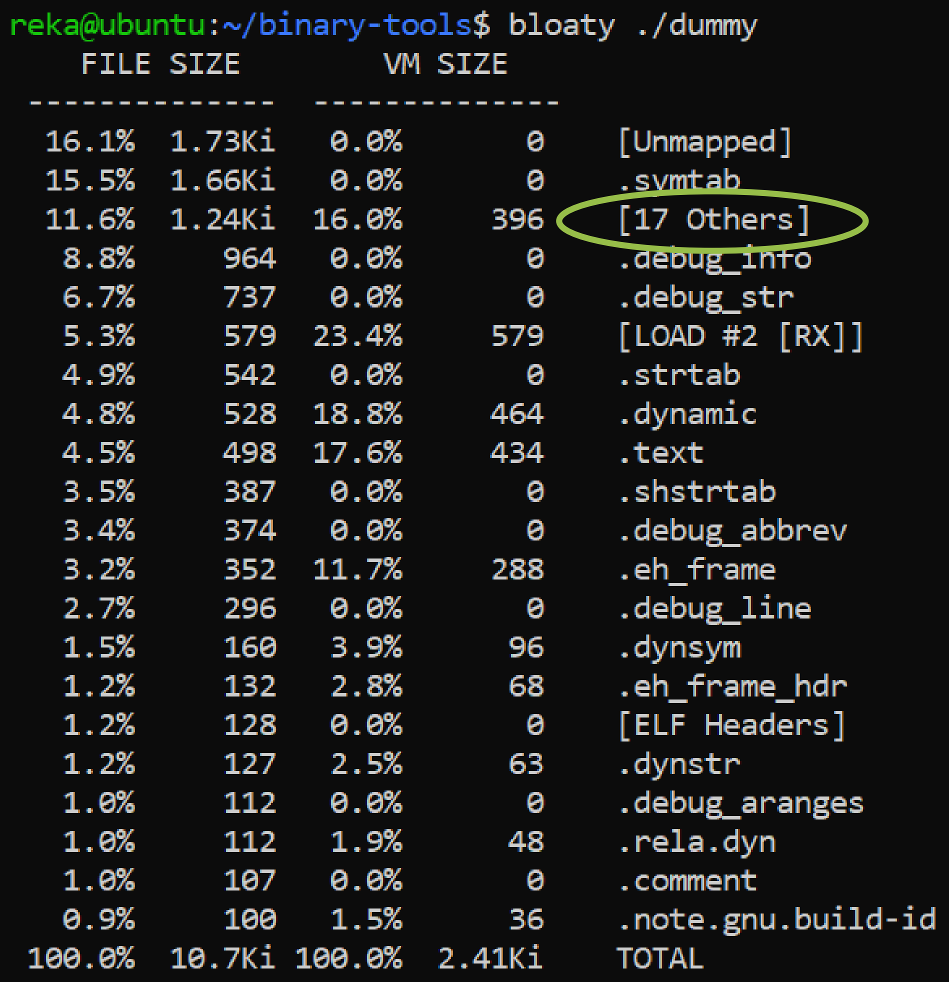
- Josh Haberman, 2016
- https://github.com/google/bloaty
- http://blog.reverberate.org/
- What is taking up space in your binary?
- Supports ELF and Mach-O, PE is desired



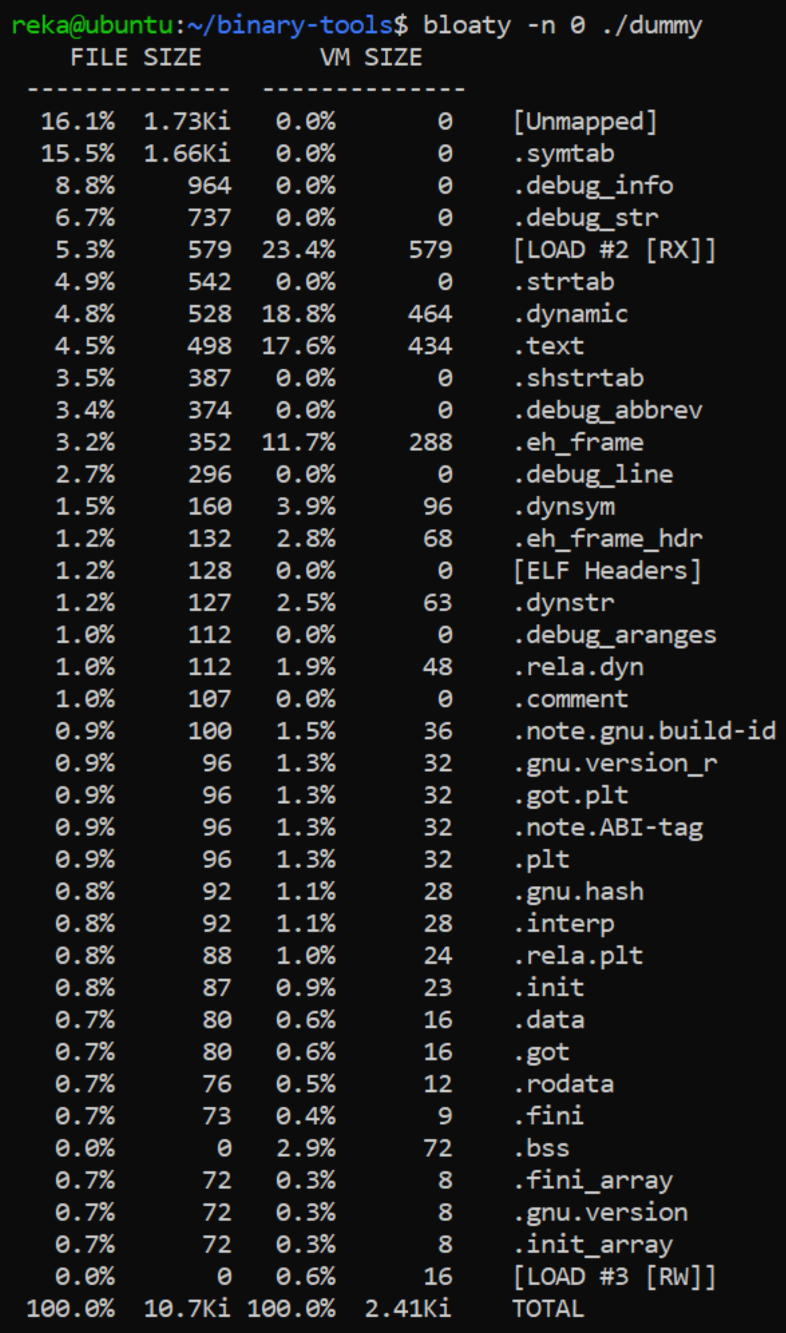
```c
#include <stdio.h>

int zero_init[10];

void do_stuff(int arg) {
  int local = arg + 2;
  int i;

  for (i = 0; i < local; ++i) {
    printf("i = %d\n", i);
  }
}

int main() {
  do_stuff(2);
  return 0;
}
```



```
reka@ubuntu:~/binary-tools$ bloaty ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  16.1%  1.73Ki   0.0%       0    [Unmapped]
  15.5%  1.66Ki   0.0%       0    .symtab
  11.6%  1.24Ki  16.0%     396    [17 Others]
   8.8%     964   0.0%       0    .debug_info
   6.7%     737   0.0%       0    .debug_str
   5.3%     579  23.4%     579    [LOAD #2 [RX]]
   4.9%     542   0.0%       0    .strtab
   4.8%     528  18.8%     464    .dynamic
   4.5%     498  17.6%     434    .text
   3.5%     387   0.0%       0    .shstrtab
   3.4%     374   0.0%       0    .debug_abbrev
   3.2%     352  11.7%     288    .eh_frame
   2.7%     296   0.0%       0    .debug_line
   1.5%     160   3.9%      96    .dynsym
   1.2%     132   2.8%      68    .eh_frame_hdr
   1.2%     128   0.0%       0    [ELF Headers]
   1.2%     127   2.5%      63    .dynstr
   1.0%     112   0.0%       0    .debug_aranges
   1.0%     112   1.9%      48    .rela.dyn
   1.0%     107   0.0%       0    .comment
   0.9%     100   1.5%      36    .note.gnu.build-id
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```



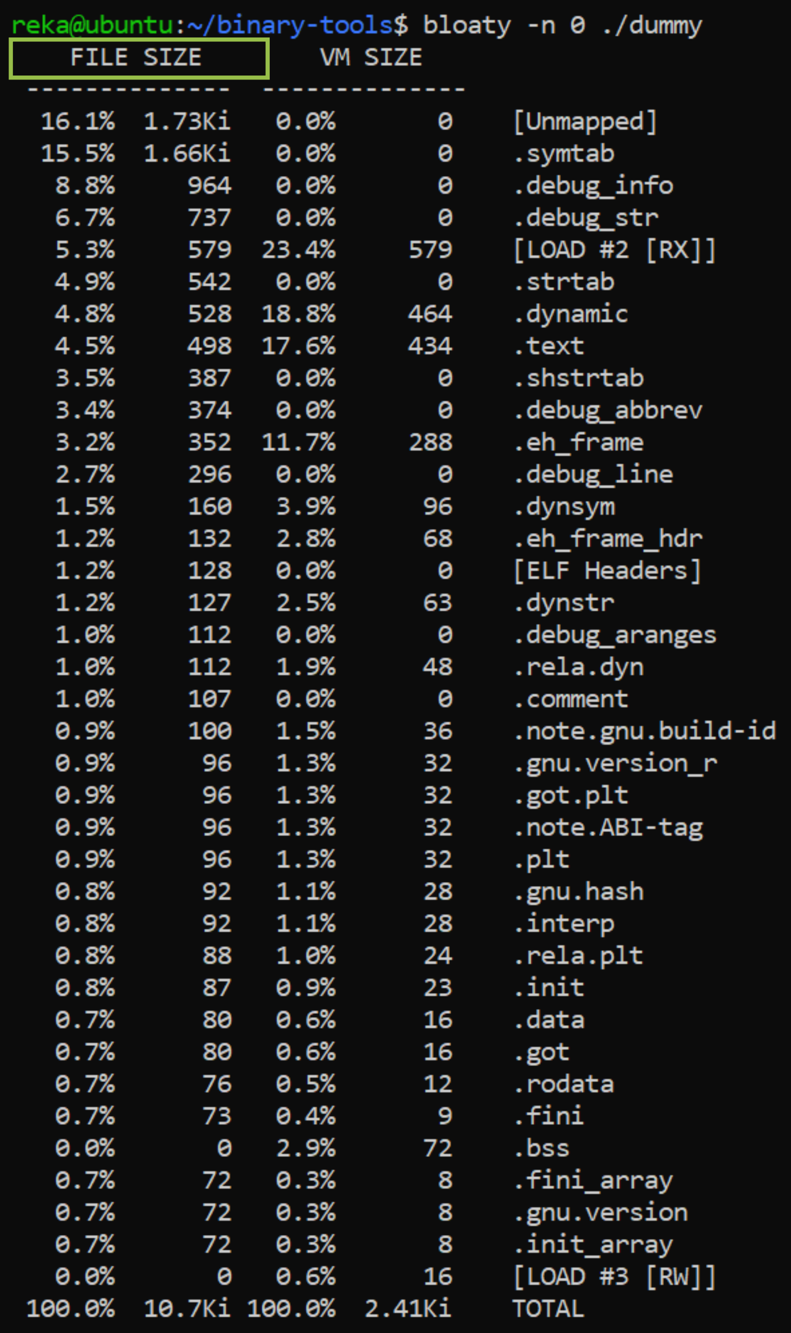
```
reka@ubuntu:~/binary-tools$ bloaty ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  16.1%  1.73Ki   0.0%       0    [Unmapped]
  15.5%  1.66Ki   0.0%       0    .symtab
  11.6%  1.24Ki  16.0%     396    [17 Others]
   8.8%     964   0.0%       0    .debug_info
   6.7%     737   0.0%       0    .debug_str
   5.3%     579  23.4%     579    [LOAD #2 [RX]]
   4.9%     542   0.0%       0    .strtab
   4.8%     528  18.8%     464    .dynamic
   4.5%     498  17.6%     434    .text
   3.5%     387   0.0%       0    .shstrtab
   3.4%     374   0.0%       0    .debug_abbrev
   3.2%     352  11.7%     288    .eh_frame
   2.7%     296   0.0%       0    .debug_line
   1.5%     160   3.9%      96    .dynsym
   1.2%     132   2.8%      68    .eh_frame_hdr
   1.2%     128   0.0%       0    [ELF Headers]
   1.2%     127   2.5%      63    .dynstr
   1.0%     112   0.0%       0    .debug_aranges
   1.0%     112   1.9%      48    .rela.dyn
   1.0%     107   0.0%       0    .comment
   0.9%     100   1.5%      36    .note.gnu.build-id
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```



```
reka@ubuntu:~/binary-tools$ bloaty -n 0 ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  16.1%  1.73Ki   0.0%       0    [Unmapped]
  15.5%  1.66Ki   0.0%       0    .symtab
   8.8%     964   0.0%       0    .debug_info
   6.7%     737   0.0%       0    .debug_str
   5.3%     579  23.4%     579    [LOAD #2 [RX]]
   4.9%     542   0.0%       0    .strtab
   4.8%     528  18.8%     464    .dynamic
   4.5%     498  17.6%     434    .text
   3.5%     387   0.0%       0    .shstrtab
   3.4%     374   0.0%       0    .debug_abbrev
   3.2%     352  11.7%     288    .eh_frame
   2.7%     296   0.0%       0    .debug_line
   1.5%     160   3.9%      96    .dynsym
   1.2%     132   2.8%      68    .eh_frame_hdr
   1.2%     128   0.0%       0    [ELF Headers]
   1.2%     127   2.5%      63    .dynstr
   1.0%     112   0.0%       0    .debug_aranges
   1.0%     112   1.9%      48    .rela.dyn
   1.0%     107   0.0%       0    .comment
   0.9%     100   1.5%      36    .note.gnu.build-id
   0.9%      96   1.3%      32    .gnu.version_r
   0.9%      96   1.3%      32    .got.plt
   0.9%      96   1.3%      32    .note.ABI-tag
   0.9%      96   1.3%      32    .plt
   0.8%      92   1.1%      28    .gnu.hash
   0.8%      92   1.1%      28    .interp
   0.8%      88   1.0%      24    .rela.plt
   0.8%      87   0.9%      23    .init
   0.7%      80   0.6%      16    .data
   0.7%      80   0.6%      16    .got
   0.7%      76   0.5%      12    .rodata
   0.7%      73   0.4%       9    .fini
   0.0%       0   2.9%      72    .bss
   0.7%      72   0.3%       8    .fini_array
   0.7%      72   0.3%       8    .gnu.version
   0.7%      72   0.3%       8    .init_array
   0.0%       0   0.6%      16    [LOAD #3 [RW]]
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

File size

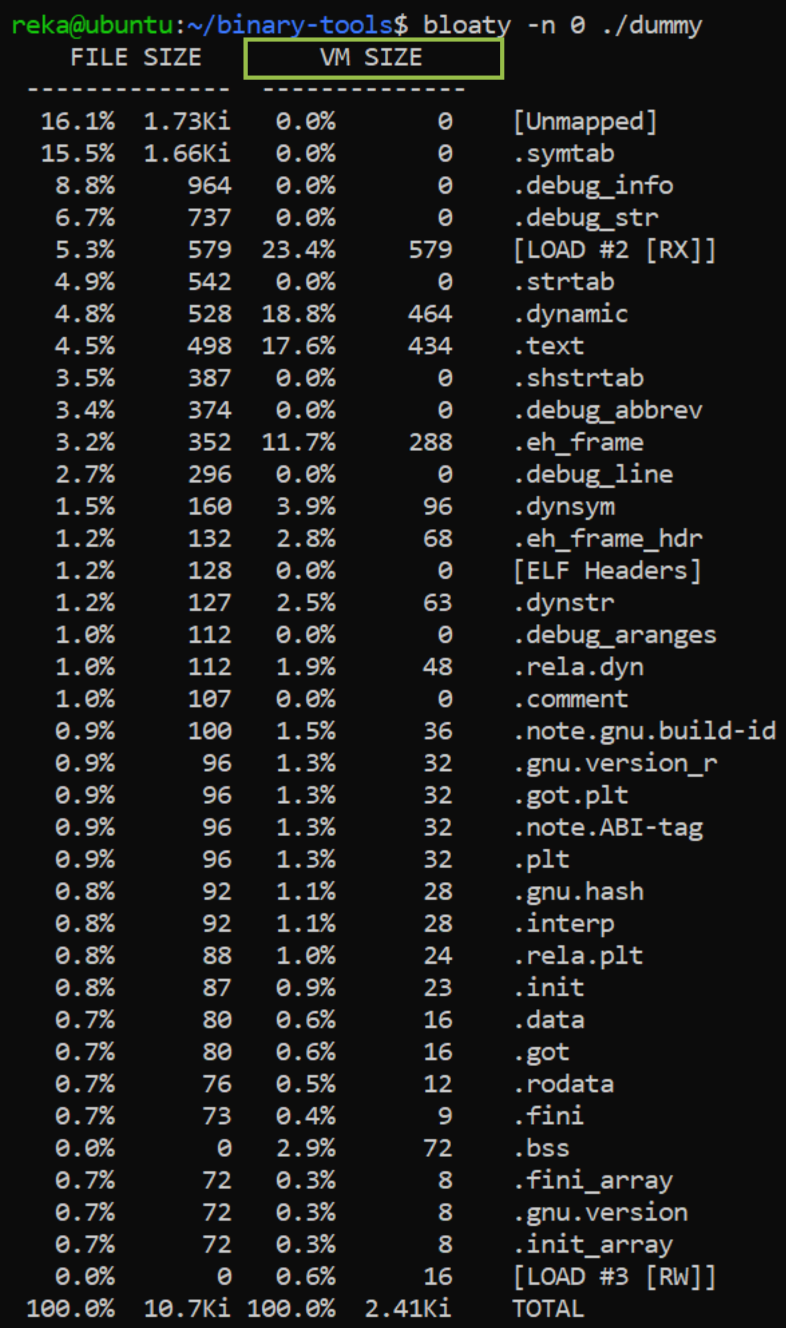
Bytes on disk



```
reka@ubuntu:~/binary-tools$ bloaty -n 0 ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  16.1%  1.73Ki   0.0%       0    [Unmapped]
  15.5%  1.66Ki   0.0%       0    .symtab
   8.8%     964   0.0%       0    .debug_info
   6.7%     737   0.0%       0    .debug_str
   5.3%     579  23.4%     579    [LOAD #2 [RX]]
   4.9%     542   0.0%       0    .strtab
   4.8%     528  18.8%     464    .dynamic
   4.5%     498  17.6%     434    .text
   3.5%     387   0.0%       0    .shstrtab
   3.4%     374   0.0%       0    .debug_abbrev
   3.2%     352  11.7%     288    .eh_frame
   2.7%     296   0.0%       0    .debug_line
   1.5%     160   3.9%      96    .dynsym
   1.2%     132   2.8%      68    .eh_frame_hdr
   1.2%     128   0.0%       0    [ELF Headers]
   1.2%     127   2.5%      63    .dynstr
   1.0%     112   0.0%       0    .debug_aranges
   1.0%     112   1.9%      48    .rela.dyn
   1.0%     107   0.0%       0    .comment
   0.9%     100   1.5%      36    .note.gnu.build-id
   0.9%      96   1.3%      32    .gnu.version_r
   0.9%      96   1.3%      32    .got.plt
   0.9%      96   1.3%      32    .note.ABI-tag
   0.9%      96   1.3%      32    .plt
   0.8%      92   1.1%      28    .gnu.hash
   0.8%      92   1.1%      28    .interp
   0.8%      88   1.0%      24    .rela.plt
   0.8%      87   0.9%      23    .init
   0.7%      80   0.6%      16    .data
   0.7%      80   0.6%      16    .got
   0.7%      76   0.5%      12    .rodata
   0.7%      73   0.4%       9    .fini
   0.0%       0   2.9%      72    .bss
   0.7%      72   0.3%       8    .fini_array
   0.7%      72   0.3%       8    .gnu.version
   0.7%      72   0.3%       8    .init_array
   0.0%       0   0.6%      16    [LOAD #3 [RW]]
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

File size

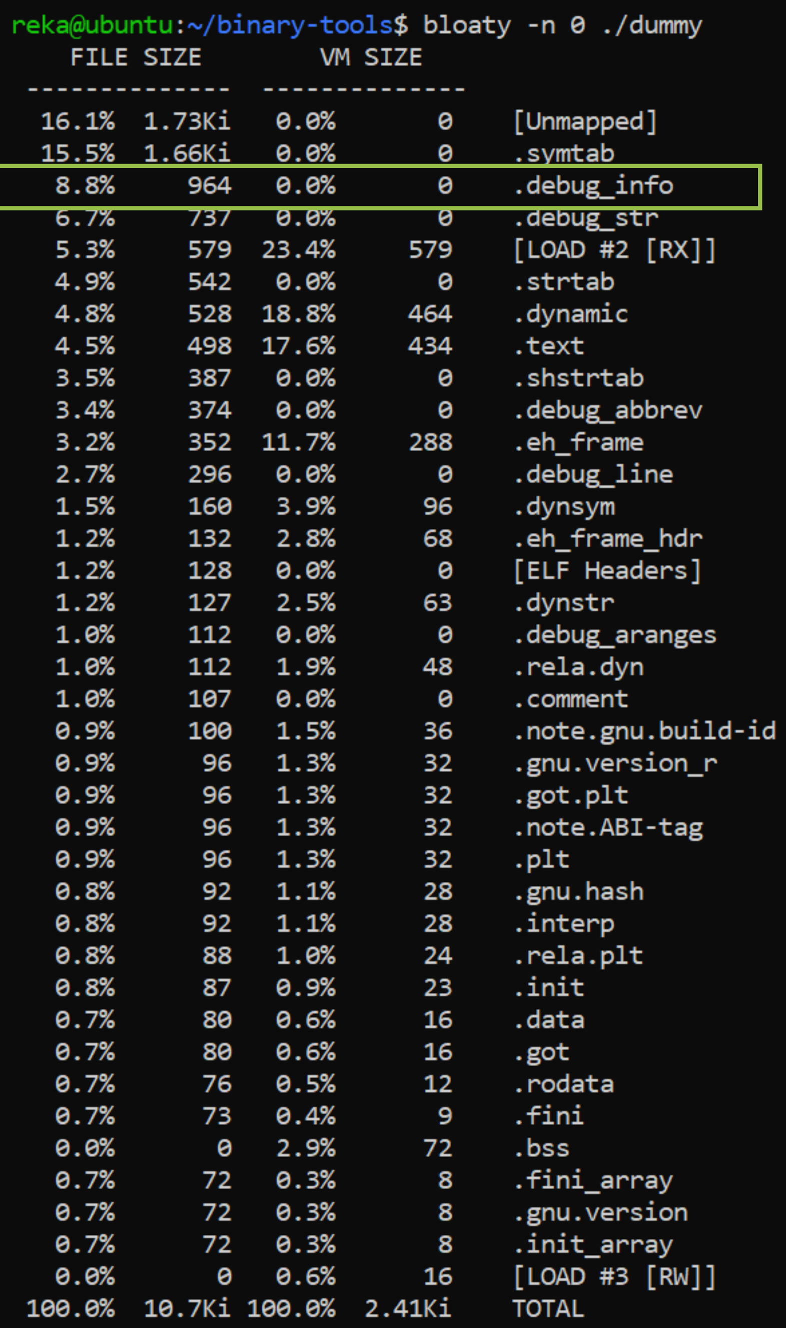
Bytes on disk



```
reka@ubuntu:~/binary-tools$ bloaty -n 0 ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  16.1%  1.73Ki   0.0%       0    [Unmapped]
  15.5%  1.66Ki   0.0%       0    .symtab
   8.8%     964   0.0%       0    .debug_info
   6.7%     737   0.0%       0    .debug_str
   5.3%     579  23.4%     579    [LOAD #2 [RX]]
   4.9%     542   0.0%       0    .strtab
   4.8%     528  18.8%     464    .dynamic
   4.5%     498  17.6%     434    .text
   3.5%     387   0.0%       0    .shstrtab
   3.4%     374   0.0%       0    .debug_abbrev
   3.2%     352  11.7%     288    .eh_frame
   2.7%     296   0.0%       0    .debug_line
   1.5%     160   3.9%      96    .dynsym
   1.2%     132   2.8%      68    .eh_frame_hdr
   1.2%     128   0.0%       0    [ELF Headers]
   1.2%     127   2.5%      63    .dynstr
   1.0%     112   0.0%       0    .debug_aranges
   1.0%     112   1.9%      48    .rela.dyn
   1.0%     107   0.0%       0    .comment
   0.9%     100   1.5%      36    .note.gnu.build-id
   0.9%      96   1.3%      32    .gnu.version_r
   0.9%      96   1.3%      32    .got.plt
   0.9%      96   1.3%      32    .note.ABI-tag
   0.9%      96   1.3%      32    .plt
   0.8%      92   1.1%      28    .gnu.hash
   0.8%      92   1.1%      28    .interp
   0.8%      88   1.0%      24    .rela.plt
   0.8%      87   0.9%      23    .init
   0.7%      80   0.6%      16    .data
   0.7%      80   0.6%      16    .got
   0.7%      76   0.5%      12    .rodata
   0.7%      73   0.4%       9    .fini
   0.0%       0   2.9%      72    .bss
   0.7%      72   0.3%       8    .fini_array
   0.7%      72   0.3%       8    .gnu.version
   0.7%      72   0.3%       8    .init_array
   0.0%       0   0.6%      16    [LOAD #3 [RW]]
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

VM size

Bytes in virtual memory after the executable is loaded



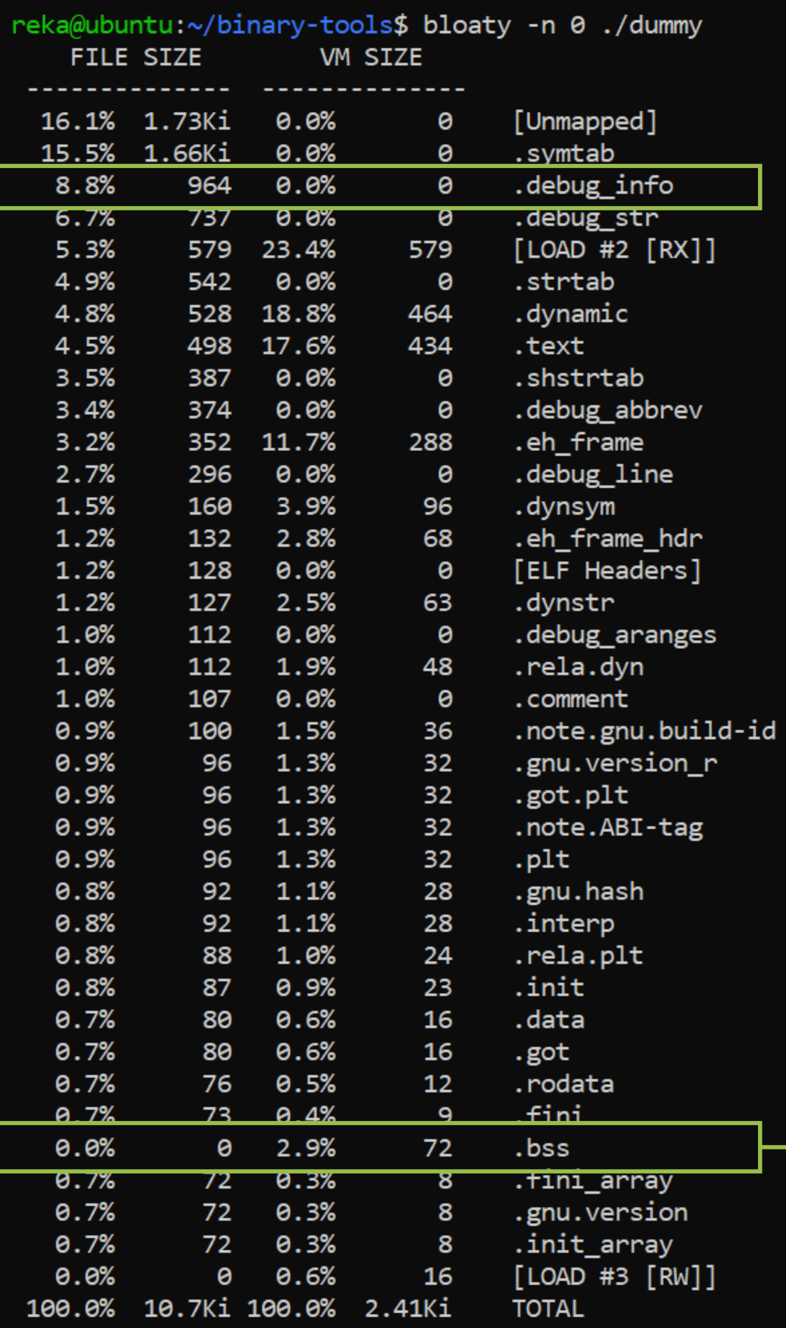
```
reka@ubuntu:~/binary-tools$ bloaty -n 0 ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  16.1%  1.73Ki   0.0%       0    [Unmapped]
  15.5%  1.66Ki   0.0%       0    .symtab
   8.8%     964   0.0%       0    .debug_info
   6.7%     737   0.0%       0    .debug_str
   5.3%     579  23.4%     579    [LOAD #2 [RX]]
   4.9%     542   0.0%       0    .strtab
   4.8%     528  18.8%     464    .dynamic
   4.5%     498  17.6%     434    .text
   3.5%     387   0.0%       0    .shstrtab
   3.4%     374   0.0%       0    .debug_abbrev
   3.2%     352  11.7%     288    .eh_frame
   2.7%     296   0.0%       0    .debug_line
   1.5%     160   3.9%      96    .dynsym
   1.2%     132   2.8%      68    .eh_frame_hdr
   1.2%     128   0.0%       0    [ELF Headers]
   1.2%     127   2.5%      63    .dynstr
   1.0%     112   0.0%       0    .debug_aranges
   1.0%     112   1.9%      48    .rela.dyn
   1.0%     107   0.0%       0    .comment
   0.9%     100   1.5%      36    .note.gnu.build-id
   0.9%      96   1.3%      32    .gnu.version_r
   0.9%      96   1.3%      32    .got.plt
   0.9%      96   1.3%      32    .note.ABI-tag
   0.9%      96   1.3%      32    .plt
   0.8%      92   1.1%      28    .gnu.hash
   0.8%      92   1.1%      28    .interp
   0.8%      88   1.0%      24    .rela.plt
   0.8%      87   0.9%      23    .init
   0.7%      80   0.6%      16    .data
   0.7%      80   0.6%      16    .got
   0.7%      76   0.5%      12    .rodata
   0.7%      73   0.4%       9    .fini
   0.0%       0   2.9%      72    .bss
   0.7%      72   0.3%       8    .fini_array
   0.7%      72   0.3%       8    .gnu.version
   0.7%      72   0.3%       8    .init_array
   0.0%       0   0.6%      16    [LOAD #3 [RW]]
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

File size

Bytes on disk

VM size

Bytes in virtual memory after the executable is loaded



```
reka@ubuntu:~/binary-tools$ bloaty -n 0 ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  16.1%  1.73Ki   0.0%       0    [Unmapped]
  15.5%  1.66Ki   0.0%       0    .symtab
   8.8%     964   0.0%       0    .debug_info
   6.7%     737   0.0%       0    .debug_str
   5.3%     579  23.4%     579    [LOAD #2 [RX]]
   4.9%     542   0.0%       0    .strtab
   4.8%     528  18.8%     464    .dynamic
   4.5%     498  17.6%     434    .text
   3.5%     387   0.0%       0    .shstrtab
   3.4%     374   0.0%       0    .debug_abbrev
   3.2%     352  11.7%     288    .eh_frame
   2.7%     296   0.0%       0    .debug_line
   1.5%     160   3.9%      96    .dynsym
   1.2%     132   2.8%      68    .eh_frame_hdr
   1.2%     128   0.0%       0    [ELF Headers]
   1.2%     127   2.5%      63    .dynstr
   1.0%     112   0.0%       0    .debug_aranges
   1.0%     112   1.9%      48    .rela.dyn
   1.0%     107   0.0%       0    .comment
   0.9%     100   1.5%      36    .note.gnu.build-id
   0.9%      96   1.3%      32    .gnu.version_r
   0.9%      96   1.3%      32    .got.plt
   0.9%      96   1.3%      32    .note.ABI-tag
   0.9%      96   1.3%      32    .plt
   0.8%      92   1.1%      28    .gnu.hash
   0.8%      92   1.1%      28    .interp
   0.8%      88   1.0%      24    .rela.plt
   0.8%      87   0.9%      23    .init
   0.7%      80   0.6%      16    .data
   0.7%      80   0.6%      16    .got
   0.7%      76   0.5%      12    .rodata
   0.7%      73   0.4%       9    .fini
   0.0%       0   2.9%      72    .bss
   0.7%      72   0.3%       8    .fini_array
   0.7%      72   0.3%       8    .gnu.version
   0.7%      72   0.3%       8    .init_array
   0.0%       0   0.6%      16    [LOAD #3 [RW]]
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

File size

Bytes on disk

VM size

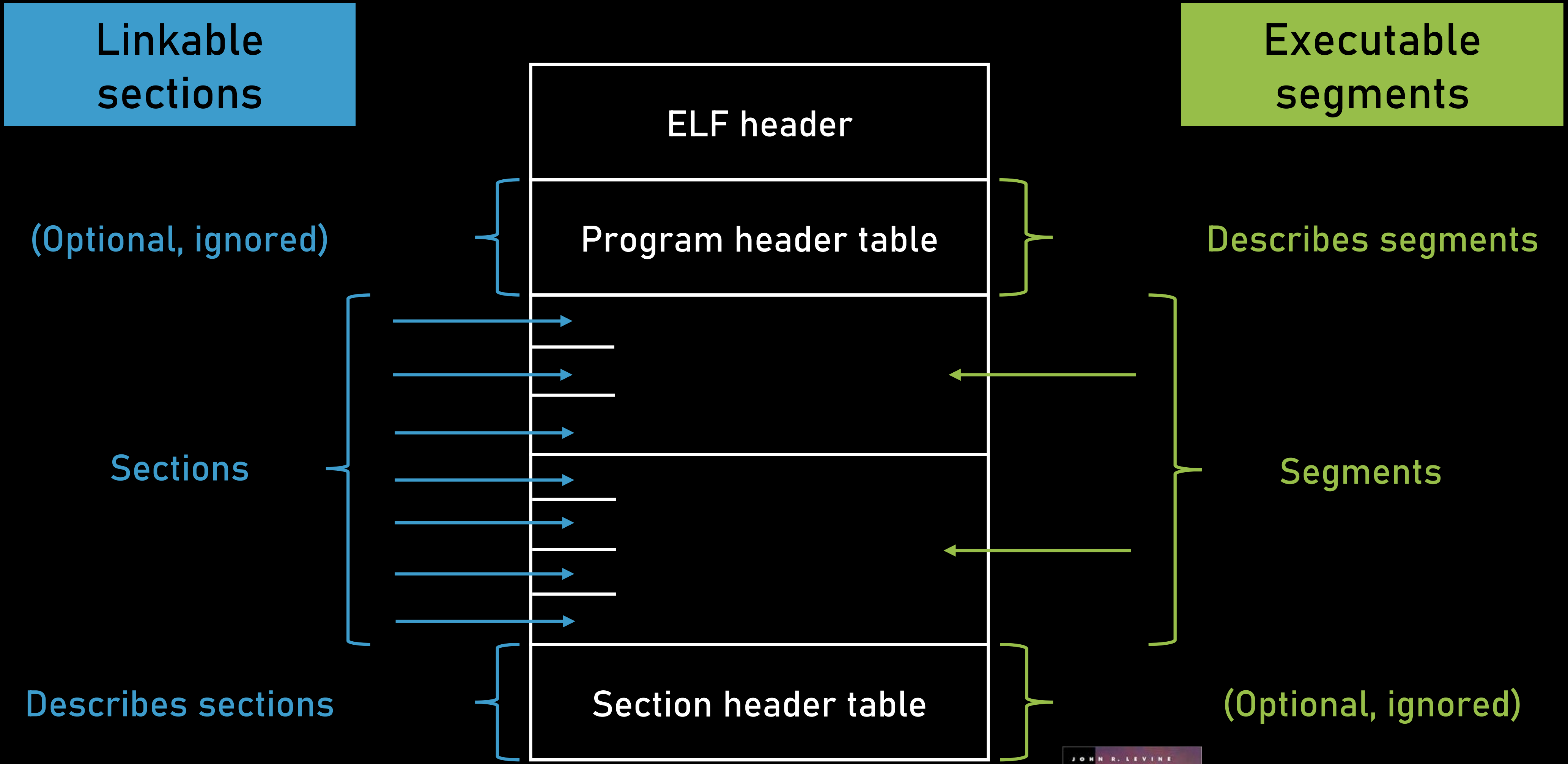
Bytes in virtual memory after the executable is loaded

ELF (Executable and Linkage Format)

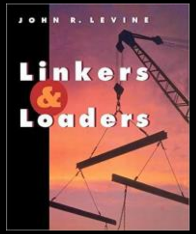
File type for binaries on Linux

3 flavors:

- Relocatable – needs to be processed by the linker
- Executable – all symbols resolved (except for shared lib symbols), runnable
- Shared object – has both symbol info for the linker and runnable code



John Levine: Linkers and Loaders



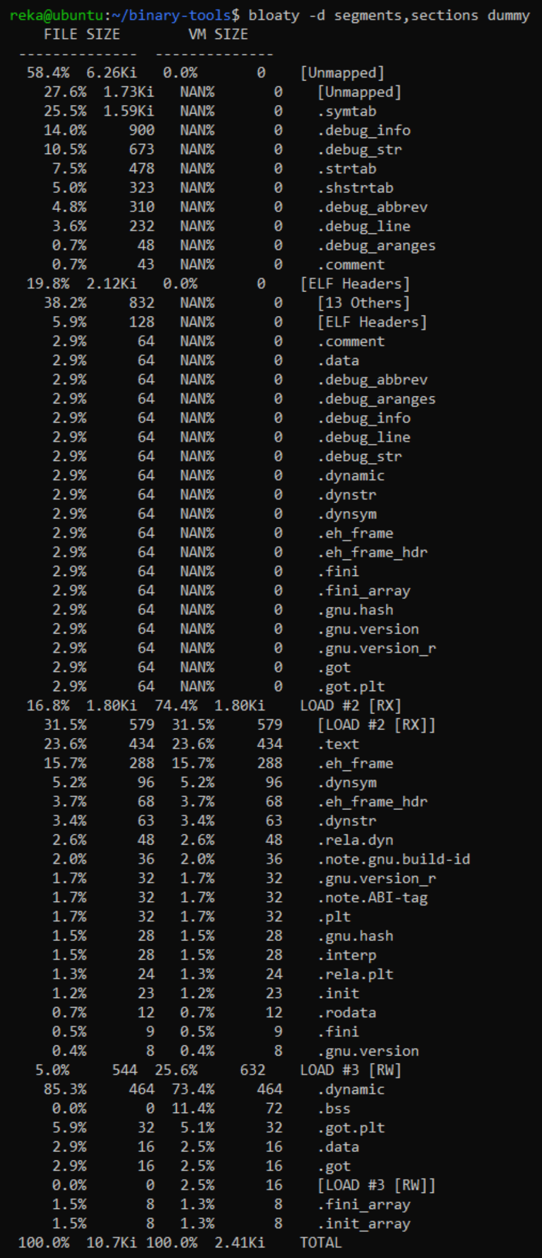
Segments and sections

```
reka@ubuntu:~/binary-tools$ bloaty -d segments ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  58.4%  6.26Ki   0.0%       0    [Unmapped]
  19.8%  2.12Ki   0.0%       0    [ELF Headers]
  16.8%  1.80Ki  74.4%  1.80Ki    LOAD #2 [RX]
   5.0%     544  25.6%     632    LOAD #3 [RW]
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

Segments and sections

```
reka@ubuntu:~/binary-tools$ bloaty -d segments ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  58.4%  6.26Ki   0.0%       0    [Unmapped]
  19.8%  2.12Ki   0.0%       0    [ELF Headers]
  16.8%  1.80Ki  74.4%  1.80Ki    LOAD #2 [RX]
   5.0%     544  25.6%     632    LOAD #3 [RW]
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

```
reka@ubuntu:~/binary-tools$ bloaty -d segments,sections dummy
```

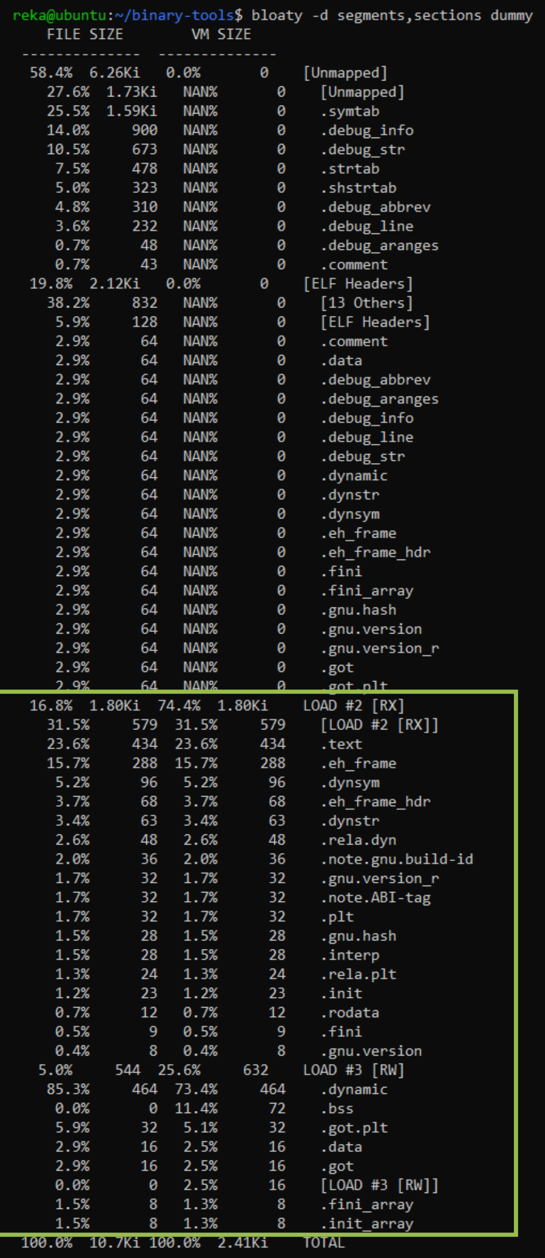


```
reka@ubuntu:~/binary-tools$ bloaty -d segments,sections dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  58.4%  6.26Ki   0.0%       0    [Unmapped]
    27.6%  1.73Ki   NAN%       0    [Unmapped]
    25.5%  1.59Ki   NAN%       0    .symtab
    14.0%     900   NAN%       0    .debug_info
    10.5%     673   NAN%       0    .debug_str
     7.5%     478   NAN%       0    .strtab
     5.0%     323   NAN%       0    .shstrtab
     4.8%     310   NAN%       0    .debug_abbrev
     3.6%     232   NAN%       0    .debug_line
     0.7%      48   NAN%       0    .debug_aranges
     0.7%      43   NAN%       0    .comment
  19.8%  2.12Ki   0.0%       0    [ELF Headers]
    38.2%     832   NAN%       0    [13 Others]
     5.9%     128   NAN%       0    [ELF Headers]
     2.9%      64   NAN%       0    .comment
     2.9%      64   NAN%       0    .data
     2.9%      64   NAN%       0    .debug_abbrev
     2.9%      64   NAN%       0    .debug_aranges
     2.9%      64   NAN%       0    .debug_info
     2.9%      64   NAN%       0    .debug_line
     2.9%      64   NAN%       0    .debug_str
     2.9%      64   NAN%       0    .dynamic
     2.9%      64   NAN%       0    .dynstr
     2.9%      64   NAN%       0    .dynsym
     2.9%      64   NAN%       0    .eh_frame
     2.9%      64   NAN%       0    .eh_frame_hdr
     2.9%      64   NAN%       0    .fini
     2.9%      64   NAN%       0    .fini_array
     2.9%      64   NAN%       0    .gnu.hash
     2.9%      64   NAN%       0    .gnu.version
     2.9%      64   NAN%       0    .gnu.version_r
     2.9%      64   NAN%       0    .got
     2.9%      64   NAN%       0    .got.plt
  16.8%  1.80Ki  74.4%  1.80Ki    LOAD #2 [RX]
    31.5%     579  31.5%     579    [LOAD #2 [RX]]
    23.6%     434  23.6%     434    .text
    15.7%     288  15.7%     288    .eh_frame
     5.2%      96   5.2%      96    .dynsym
     3.7%      68   3.7%      68    .eh_frame_hdr
     3.4%      63   3.4%      63    .dynstr
     2.6%      48   2.6%      48    .rela.dyn
     2.0%      36   2.0%      36    .note.gnu.build-id
     1.7%      32   1.7%      32    .gnu.version_r
     1.7%      32   1.7%      32    .note.ABI-tag
     1.7%      32   1.7%      32    .plt
     1.5%      28   1.5%      28    .gnu.hash
     1.5%      28   1.5%      28    .interp
     1.3%      24   1.3%      24    .rela.plt
     1.2%      23   1.2%      23    .init
     0.7%      12   0.7%      12    .rodata
     0.5%       9   0.5%       9    .fini
     0.4%       8   0.4%       8    .gnu.version
   5.0%     544  25.6%     632    LOAD #3 [RW]
    85.3%     464  73.4%     464    .dynamic
     0.0%       0  11.4%      72    .bss
     5.9%      32   5.1%      32    .got.plt
     2.9%      16   2.5%      16    .data
     2.9%      16   2.5%      16    .got
     0.0%       0   2.5%      16    [LOAD #3 [RW]]
     1.5%       8   1.3%       8    .fini_array
     1.5%       8   1.3%       8    .init_array
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

Segments and sections

```
reka@ubuntu:~/binary-tools$ bloaty -d segments ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  58.4%  6.26Ki   0.0%       0    [Unmapped]
  19.8%  2.12Ki   0.0%       0    [ELF Headers]
  16.8%  1.80Ki  74.4%  1.80Ki    LOAD #2 [RX]
   5.0%     544  25.6%     632    LOAD #3 [RW]
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

```
reka@ubuntu:~/binary-tools$ bloaty -d segments,sections dummy
```

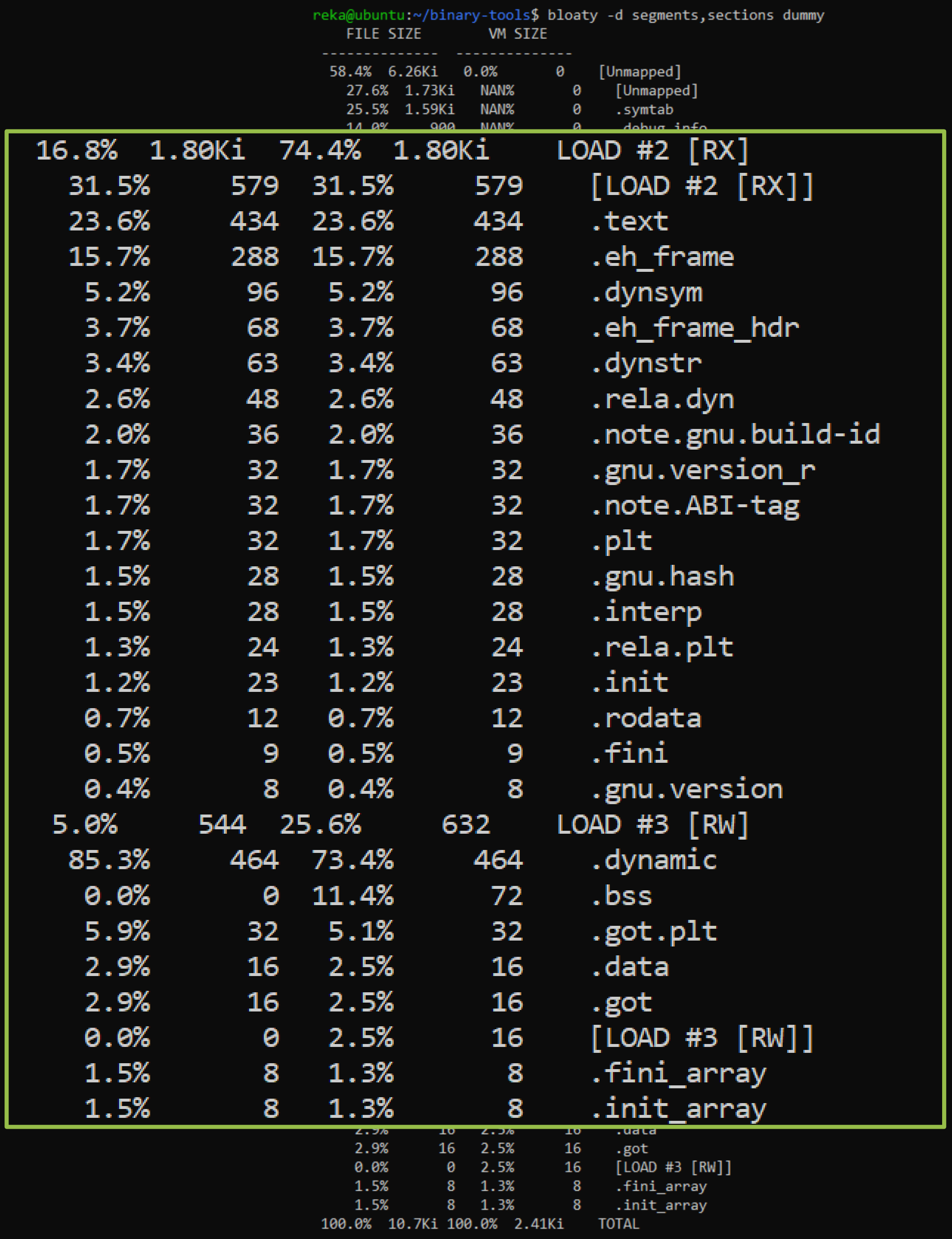


```
reka@ubuntu:~/binary-tools$ bloaty -d segments,sections dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  58.4%  6.26Ki   0.0%       0    [Unmapped]
    27.6%  1.73Ki   NAN%       0    [Unmapped]
    25.5%  1.59Ki   NAN%       0    .symtab
    14.0%     900   NAN%       0    .debug_info
    10.5%     673   NAN%       0    .debug_str
     7.5%     478   NAN%       0    .strtab
     5.0%     323   NAN%       0    .shstrtab
     4.8%     310   NAN%       0    .debug_abbrev
     3.6%     232   NAN%       0    .debug_line
     0.7%      48   NAN%       0    .debug_aranges
     0.7%      43   NAN%       0    .comment
  19.8%  2.12Ki   0.0%       0    [ELF Headers]
    38.2%     832   NAN%       0    [13 Others]
     5.9%     128   NAN%       0    [ELF Headers]
     2.9%      64   NAN%       0    .comment
     2.9%      64   NAN%       0    .data
     2.9%      64   NAN%       0    .debug_abbrev
     2.9%      64   NAN%       0    .debug_aranges
     2.9%      64   NAN%       0    .debug_info
     2.9%      64   NAN%       0    .debug_line
     2.9%      64   NAN%       0    .debug_str
     2.9%      64   NAN%       0    .dynamic
     2.9%      64   NAN%       0    .dynstr
     2.9%      64   NAN%       0    .dynsym
     2.9%      64   NAN%       0    .eh_frame
     2.9%      64   NAN%       0    .eh_frame_hdr
     2.9%      64   NAN%       0    .fini
     2.9%      64   NAN%       0    .fini_array
     2.9%      64   NAN%       0    .gnu.hash
     2.9%      64   NAN%       0    .gnu.version
     2.9%      64   NAN%       0    .gnu.version_r
     2.9%      64   NAN%       0    .got
     2.9%      64   NAN%       0    .got.plt
  16.8%  1.80Ki  74.4%  1.80Ki    LOAD #2 [RX]
    31.5%     579  31.5%     579    [LOAD #2 [RX]]
    23.6%     434  23.6%     434    .text
    15.7%     288  15.7%     288    .eh_frame
     5.2%      96   5.2%      96    .dynsym
     3.7%      68   3.7%      68    .eh_frame_hdr
     3.4%      63   3.4%      63    .dynstr
     2.6%      48   2.6%      48    .rela.dyn
     2.0%      36   2.0%      36    .note.gnu.build-id
     1.7%      32   1.7%      32    .gnu.version_r
     1.7%      32   1.7%      32    .note.ABI-tag
     1.7%      32   1.7%      32    .plt
     1.5%      28   1.5%      28    .gnu.hash
     1.5%      28   1.5%      28    .interp
     1.3%      24   1.3%      24    .rela.plt
     1.2%      23   1.2%      23    .init
     0.7%      12   0.7%      12    .rodata
     0.5%       9   0.5%       9    .fini
     0.4%       8   0.4%       8    .gnu.version
   5.0%     544  25.6%     632    LOAD #3 [RW]
    85.3%     464  73.4%     464    .dynamic
     0.0%       0  11.4%      72    .bss
     5.9%      32   5.1%      32    .got.plt
     2.9%      16   2.5%      16    .data
     2.9%      16   2.5%      16    .got
     0.0%       0   2.5%      16    [LOAD #3 [RW]]
     1.5%       8   1.3%       8    .fini_array
     1.5%       8   1.3%       8    .init_array
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

Segments and sections

```
reka@ubuntu:~/binary-tools$ bloaty -d segments ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  58.4%  6.26Ki   0.0%       0    [Unmapped]
  19.8%  2.12Ki   0.0%       0    [ELF Headers]
  16.8%  1.80Ki  74.4%  1.80Ki    LOAD #2 [RX]
   5.0%     544  25.6%     632    LOAD #3 [RW]
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

```
reka@ubuntu:~/binary-tools$ bloaty -d segments,sections dummy
```

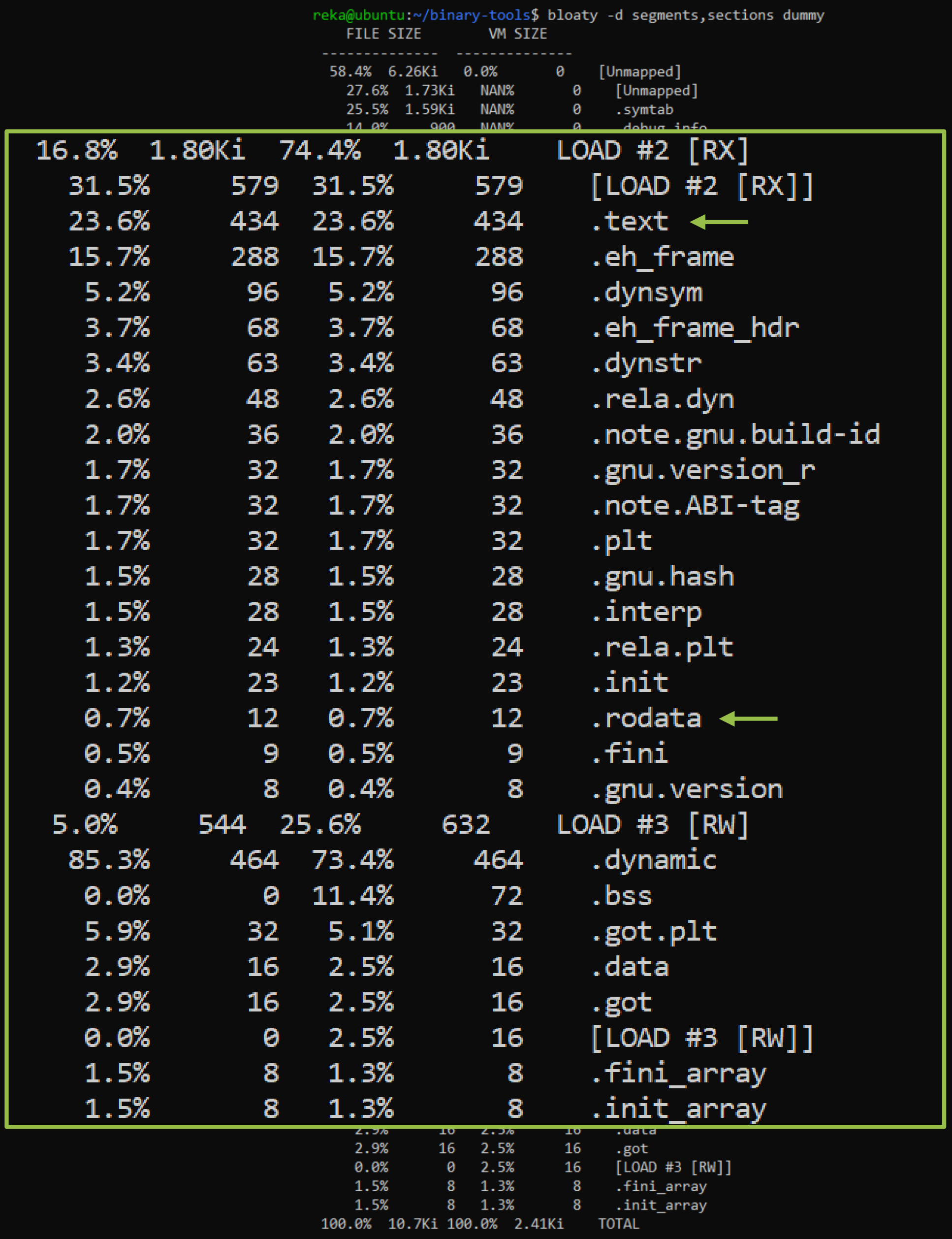


```
reka@ubuntu:~/binary-tools$ bloaty -d segments,sections dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  58.4%  6.26Ki   0.0%       0    [Unmapped]
    27.6%  1.73Ki   NAN%       0    [Unmapped]
    25.5%  1.59Ki   NAN%       0    .symtab
    14.0%     900   NAN%       0    .debug_info
  16.8%  1.80Ki  74.4%  1.80Ki    LOAD #2 [RX]
    31.5%     579  31.5%     579    [LOAD #2 [RX]]
    23.6%     434  23.6%     434    .text
    15.7%     288  15.7%     288    .eh_frame
     5.2%      96   5.2%      96    .dynsym
     3.7%      68   3.7%      68    .eh_frame_hdr
     3.4%      63   3.4%      63    .dynstr
     2.6%      48   2.6%      48    .rela.dyn
     2.0%      36   2.0%      36    .note.gnu.build-id
     1.7%      32   1.7%      32    .gnu.version_r
     1.7%      32   1.7%      32    .note.ABI-tag
     1.7%      32   1.7%      32    .plt
     1.5%      28   1.5%      28    .gnu.hash
     1.5%      28   1.5%      28    .interp
     1.3%      24   1.3%      24    .rela.plt
     1.2%      23   1.2%      23    .init
     0.7%      12   0.7%      12    .rodata
     0.5%       9   0.5%       9    .fini
     0.4%       8   0.4%       8    .gnu.version
   5.0%     544  25.6%     632    LOAD #3 [RW]
    85.3%     464  73.4%     464    .dynamic
     0.0%       0  11.4%      72    .bss
     5.9%      32   5.1%      32    .got.plt
     2.9%      16   2.5%      16    .data
     2.9%      16   2.5%      16    .got
     0.0%       0   2.5%      16    [LOAD #3 [RW]]
     1.5%       8   1.3%       8    .fini_array
     1.5%       8   1.3%       8    .init_array
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

Segments and sections

```
reka@ubuntu:~/binary-tools$ bloaty -d segments ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  58.4%  6.26Ki   0.0%       0    [Unmapped]
  19.8%  2.12Ki   0.0%       0    [ELF Headers]
  16.8%  1.80Ki  74.4%  1.80Ki    LOAD #2 [RX]  ←
   5.0%     544  25.6%     632    LOAD #3 [RW]
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

```
reka@ubuntu:~/binary-tools$ bloaty -d segments,sections dummy
```



```
reka@ubuntu:~/binary-tools$ bloaty -d segments,sections dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  58.4%  6.26Ki   0.0%       0    [Unmapped]
    27.6%  1.73Ki   NAN%       0    [Unmapped]
    25.5%  1.59Ki   NAN%       0    .symtab
    14.0%     900   NAN%       0    .debug_info
  16.8%  1.80Ki  74.4%  1.80Ki    LOAD #2 [RX]
    31.5%     579  31.5%     579    [LOAD #2 [RX]]
    23.6%     434  23.6%     434    .text  ←
    15.7%     288  15.7%     288    .eh_frame
     5.2%      96   5.2%      96    .dynsym
     3.7%      68   3.7%      68    .eh_frame_hdr
     3.4%      63   3.4%      63    .dynstr
     2.6%      48   2.6%      48    .rela.dyn
     2.0%      36   2.0%      36    .note.gnu.build-id
     1.7%      32   1.7%      32    .gnu.version_r
     1.7%      32   1.7%      32    .note.ABI-tag
     1.7%      32   1.7%      32    .plt
     1.5%      28   1.5%      28    .gnu.hash
     1.5%      28   1.5%      28    .interp
     1.3%      24   1.3%      24    .rela.plt
     1.2%      23   1.2%      23    .init
     0.7%      12   0.7%      12    .rodata  ←
     0.5%       9   0.5%       9    .fini
     0.4%       8   0.4%       8    .gnu.version
   5.0%     544  25.6%     632    LOAD #3 [RW]
    85.3%     464  73.4%     464    .dynamic
     0.0%       0  11.4%      72    .bss
     5.9%      32   5.1%      32    .got.plt
     2.9%      16   2.5%      16    .data
     2.9%      16   2.5%      16    .got
     0.0%       0   2.5%      16    [LOAD #3 [RW]]
     1.5%       8   1.3%       8    .fini_array
     1.5%       8   1.3%       8    .init_array
```

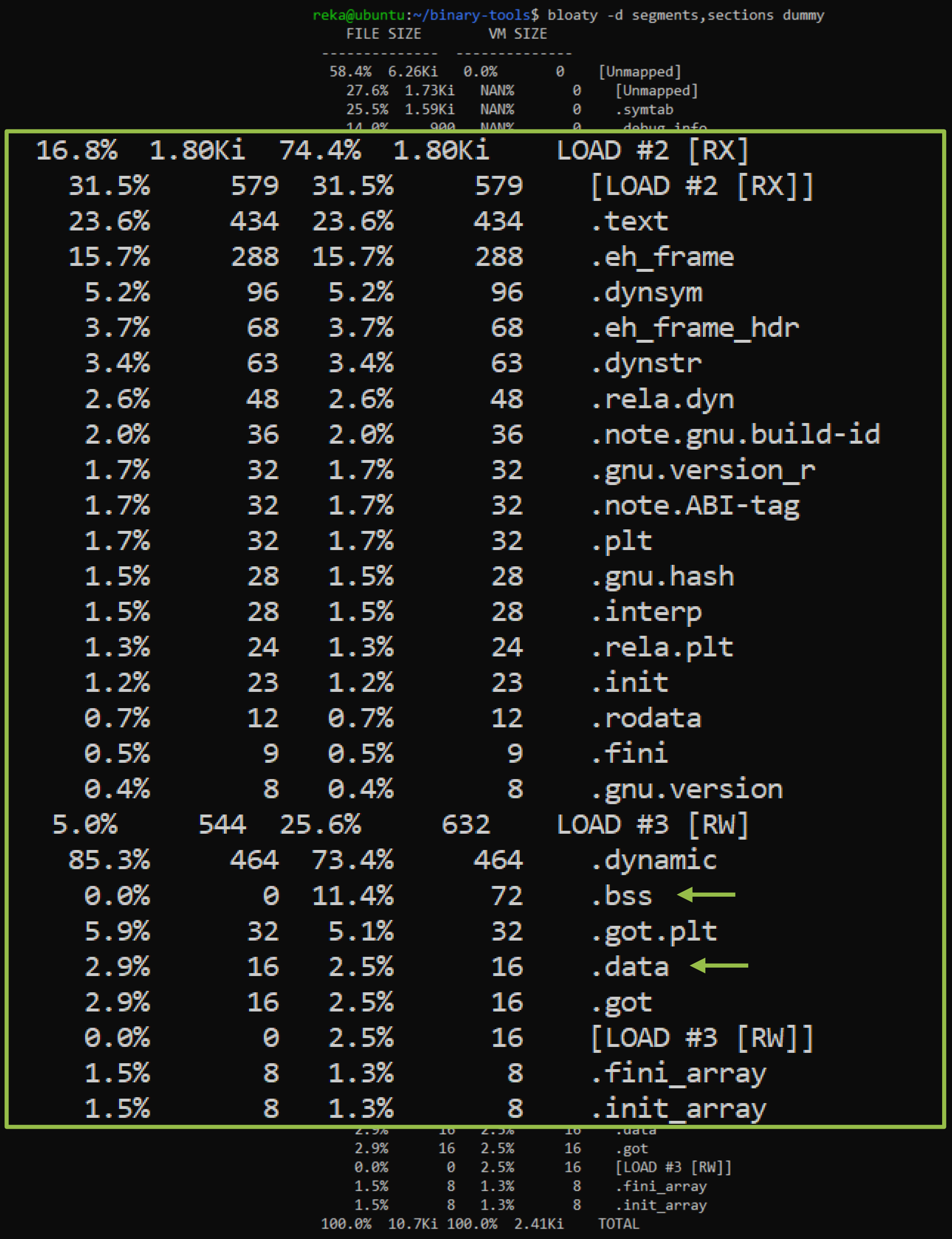
```
     2.9%      16   2.5%      16    .data
     2.9%      16   2.5%      16    .got
     0.0%       0   2.5%      16    [LOAD #3 [RW]]
     1.5%       8   1.3%       8    .fini_array
     1.5%       8   1.3%       8    .init_array
```

```
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

Segments and sections

```
reka@ubuntu:~/binary-tools$ bloaty -d segments ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  58.4%  6.26Ki   0.0%       0    [Unmapped]
  19.8%  2.12Ki   0.0%       0    [ELF Headers]
  16.8%  1.80Ki  74.4%  1.80Ki    LOAD #2 [RX]
   5.0%     544  25.6%     632    LOAD #3 [RW]
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

```
reka@ubuntu:~/binary-tools$ bloaty -d segments,sections dummy
```

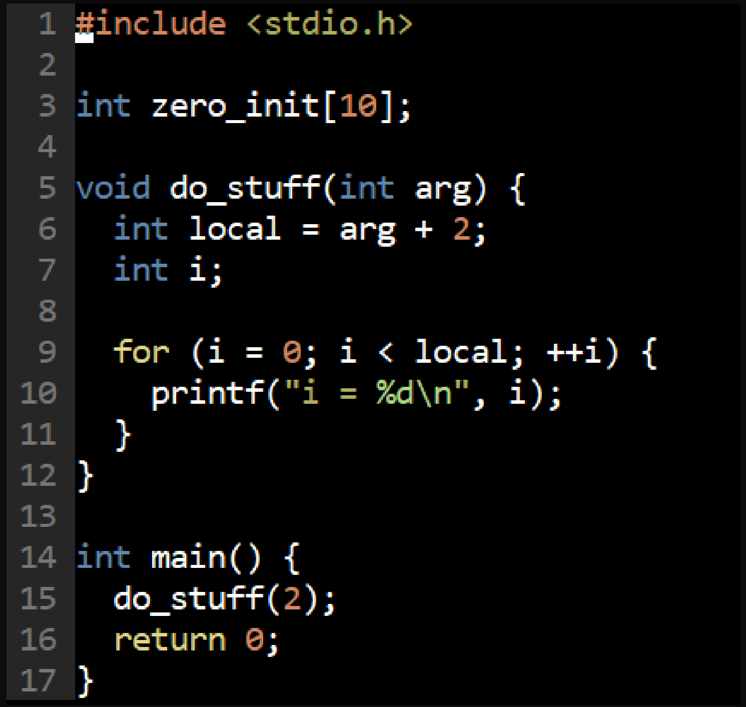


```
reka@ubuntu:~/binary-tools$ bloaty -d segments,sections dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  58.4%  6.26Ki   0.0%       0    [Unmapped]
    27.6%  1.73Ki   NAN%       0    [Unmapped]
    25.5%  1.59Ki   NAN%       0    .symtab
    14.0%     900   NAN%       0    .debug_info
  16.8%  1.80Ki  74.4%  1.80Ki    LOAD #2 [RX]
    31.5%     579  31.5%     579    [LOAD #2 [RX]]
    23.6%     434  23.6%     434    .text
    15.7%     288  15.7%     288    .eh_frame
     5.2%      96   5.2%      96    .dynsym
     3.7%      68   3.7%      68    .eh_frame_hdr
     3.4%      63   3.4%      63    .dynstr
     2.6%      48   2.6%      48    .rela.dyn
     2.0%      36   2.0%      36    .note.gnu.build-id
     1.7%      32   1.7%      32    .gnu.version_r
     1.7%      32   1.7%      32    .note.ABI-tag
     1.7%      32   1.7%      32    .plt
     1.5%      28   1.5%      28    .gnu.hash
     1.5%      28   1.5%      28    .interp
     1.3%      24   1.3%      24    .rela.plt
     1.2%      23   1.2%      23    .init
     0.7%      12   0.7%      12    .rodata
     0.5%       9   0.5%       9    .fini
     0.4%       8   0.4%       8    .gnu.version
   5.0%     544  25.6%     632    LOAD #3 [RW]
    85.3%     464  73.4%     464    .dynamic
     0.0%       0  11.4%      72    .bss
     5.9%      32   5.1%      32    .got.plt
     2.9%      16   2.5%      16    .data
     2.9%      16   2.5%      16    .got
     0.0%       0   2.5%      16    [LOAD #3 [RW]]
     1.5%       8   1.3%       8    .fini_array
     1.5%       8   1.3%       8    .init_array
```

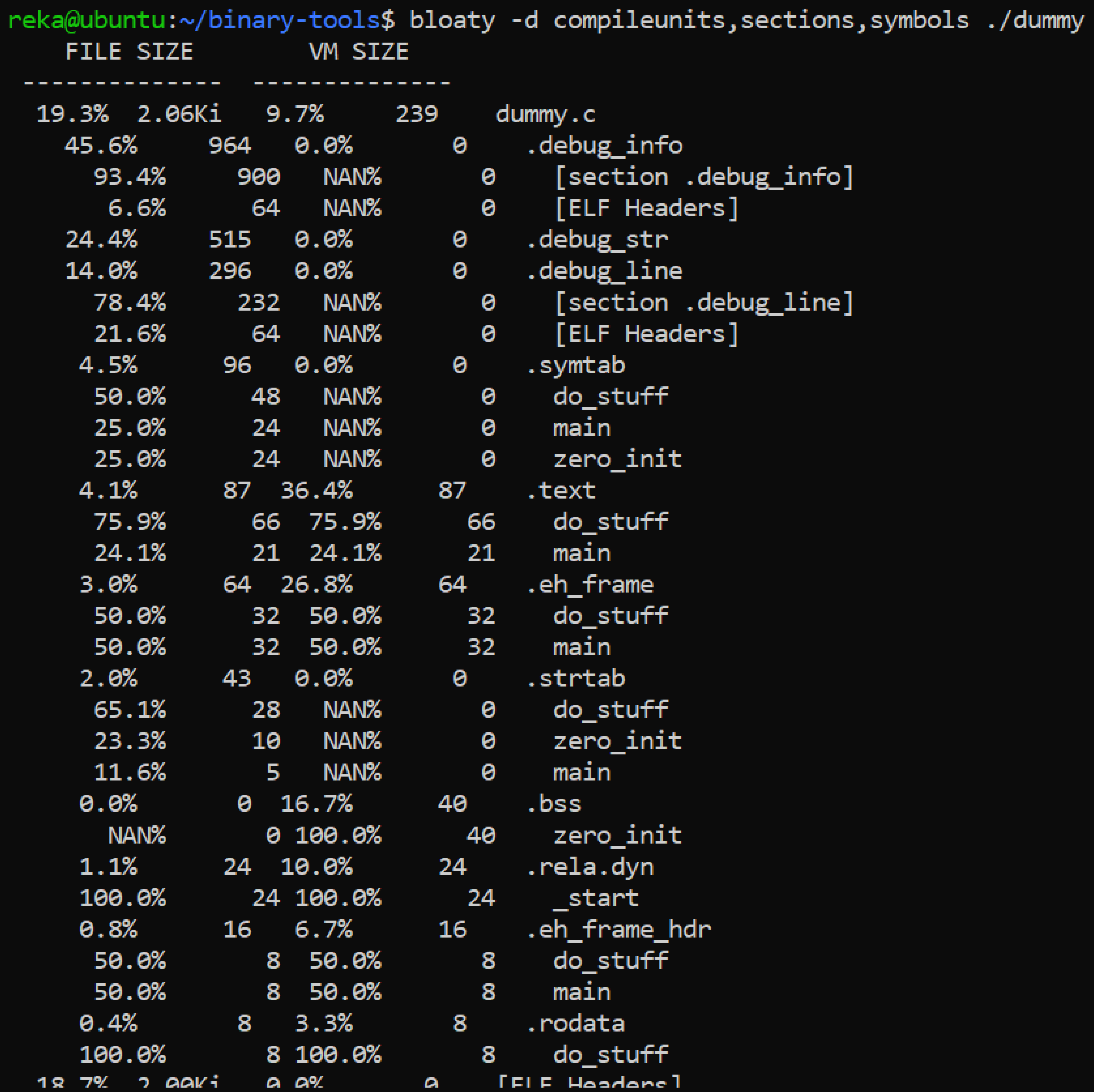
```
     2.9%      16   2.5%      16    .data
     2.9%      16   2.5%      16    .got
     0.0%       0   2.5%      16    [LOAD #3 [RW]]
     1.5%       8   1.3%       8    .fini_array
     1.5%       8   1.3%       8    .init_array
```

```
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

Track changes

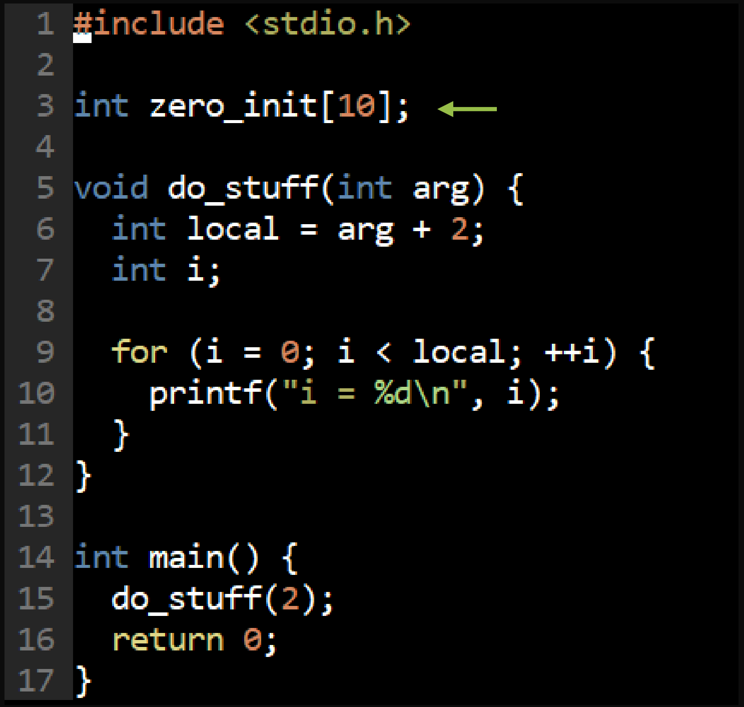


```c
#include <stdio.h>

int zero_init[10];

void do_stuff(int arg) {
  int local = arg + 2;
  int i;

  for (i = 0; i < local; ++i) {
    printf("i = %d\n", i);
  }
}

int main() {
  do_stuff(2);
  return 0;
}
```

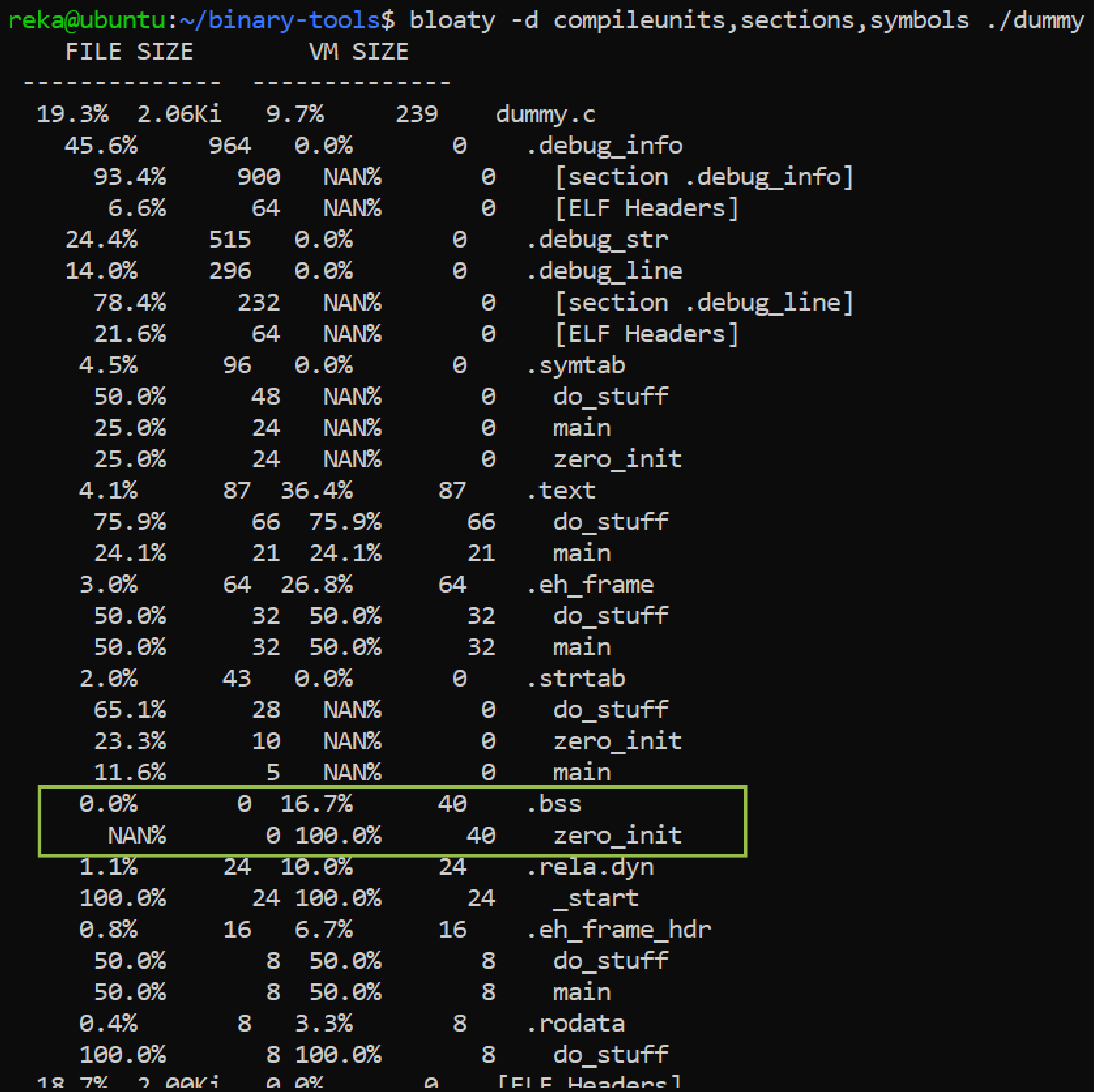


```
reka@ubuntu:~/binary-tools$ bloaty -d compileunits,sections,symbols ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  19.3%  2.06Ki   9.7%     239    dummy.c
    45.6%     964   0.0%       0    .debug_info
      93.4%     900   NAN%       0      [section .debug_info]
       6.6%      64   NAN%       0      [ELF Headers]
    24.4%     515   0.0%       0    .debug_str
    14.0%     296   0.0%       0    .debug_line
      78.4%     232   NAN%       0      [section .debug_line]
      21.6%      64   NAN%       0      [ELF Headers]
     4.5%      96   0.0%       0    .symtab
      50.0%      48   NAN%       0      do_stuff
      25.0%      24   NAN%       0      main
      25.0%      24   NAN%       0      zero_init
     4.1%      87  36.4%      87    .text
      75.9%      66  75.9%      66      do_stuff
      24.1%      21  24.1%      21      main
     3.0%      64  26.8%      64    .eh_frame
      50.0%      32  50.0%      32      do_stuff
      50.0%      32  50.0%      32      main
     2.0%      43   0.0%       0    .strtab
      65.1%      28   NAN%       0      do_stuff
      23.3%      10   NAN%       0      zero_init
      11.6%       5   NAN%       0      main
     0.0%       0  16.7%      40    .bss
       NAN%       0 100.0%      40      zero_init
     1.1%      24  10.0%      24    .rela.dyn
     100.0%      24 100.0%      24      _start
     0.8%      16   6.7%      16    .eh_frame_hdr
      50.0%       8  50.0%       8      do_stuff
      50.0%       8  50.0%       8      main
     0.4%       8   3.3%       8    .rodata
     100.0%       8 100.0%       8      do_stuff
  18.7%  2.00Ki   0.0%       0    [ELF Headers]
```

Track changes

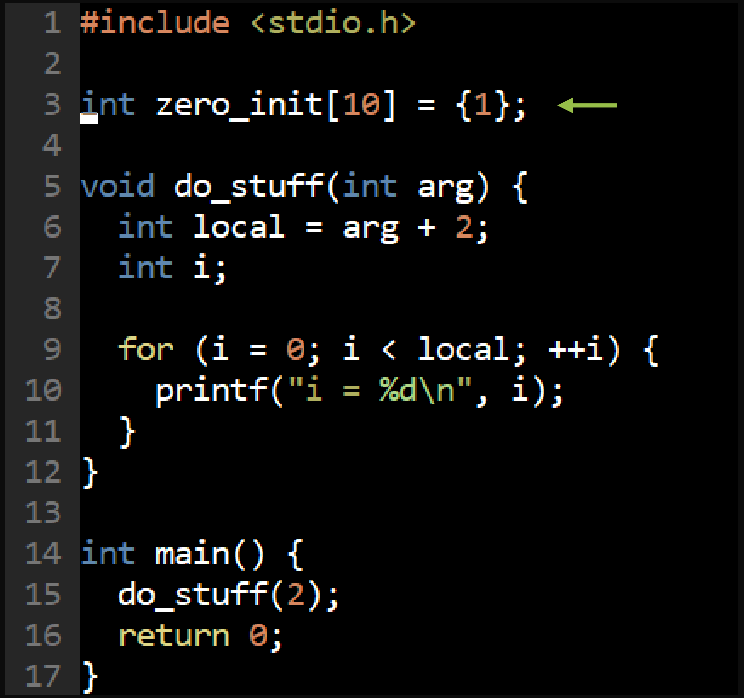


```c
#include <stdio.h>

int zero_init[10];

void do_stuff(int arg) {
  int local = arg + 2;
  int i;

  for (i = 0; i < local; ++i) {
    printf("i = %d\n", i);
  }
}

int main() {
  do_stuff(2);
  return 0;
}
```

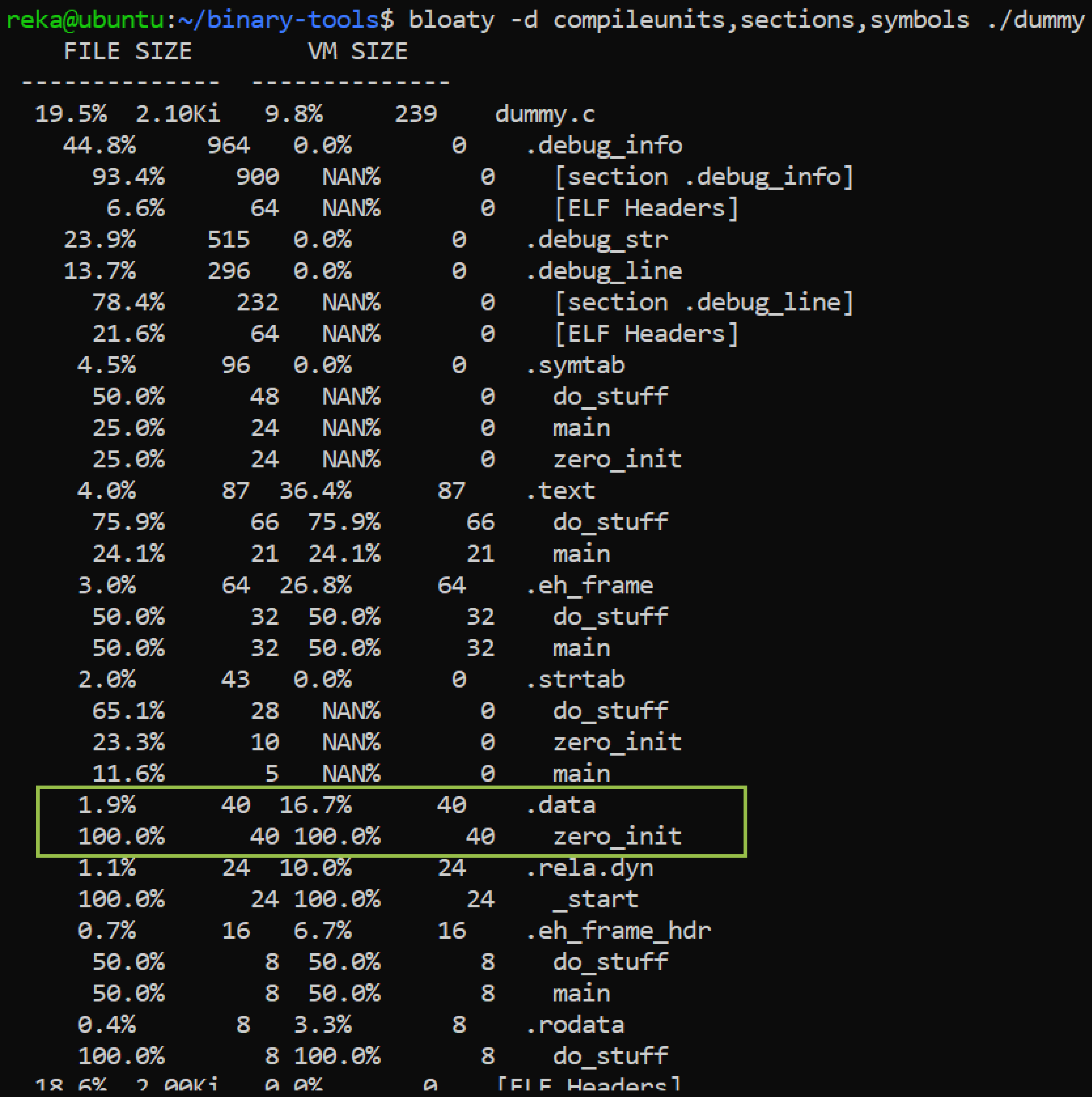


```
reka@ubuntu:~/binary-tools$ bloaty -d compileunits,sections,symbols ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  19.3%  2.06Ki   9.7%     239    dummy.c
    45.6%     964   0.0%       0    .debug_info
      93.4%     900   NAN%       0    [section .debug_info]
       6.6%      64   NAN%       0    [ELF Headers]
    24.4%     515   0.0%       0    .debug_str
    14.0%     296   0.0%       0    .debug_line
      78.4%     232   NAN%       0    [section .debug_line]
      21.6%      64   NAN%       0    [ELF Headers]
     4.5%      96   0.0%       0    .symtab
      50.0%      48   NAN%       0    do_stuff
      25.0%      24   NAN%       0    main
      25.0%      24   NAN%       0    zero_init
     4.1%      87  36.4%      87    .text
      75.9%      66  75.9%      66    do_stuff
      24.1%      21  24.1%      21    main
     3.0%      64  26.8%      64    .eh_frame
      50.0%      32  50.0%      32    do_stuff
      50.0%      32  50.0%      32    main
     2.0%      43   0.0%       0    .strtab
      65.1%      28   NAN%       0    do_stuff
      23.3%      10   NAN%       0    zero_init
      11.6%       5   NAN%       0    main
     0.0%       0  16.7%      40    .bss
       NAN%       0 100.0%      40    zero_init
     1.1%      24  10.0%      24    .rela.dyn
     100.0%      24 100.0%      24    _start
     0.8%      16   6.7%      16    .eh_frame_hdr
      50.0%       8  50.0%       8    do_stuff
      50.0%       8  50.0%       8    main
     0.4%       8   3.3%       8    .rodata
     100.0%       8 100.0%       8    do_stuff
  18.7%  2.00Ki   0.0%       0    [ELF Headers]
```

Track changes



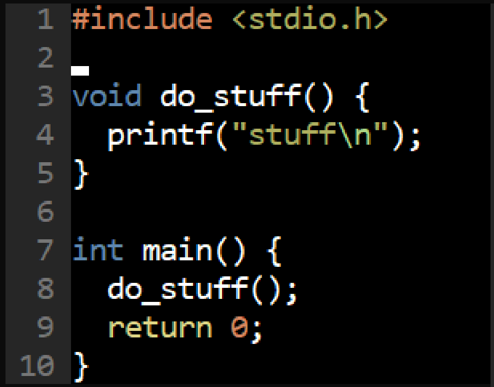
```c
#include <stdio.h>

int zero_init[10] = {1};

void do_stuff(int arg) {
  int local = arg + 2;
  int i;

  for (i = 0; i < local; ++i) {
    printf("i = %d\n", i);
  }
}

int main() {
  do_stuff(2);
  return 0;
}
```



```
reka@ubuntu:~/binary-tools$ bloaty -d compileunits,sections,symbols ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  19.5%  2.10Ki   9.8%     239    dummy.c
    44.8%     964   0.0%       0    .debug_info
      93.4%     900   NAN%       0    [section .debug_info]
       6.6%      64   NAN%       0    [ELF Headers]
    23.9%     515   0.0%       0    .debug_str
    13.7%     296   0.0%       0    .debug_line
      78.4%     232   NAN%       0    [section .debug_line]
      21.6%      64   NAN%       0    [ELF Headers]
     4.5%      96   0.0%       0    .symtab
      50.0%      48   NAN%       0    do_stuff
      25.0%      24   NAN%       0    main
      25.0%      24   NAN%       0    zero_init
     4.0%      87  36.4%      87    .text
      75.9%      66  75.9%      66    do_stuff
      24.1%      21  24.1%      21    main
     3.0%      64  26.8%      64    .eh_frame
      50.0%      32  50.0%      32    do_stuff
      50.0%      32  50.0%      32    main
     2.0%      43   0.0%       0    .strtab
      65.1%      28   NAN%       0    do_stuff
      23.3%      10   NAN%       0    zero_init
      11.6%       5   NAN%       0    main
     1.9%      40  16.7%      40    .data
     100.0%      40 100.0%      40    zero_init
     1.1%      24  10.0%      24    .rela.dyn
     100.0%      24 100.0%      24    _start
     0.7%      16   6.7%      16    .eh_frame_hdr
      50.0%       8  50.0%       8    do_stuff
      50.0%       8  50.0%       8    main
     0.4%       8   3.3%       8    .rodata
     100.0%       8 100.0%       8    do_stuff
  18.6%  2.00Ki   0.0%       0    [ELF Headers]
```

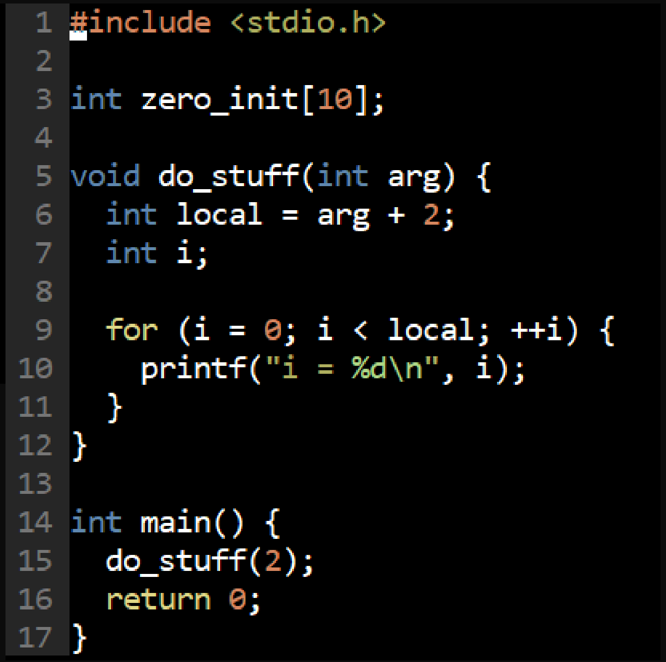
Diff mode

```
$ vim olddummy.c
```



```c
#include <stdio.h>

void do_stuff() {
  printf("stuff\n");
}

int main() {
  do_stuff();
  return 0;
}
```

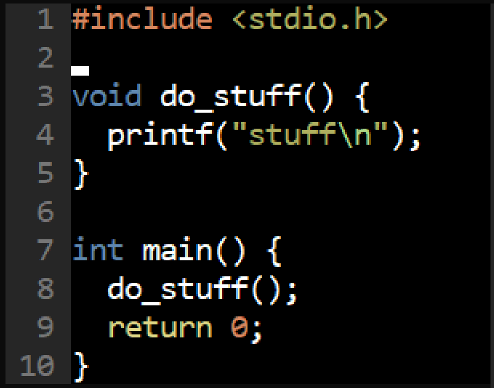
```
$ vim dummy.c
```



```c
#include <stdio.h>

int zero_init[10];

void do_stuff(int arg) {
  int local = arg + 2;
  int i;

  for (i = 0; i < local; ++i) {
    printf("i = %d\n", i);
  }
}

int main() {
  do_stuff(2);
  return 0;
}
```

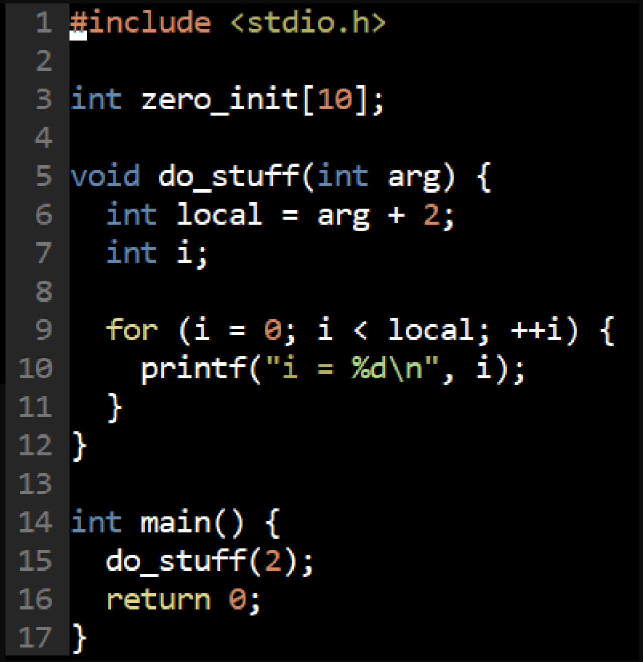
Diff mode

```
$ vim olddummy.c
```

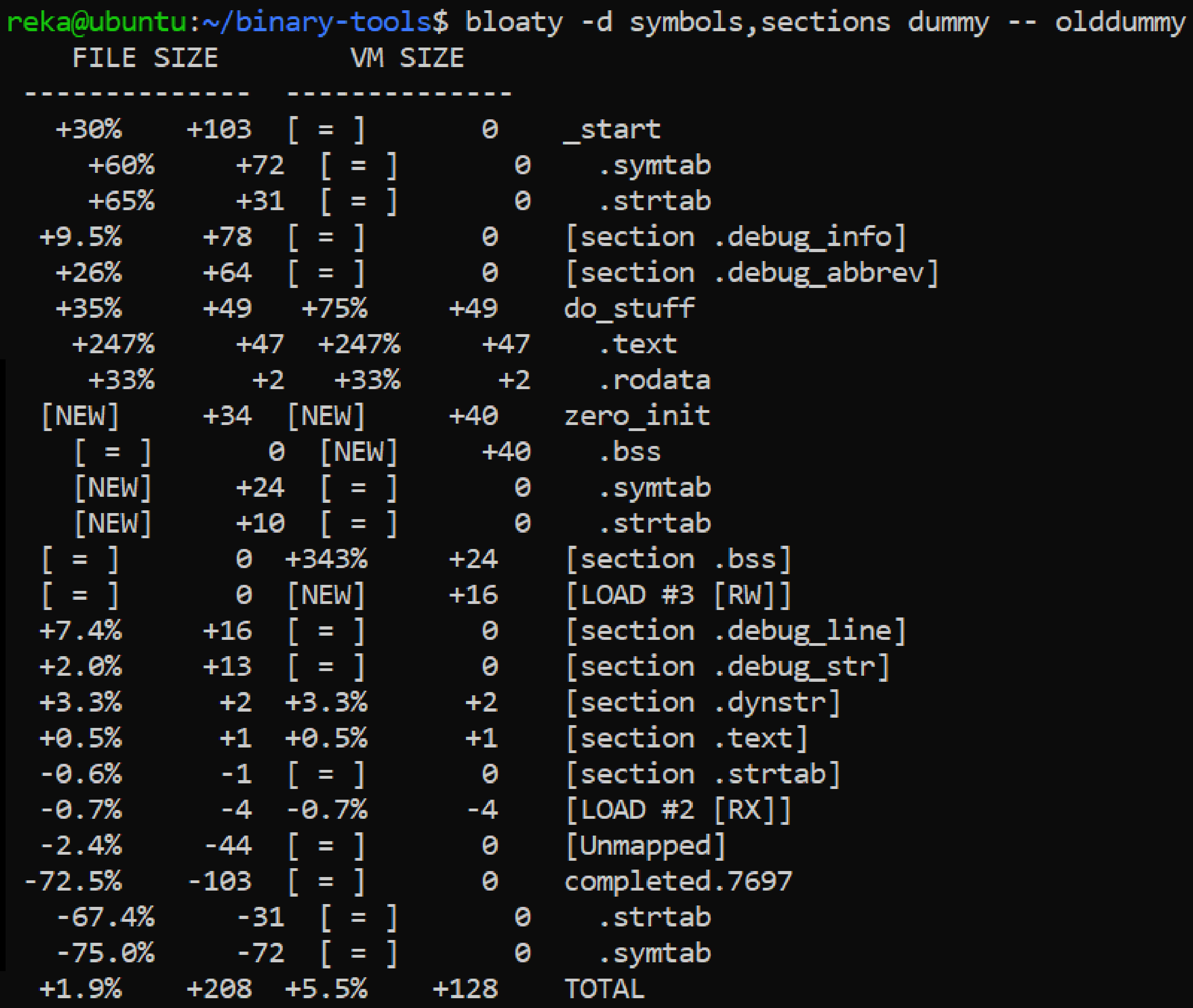


```c
#include <stdio.h>

void do_stuff() {
  printf("stuff\n");
}

int main() {
  do_stuff();
  return 0;
}
```

```
$ vim dummy.c
```



```c
#include <stdio.h>

int zero_init[10];

void do_stuff(int arg) {
  int local = arg + 2;
  int i;

  for (i = 0; i < local; ++i) {
    printf("i = %d\n", i);
  }
}

int main() {
  do_stuff(2);
  return 0;
}
```



```
reka@ubuntu:~/binary-tools$ bloaty -d symbols,sections dummy -- olddummy
    FILE SIZE        VM SIZE
 -------------- --------------
   +30%    +103  [ = ]       0    _start
     +60%     +72  [ = ]       0      .symtab
     +65%     +31  [ = ]       0      .strtab
  +9.5%     +78  [ = ]       0    [section .debug_info]
   +26%     +64  [ = ]       0    [section .debug_abbrev]
   +35%     +49   +75%     +49    do_stuff
    +247%     +47  +247%     +47      .text
     +33%      +2   +33%      +2      .rodata
  [NEW]     +34  [NEW]     +40    zero_init
    [ = ]       0  [NEW]     +40      .bss
    [NEW]     +24  [ = ]       0      .symtab
    [NEW]     +10  [ = ]       0      .strtab
  [ = ]       0  +343%     +24    [section .bss]
  [ = ]       0  [NEW]     +16    [LOAD #3 [RW]]
  +7.4%     +16  [ = ]       0    [section .debug_line]
  +2.0%     +13  [ = ]       0    [section .debug_str]
  +3.3%      +2  +3.3%      +2    [section .dynstr]
  +0.5%      +1  +0.5%      +1    [section .text]
  -0.6%      -1  [ = ]       0    [section .strtab]
  -0.7%      -4  -0.7%      -4    [LOAD #2 [RX]]
  -2.4%     -44  [ = ]       0    [Unmapped]
 -72.5%    -103  [ = ]       0    completed.7697
   -67.4%     -31  [ = ]       0      .strtab
   -75.0%     -72  [ = ]       0      .symtab
  +1.9%    +208  +5.5%    +128    TOTAL
```

Diff mode

```
$ vim olddummy.c
```

```c
#include <stdio.h>

void do_stuff() {
  printf("stuff\n");
}

int main() {
  do_stuff();
  return 0;
}
```

```
$ vim dummy.c
```

```c
#include <stdio.h>

int zero_init[10];

void do_stuff(int arg) {
  int local = arg + 2;
  int i;

  for (i = 0; i < local; ++i) {
    printf("i = %d\n", i);
  }
}

int main() {
  do_stuff(2);
  return 0;
}
```

```
reka@ubuntu:~/binary-tools$ bloaty -d symbols,sections dummy -- olddummy
    FILE SIZE        VM SIZE
 --------------  --------------
  +30%    +103  [ = ]       0    _start
    +60%     +72  [ = ]       0    .symtab
    +65%     +31  [ = ]       0    .strtab
 +9.5%     +78  [ = ]       0    [section .debug_info]
  +26%     +64  [ = ]       0    [section .debug_abbrev]
  +35%     +49  +75%     +49    do_stuff
   +247%     +47  +247%     +47    .text
    +33%      +2  +33%      +2    .rodata
 [NEW]     +34  [NEW]     +40    zero_init
   [ = ]       0  [NEW]     +40    .bss
   [NEW]     +24  [ = ]       0    .symtab
   [NEW]     +10  [ = ]       0    .strtab
 [ = ]       0  +343%     +24    [section .bss]
 [ = ]       0  [NEW]     +16    [LOAD #3 [RW]]
 +7.4%     +16  [ = ]       0    [section .debug_line]
 +2.0%     +13  [ = ]       0    [section .debug_str]
 +3.3%      +2  +3.3%      +2    [section .dynstr]
 +0.5%      +1  +0.5%      +1    [section .text]
 -0.6%      -1  [ = ]       0    [section .strtab]
 -0.7%      -4  -0.7%      -4    [LOAD #2 [RX]]
 -2.4%     -44  [ = ]       0    [Unmapped]
-72.5%    -103  [ = ]       0    completed.7697
  -67.4%     -31  [ = ]       0    .strtab
  -75.0%     -72  [ = ]       0    .symtab
 +1.9%    +208  +5.5%    +128    TOTAL
```

Diff mode

```
$ vim olddummy.c
```

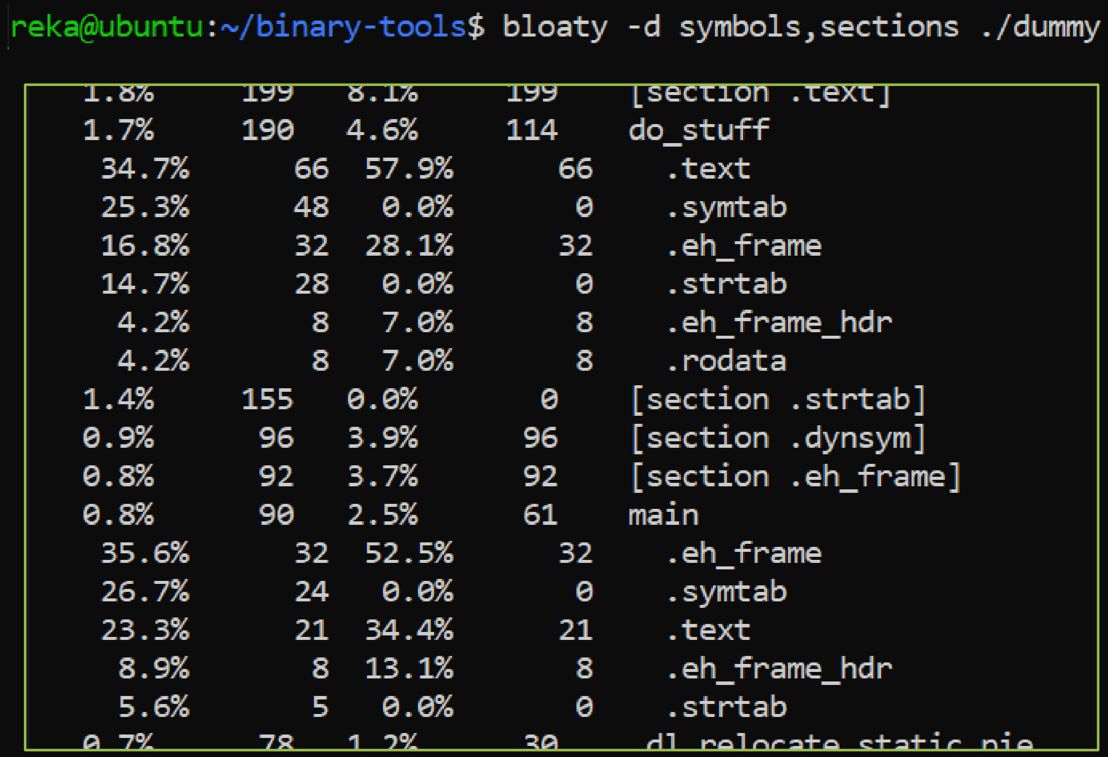
```c
#include <stdio.h>

void do_stuff() {
  printf("stuff\n");
}

int main() {
  do_stuff();
  return 0;
}
```

```
$ vim dummy.c
```

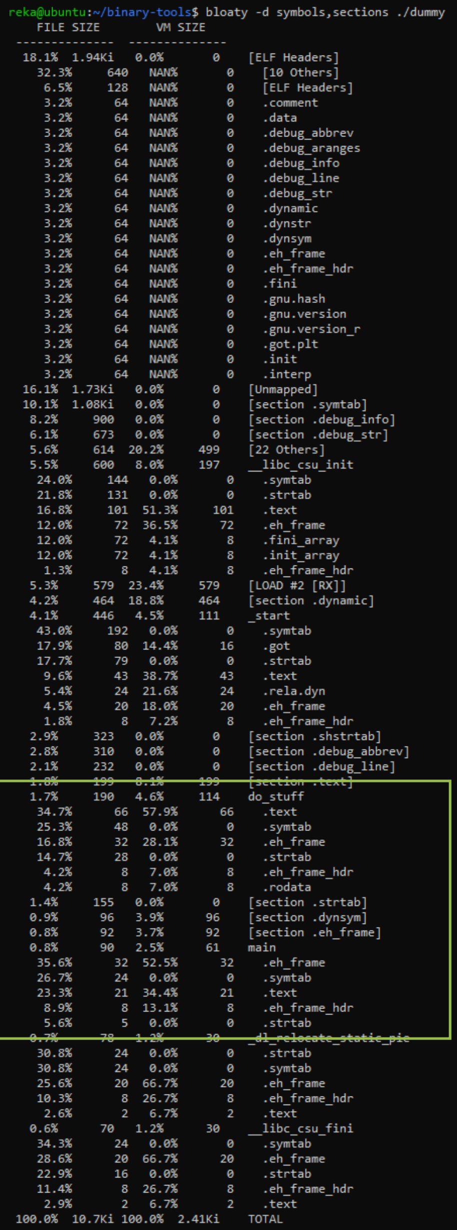
```c
#include <stdio.h>

int zero_init[10];

void do_stuff(int arg) {
  int local = arg + 2;
  int i;

  for (i = 0; i < local; ++i) {
    printf("i = %d\n", i);
  }
}

int main() {
  do_stuff(2);
  return 0;
}
```

```
reka@ubuntu:~/binary-tools$ bloaty -d symbols,sections dummy -- olddummy
    FILE SIZE        VM SIZE
 --------------  --------------
   +30%    +103  [ = ]       0    _start
     +60%     +72  [ = ]       0      .symtab
     +65%     +31  [ = ]       0      .strtab
  +9.5%     +78  [ = ]       0    [section .debug_info]
   +26%     +64  [ = ]       0    [section .debug_abbrev]
   +35%     +49  +75%     +49    do_stuff
    +247%     +47  +247%     +47      .text
     +33%      +2   +33%      +2      .rodata
  [NEW]     +34  [NEW]     +40    zero_init
    [ = ]       0  [NEW]     +40      .bss
    [NEW]     +24  [ = ]       0      .symtab
    [NEW]     +10  [ = ]       0      .strtab
  [ = ]       0  +343%     +24    [section .bss]
  [ = ]       0  [NEW]     +16    [LOAD #3 [RW]]
  +7.4%     +16  [ = ]       0    [section .debug_line]
  +2.0%     +13  [ = ]       0    [section .debug_str]
  +3.3%      +2  +3.3%      +2    [section .dynstr]
  +0.5%      +1  +0.5%      +1    [section .text]
  -0.6%      -1  [ = ]       0    [section .strtab]
  -0.7%      -4  -0.7%      -4    [LOAD #2 [RX]]
  -2.4%     -44  [ = ]       0    [Unmapped]
 -72.5%    -103  [ = ]       0    completed.7697
   -67.4%     -31  [ = ]       0      .strtab
   -75.0%     -72  [ = ]       0      .symtab
  +1.9%    +208  +5.5%    +128    TOTAL
```

Data Sources

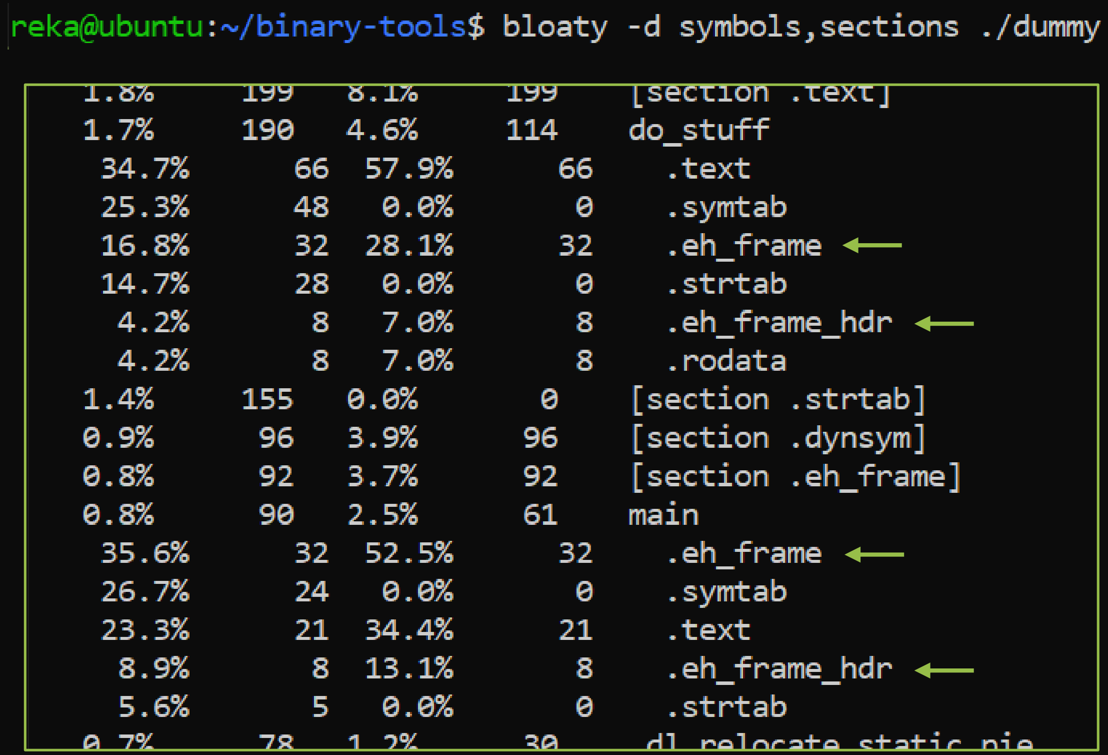


```
reka@ubuntu:~/binary-tools$ bloaty -d symbols,sections ./dummy
   1.8%     199   8.1%     199    [section .text]
   1.7%     190   4.6%     114    do_stuff
    34.7%      66  57.9%      66      .text
    25.3%      48   0.0%       0      .symtab
    16.8%      32  28.1%      32      .eh_frame
    14.7%      28   0.0%       0      .strtab
     4.2%       8   7.0%       8      .eh_frame_hdr
     4.2%       8   7.0%       8      .rodata
   1.4%     155   0.0%       0    [section .strtab]
   0.9%      96   3.9%      96    [section .dynsym]
   0.8%      92   3.7%      92    [section .eh_frame]
   0.8%      90   2.5%      61    main
    35.6%      32  52.5%      32      .eh_frame
    26.7%      24   0.0%       0      .symtab
    23.3%      21  34.4%      21      .text
     8.9%       8  13.1%       8      .eh_frame_hdr
     5.6%       5   0.0%       0      .strtab
   0.7%      78   1.2%      30     _dl_relocate_static_pie
```



```
reka@ubuntu:~/binary-tools$ bloaty -d symbols,sections ./dummy
    FILE SIZE        VM SIZE
 --------------  --------------
  18.1%  1.94Ki   0.0%       0    [ELF Headers]
    32.3%     640   NAN%       0      [10 Others]
     6.5%     128   NAN%       0      [ELF Headers]
     3.2%      64   NAN%       0      .comment
     3.2%      64   NAN%       0      .data
     3.2%      64   NAN%       0      .debug_abbrev
     3.2%      64   NAN%       0      .debug_aranges
     3.2%      64   NAN%       0      .debug_info
     3.2%      64   NAN%       0      .debug_line
     3.2%      64   NAN%       0      .debug_str
     3.2%      64   NAN%       0      .dynamic
     3.2%      64   NAN%       0      .dynstr
     3.2%      64   NAN%       0      .dynsym
     3.2%      64   NAN%       0      .eh_frame
     3.2%      64   NAN%       0      .eh_frame_hdr
     3.2%      64   NAN%       0      .fini
     3.2%      64   NAN%       0      .gnu.hash
     3.2%      64   NAN%       0      .gnu.version
     3.2%      64   NAN%       0      .gnu.version_r
     3.2%      64   NAN%       0      .got.plt
     3.2%      64   NAN%       0      .init
     3.2%      64   NAN%       0      .interp
  16.1%  1.73Ki   0.0%       0    [Unmapped]
  10.1%  1.08Ki   0.0%       0    [section .symtab]
   8.2%     900   0.0%       0    [section .debug_info]
   6.1%     673   0.0%       0    [section .debug_str]
   5.6%     614  20.2%     499    [22 Others]
   5.5%     600   8.0%     197    __libc_csu_init
    24.0%     144   0.0%       0      .symtab
    21.8%     131   0.0%       0      .strtab
    16.8%     101  51.3%     101      .text
    12.0%      72  36.5%      72      .eh_frame
    12.0%      72   4.1%       8      .fini_array
    12.0%      72   4.1%       8      .init_array
     1.3%       8   4.1%       8      .eh_frame_hdr
   5.3%     579  23.4%     579    [LOAD #2 [RX]]
   4.2%     464  18.8%     464    [section .dynamic]
   4.1%     446   4.5%     111    _start
    43.0%     192   0.0%       0      .symtab
    17.9%      80  14.4%      16      .got
    17.7%      79   0.0%       0      .strtab
     9.6%      43  38.7%      43      .text
     5.4%      24  21.6%      24      .rela.dyn
     4.5%      20  18.0%      20      .eh_frame
     1.8%       8   7.2%       8      .eh_frame_hdr
   2.9%     323   0.0%       0    [section .shstrtab]
   2.8%     310   0.0%       0    [section .debug_abbrev]
   2.1%     232   0.0%       0    [section .debug_line]
   1.8%     199   8.1%     199    [section .text]
   1.7%     190   4.6%     114    do_stuff
    34.7%      66  57.9%      66      .text
    25.3%      48   0.0%       0      .symtab
    16.8%      32  28.1%      32      .eh_frame
    14.7%      28   0.0%       0      .strtab
     4.2%       8   7.0%       8      .eh_frame_hdr
     4.2%       8   7.0%       8      .rodata
   1.4%     155   0.0%       0    [section .strtab]
   0.9%      96   3.9%      96    [section .dynsym]
   0.8%      92   3.7%      92    [section .eh_frame]
   0.8%      90   2.5%      61    main
    35.6%      32  52.5%      32      .eh_frame
    26.7%      24   0.0%       0      .symtab
    23.3%      21  34.4%      21      .text
     8.9%       8  13.1%       8      .eh_frame_hdr
     5.6%       5   0.0%       0      .strtab
   0.7%      78   1.2%      30    _dl_relocate_static_pie
    30.8%      24   0.0%       0      .strtab
    30.8%      24   0.0%       0      .symtab
    25.6%      20  66.7%      20      .eh_frame
    10.3%       8  26.7%       8      .eh_frame_hdr
     2.6%       2   6.7%       2      .text
   0.6%      70   1.2%      30    __libc_csu_fini
    34.3%      24   0.0%       0      .symtab
    28.6%      20  66.7%      20      .eh_frame
    22.9%      16   0.0%       0      .strtab
    11.4%       8  26.7%       8      .eh_frame_hdr
     2.9%       2   6.7%       2      .text
 100.0%  10.7Ki 100.0%  2.41Ki    TOTAL
```

Data Sources



```
reka@ubuntu:~/binary-tools$ bloaty -d symbols,sections ./dummy
```

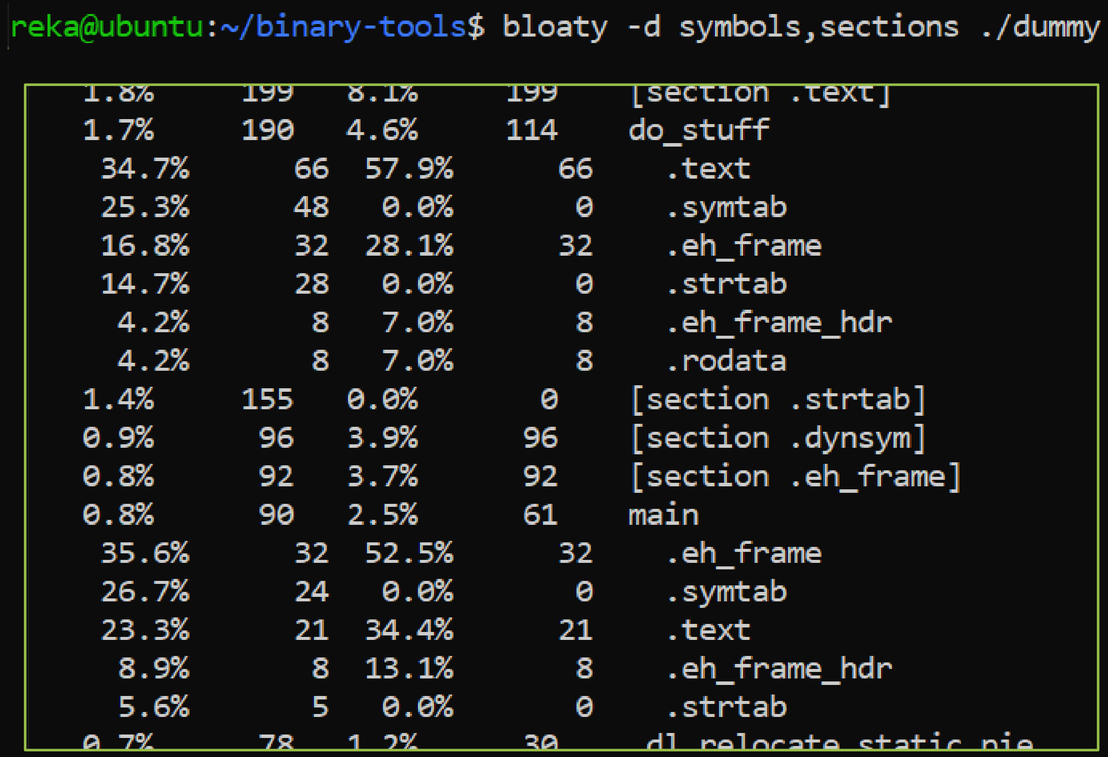
```
     1.8%     199   8.1%     199   [section .text]
     1.7%     190   4.6%     114   do_stuff
      34.7%      66  57.9%      66     .text
      25.3%      48   0.0%       0     .symtab
      16.8%      32  28.1%      32     .eh_frame ←
      14.7%      28   0.0%       0     .strtab
       4.2%       8   7.0%       8     .eh_frame_hdr ←
       4.2%       8   7.0%       8     .rodata
     1.4%     155   0.0%       0   [section .strtab]
     0.9%      96   3.9%      96   [section .dynsym]
     0.8%      92   3.7%      92   [section .eh_frame]
     0.8%      90   2.5%      61   main
      35.6%      32  52.5%      32     .eh_frame ←
      26.7%      24   0.0%       0     .symtab
      23.3%      21  34.4%      21     .text
       8.9%       8  13.1%       8     .eh_frame_hdr ←
       5.6%       5   0.0%       0     .strtab
     0.7%      78   1.2%      30    _dl_relocate_static_pie
```

The symbol table only refers to the machine code of the function

Other artifacts a function emits into the binary:

- Unwind info

Data Sources



```
reka@ubuntu:~/binary-tools$ bloaty -d symbols,sections ./dummy
```

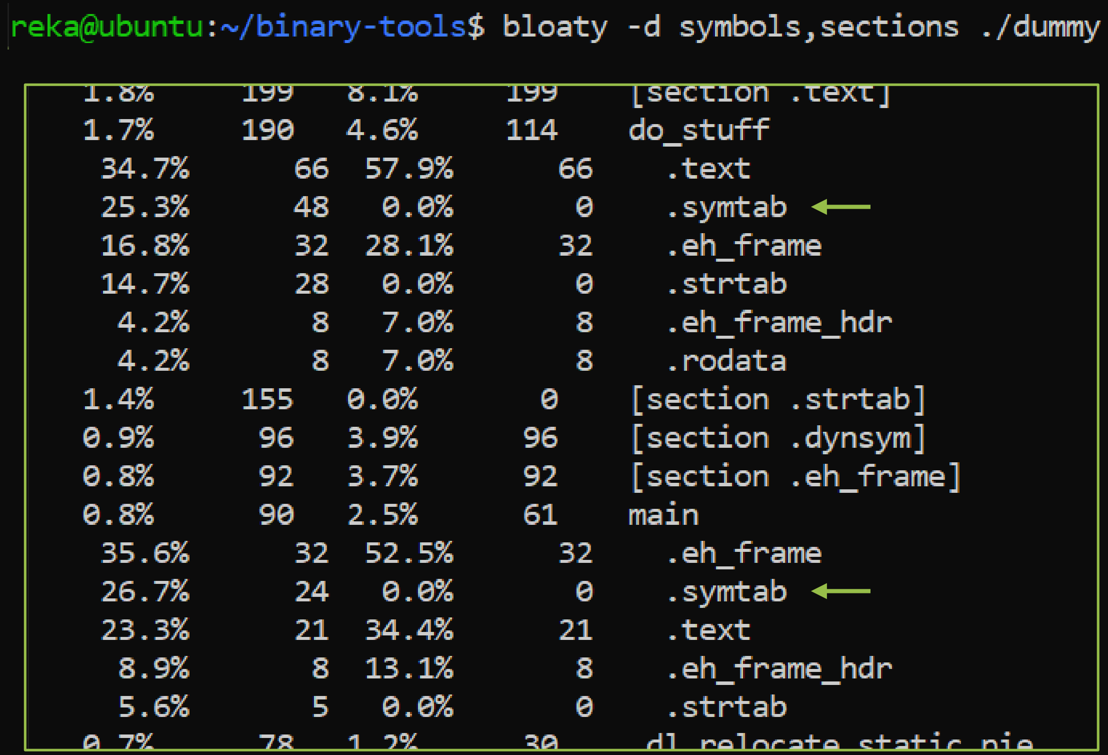
```
   1.8%     199   8.1%     199   [section .text]
   1.7%     190   4.6%     114   do_stuff
    34.7%      66  57.9%      66     .text
    25.3%      48   0.0%       0     .symtab
    16.8%      32  28.1%      32     .eh_frame
    14.7%      28   0.0%       0     .strtab
     4.2%       8   7.0%       8     .eh_frame_hdr
     4.2%       8   7.0%       8     .rodata
   1.4%     155   0.0%       0   [section .strtab]
   0.9%      96   3.9%      96   [section .dynsym]
   0.8%      92   3.7%      92   [section .eh_frame]
   0.8%      90   2.5%      61   main
    35.6%      32  52.5%      32     .eh_frame
    26.7%      24   0.0%       0     .symtab
    23.3%      21  34.4%      21     .text
     8.9%       8  13.1%       8     .eh_frame_hdr
     5.6%       5   0.0%       0     .strtab
   0.7%      78   1.2%      30    _dl_relocate_static_pie
```

The symbol table only refers to the machine code of the function

Other artifacts a function emits into the binary:

- Unwind info
- Relocation info
- Debug info

Data Sources



```
reka@ubuntu:~/binary-tools$ bloaty -d symbols,sections ./dummy
```

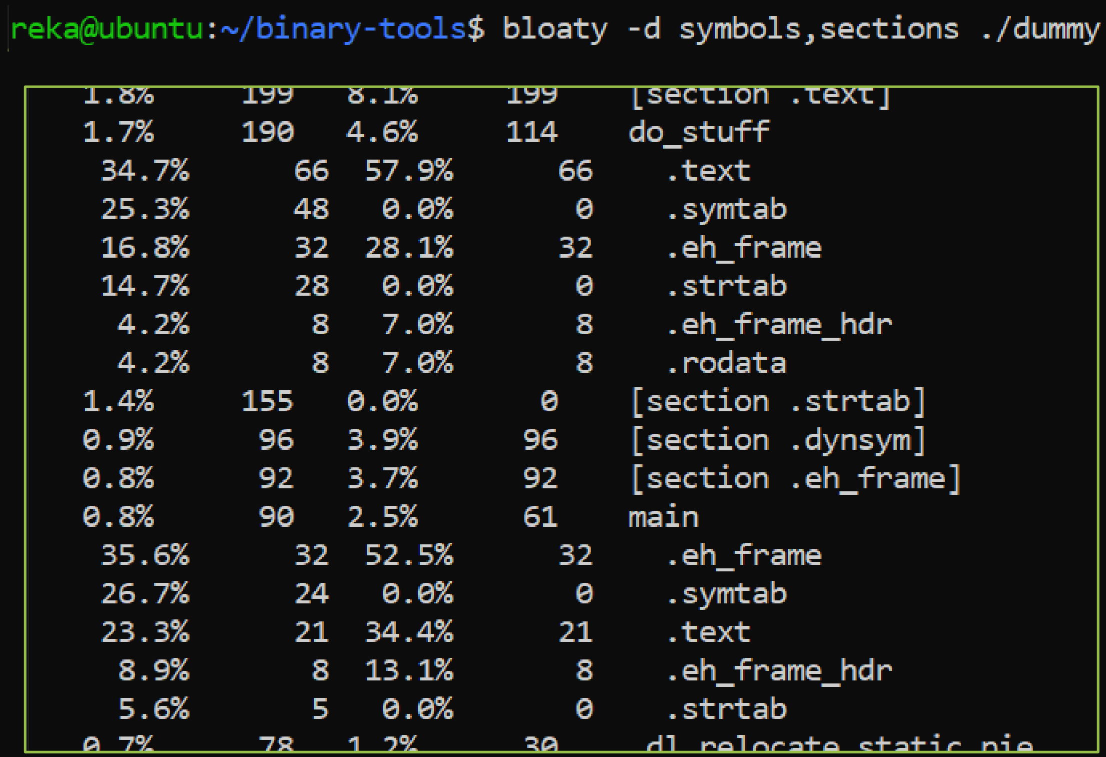
```
   1.8%     199   8.1%     199    [section .text]
   1.7%     190   4.6%     114    do_stuff
    34.7%      66  57.9%      66      .text
    25.3%      48   0.0%       0      .symtab  ←
    16.8%      32  28.1%      32      .eh_frame
    14.7%      28   0.0%       0      .strtab
     4.2%       8   7.0%       8      .eh_frame_hdr
     4.2%       8   7.0%       8      .rodata
   1.4%     155   0.0%       0    [section .strtab]
   0.9%      96   3.9%      96    [section .dynsym]
   0.8%      92   3.7%      92    [section .eh_frame]
   0.8%      90   2.5%      61    main
    35.6%      32  52.5%      32      .eh_frame
    26.7%      24   0.0%       0      .symtab  ←
    23.3%      21  34.4%      21      .text
     8.9%       8  13.1%       8      .eh_frame_hdr
     5.6%       5   0.0%       0      .strtab
   0.7%      78   1.2%      30     _dl_relocate_static_pie
```

The symbol table only refers to the machine code of the function

Other artifacts a function emits into the binary:

- Unwind info
- Relocation info
- Debug info
- Its entry in the symbol table itself

Data Sources



```
reka@ubuntu:~/binary-tools$ bloaty -d symbols,sections ./dummy
   1.8%     199   8.1%     199   [section .text]
   1.7%     190   4.6%     114   do_stuff
    34.7%      66  57.9%      66     .text
    25.3%      48   0.0%       0     .symtab
    16.8%      32  28.1%      32     .eh_frame
    14.7%      28   0.0%       0     .strtab
     4.2%       8   7.0%       8     .eh_frame_hdr
     4.2%       8   7.0%       8     .rodata
   1.4%     155   0.0%       0   [section .strtab]
   0.9%      96   3.9%      96   [section .dynsym]
   0.8%      92   3.7%      92   [section .eh_frame]
   0.8%      90   2.5%      61   main
    35.6%      32  52.5%      32     .eh_frame
    26.7%      24   0.0%       0     .symtab
    23.3%      21  34.4%      21     .text
     8.9%       8  13.1%       8     .eh_frame_hdr
     5.6%       5   0.0%       0     .strtab
   0.7%      78   1.2%      30    dl relocate static pie
```

The symbol table only refers to the machine code of the function

Other artifacts a function emits into the binary:

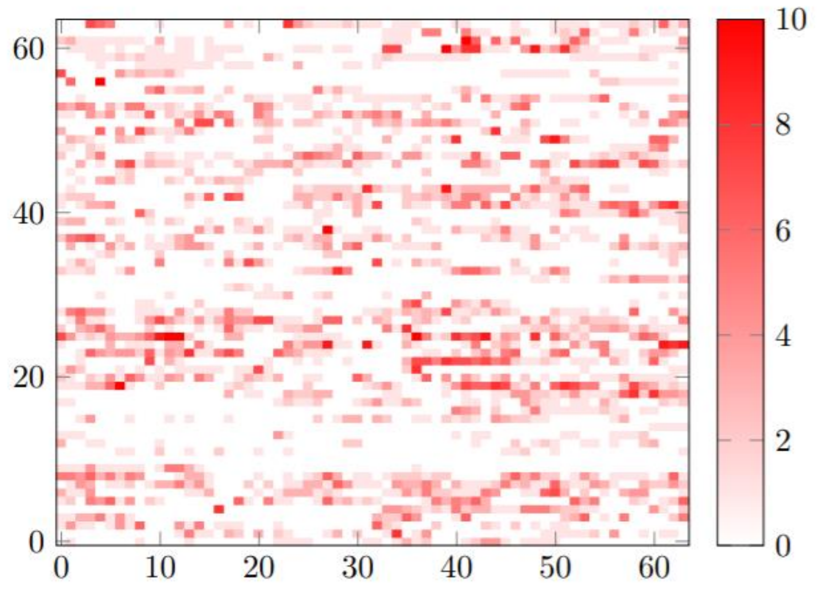
- Unwind info
- Relocation info
- Debug info
- Its entry in the symbol table itself

Bloaty attributes these to the function to get a faithful total size

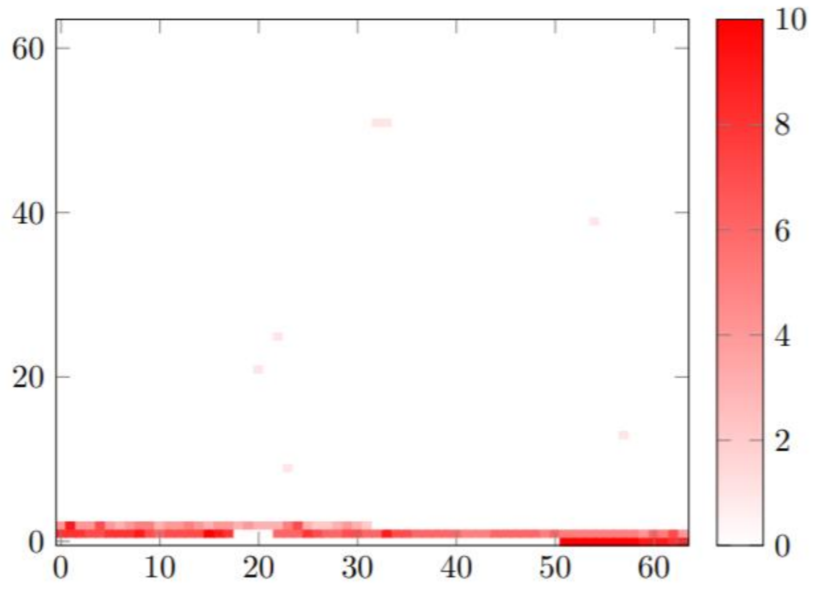
In practice, there is no 100% coverage – (Unmapped)

BOLT

- Maksim Panchenko et al., 2018
- https://github.com/facebookincubator/BOLT
- https://arxiv.org/abs/1807.06735
- Binary layout optimization
- Supports ELF x86-64, AArch64, uses LLVM libs & perf



(a) without BOLT



(b) with BOLT

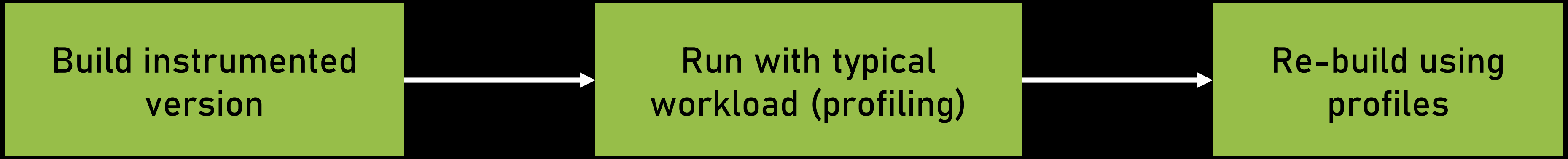
Heat maps for instruction memory accesses of the HHVM binary, without and with BOLT. Heat is in a log scale.

HHVM became 8% faster on top of compiler opts + link-time function layout tool

Clang & GCC binaries up to 20% faster with PGO+LTO+BOLT than with PGO+LTO

PGO (Profile-Guided Optimization)

Also POGO or FDO (Feedback-Directed Optimization)


Build instrumented version
Run with typical workload (profiling)
Re-build using profiles

- Inserting profile data early means that many optimizations can benefit from them
- CPU and memory overhead

LTO (Link-Time Optimization)

Also LTCG (Link-Time Code Generation)

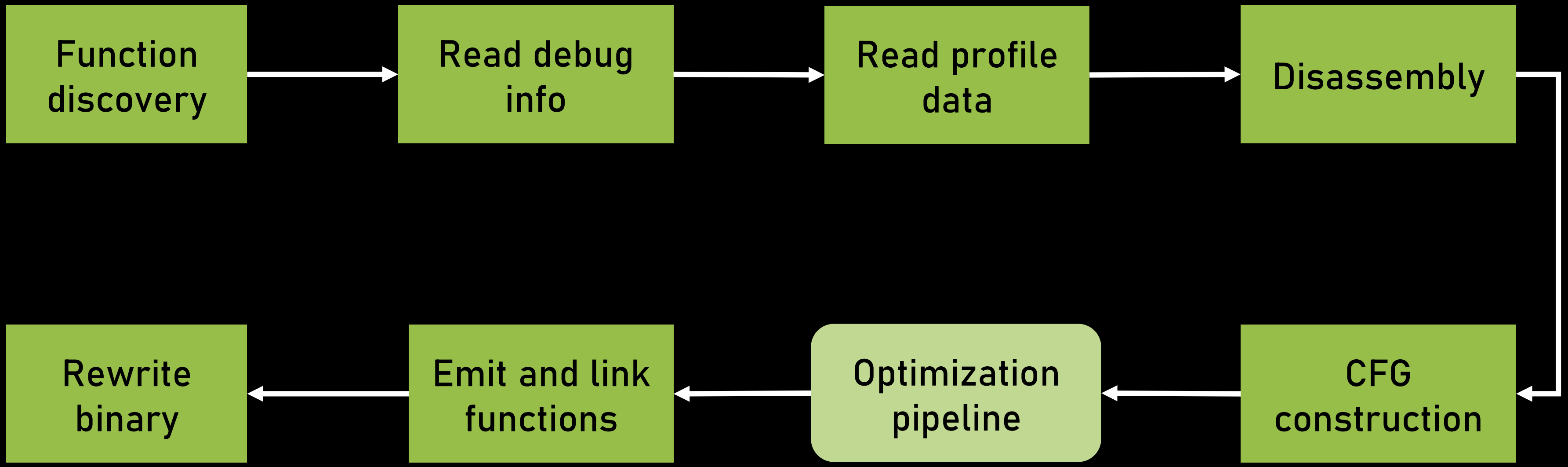
- Whole program analysis, cross-module optimization

BOLT

- Can optimize 3rd party libraries and assembly code
- Designed to work with the output of different compilers
- Uses sample-based profiling
  - Hardware profile counters e.g. Last Branch Records (Intel)
  - Negligible overhead
  - No special build required
- Binary-level profile data is applied on a binary level
  - No retrofitting needed, accurate

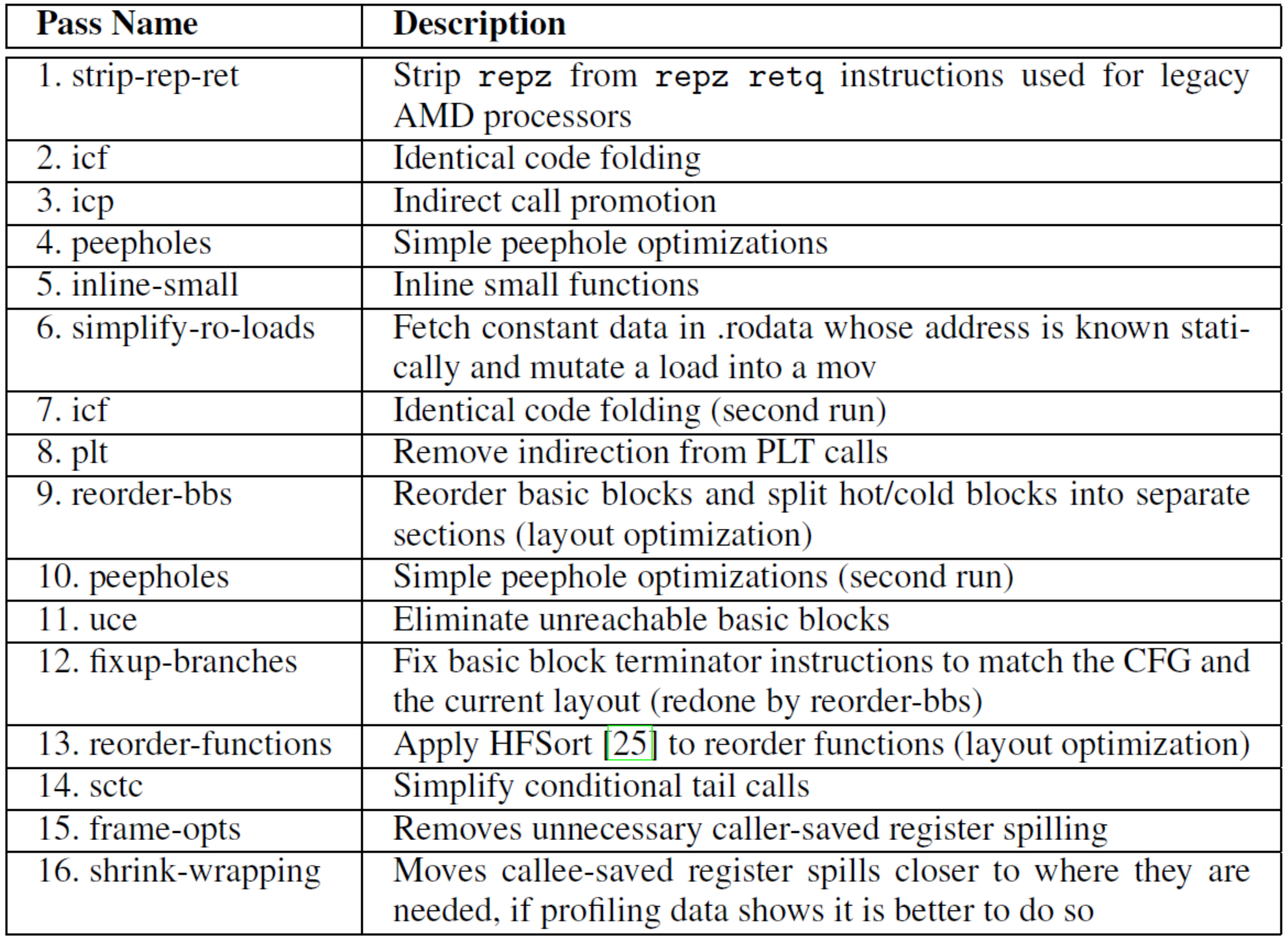
BOLT

Binary rewriting pipeline



BOLT

Optimization pipeline



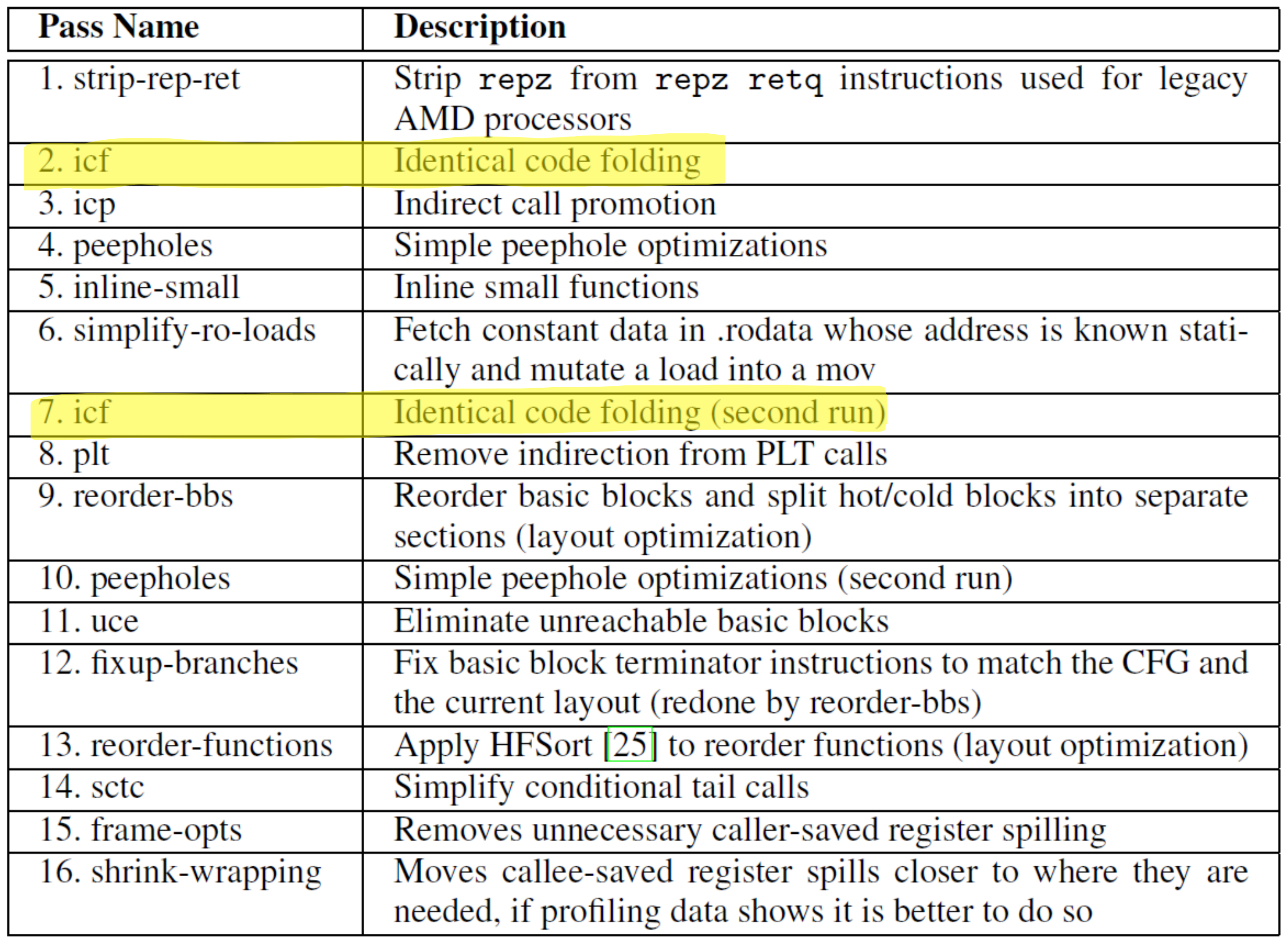
Pass Name	**Description**
1. strip-rep-ret	Strip `repz` from `repz retq` instructions used for legacy AMD processors
2. icf	Identical code folding
3. icp	Indirect call promotion
4. peepholes	Simple peephole optimizations
5. inline-small	Inline small functions
6. simplify-ro-loads	Fetch constant data in .rodata whose address is known statically and mutate a load into a mov
7. icf	Identical code folding (second run)
8. plt	Remove indirection from PLT calls
9. reorder-bbs	Reorder basic blocks and split hot/cold blocks into separate sections (layout optimization)
10. peepholes	Simple peephole optimizations (second run)
11. uce	Eliminate unreachable basic blocks
12. fixup-branches	Fix basic block terminator instructions to match the CFG and the current layout (redone by reorder-bbs)
13. reorder-functions	Apply HFSort [25] to reorder functions (layout optimization)
14. sctc	Simplify conditional tail calls
15. frame-opts	Removes unnecessary caller-saved register spilling
16. shrink-wrapping	Moves callee-saved register spills closer to where they are needed, if profiling data shows it is better to do so

BOLT

Optimization pipeline

Optimize functions compiled without `-ffunction-sections`

3% code size reduction for HHVM

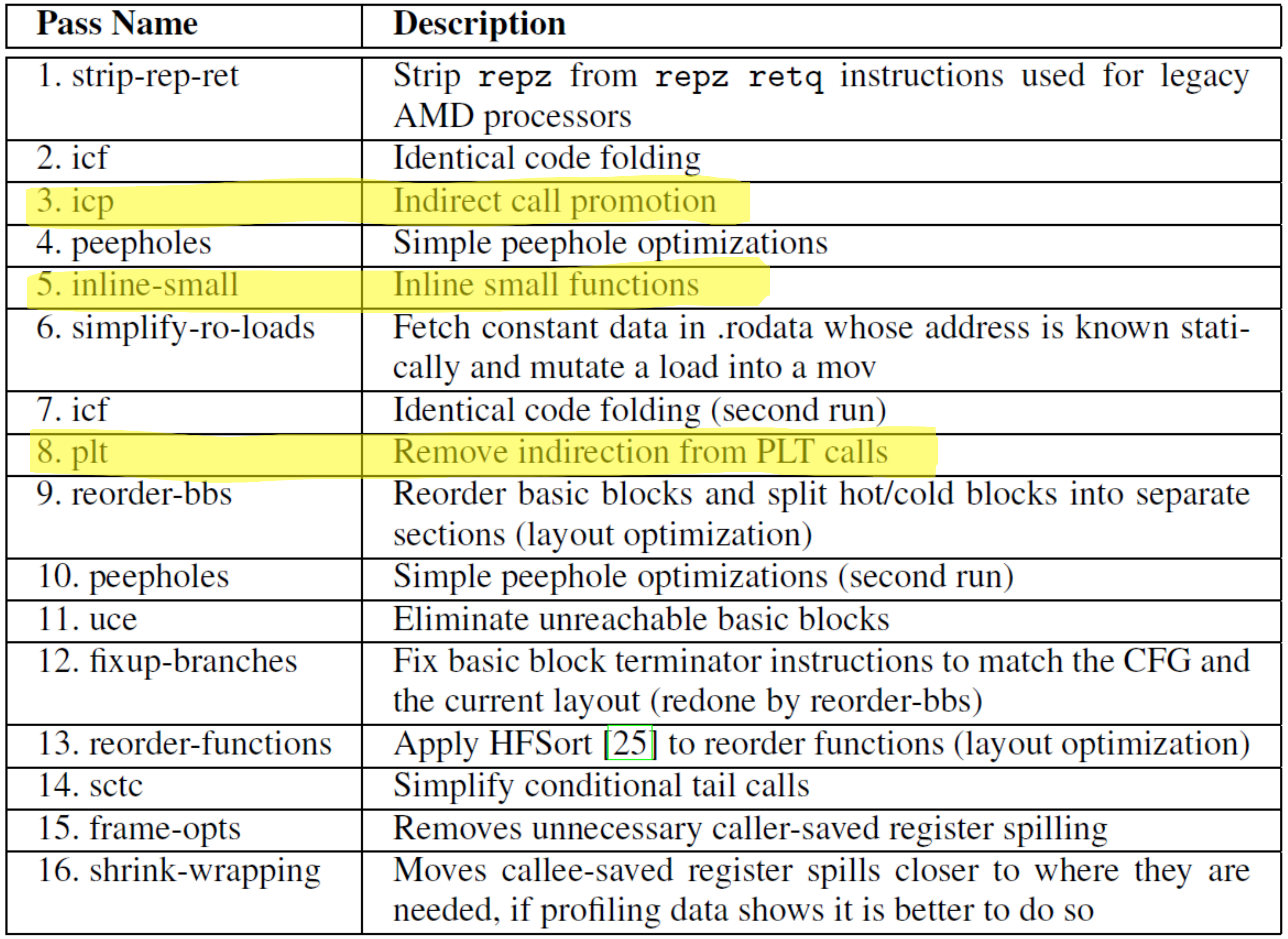


Pass Name	Description
1. strip-rep-ret	Strip `repz` from `repz retq` instructions used for legacy AMD processors
2. icf	Identical code folding
3. icp	Indirect call promotion
4. peepholes	Simple peephole optimizations
5. inline-small	Inline small functions
6. simplify-ro-loads	Fetch constant data in .rodata whose address is known statically and mutate a load into a mov
7. icf	Identical code folding (second run)
8. plt	Remove indirection from PLT calls
9. reorder-bbs	Reorder basic blocks and split hot/cold blocks into separate sections (layout optimization)
10. peepholes	Simple peephole optimizations (second run)
11. uce	Eliminate unreachable basic blocks
12. fixup-branches	Fix basic block terminator instructions to match the CFG and the current layout (redone by reorder-bbs)
13. reorder-functions	Apply HFSort [25] to reorder functions (layout optimization)
14. sctc	Simplify conditional tail calls
15. frame-opts	Removes unnecessary caller-saved register spilling
16. shrink-wrapping	Moves callee-saved register spills closer to where they are needed, if profiling data shows it is better to do so

BOLT

Optimization pipeline

Use call frequency information to eliminate or change calls

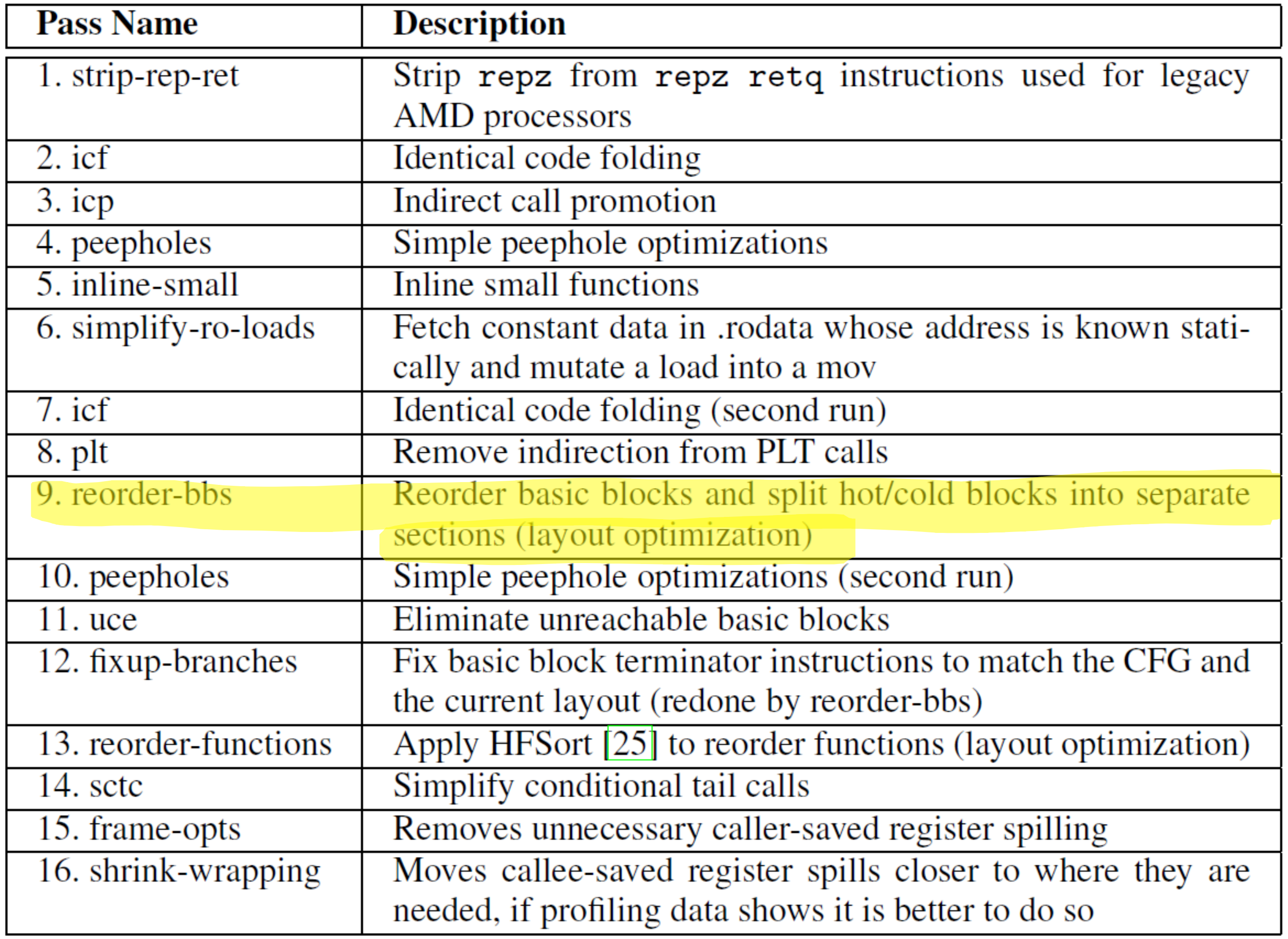


Pass Name	Description
1. strip-rep-ret	Strip `repz` from `repz retq` instructions used for legacy AMD processors
2. icf	Identical code folding
3. icp	Indirect call promotion
4. peepholes	Simple peephole optimizations
5. inline-small	Inline small functions
6. simplify-ro-loads	Fetch constant data in .rodata whose address is known statically and mutate a load into a mov
7. icf	Identical code folding (second run)
8. plt	Remove indirection from PLT calls
9. reorder-bbs	Reorder basic blocks and split hot/cold blocks into separate sections (layout optimization)
10. peepholes	Simple peephole optimizations (second run)
11. uce	Eliminate unreachable basic blocks
12. fixup-branches	Fix basic block terminator instructions to match the CFG and the current layout (redone by reorder-bbs)
13. reorder-functions	Apply HFSort [25] to reorder functions (layout optimization)
14. sctc	Simplify conditional tail calls
15. frame-opts	Removes unnecessary caller-saved register spilling
16. shrink-wrapping	Moves callee-saved register spills closer to where they are needed, if profiling data shows it is better to do so

BOLT

Optimization pipeline

Reduce the number of taken branches by making the hottest successor most likely a fall-though

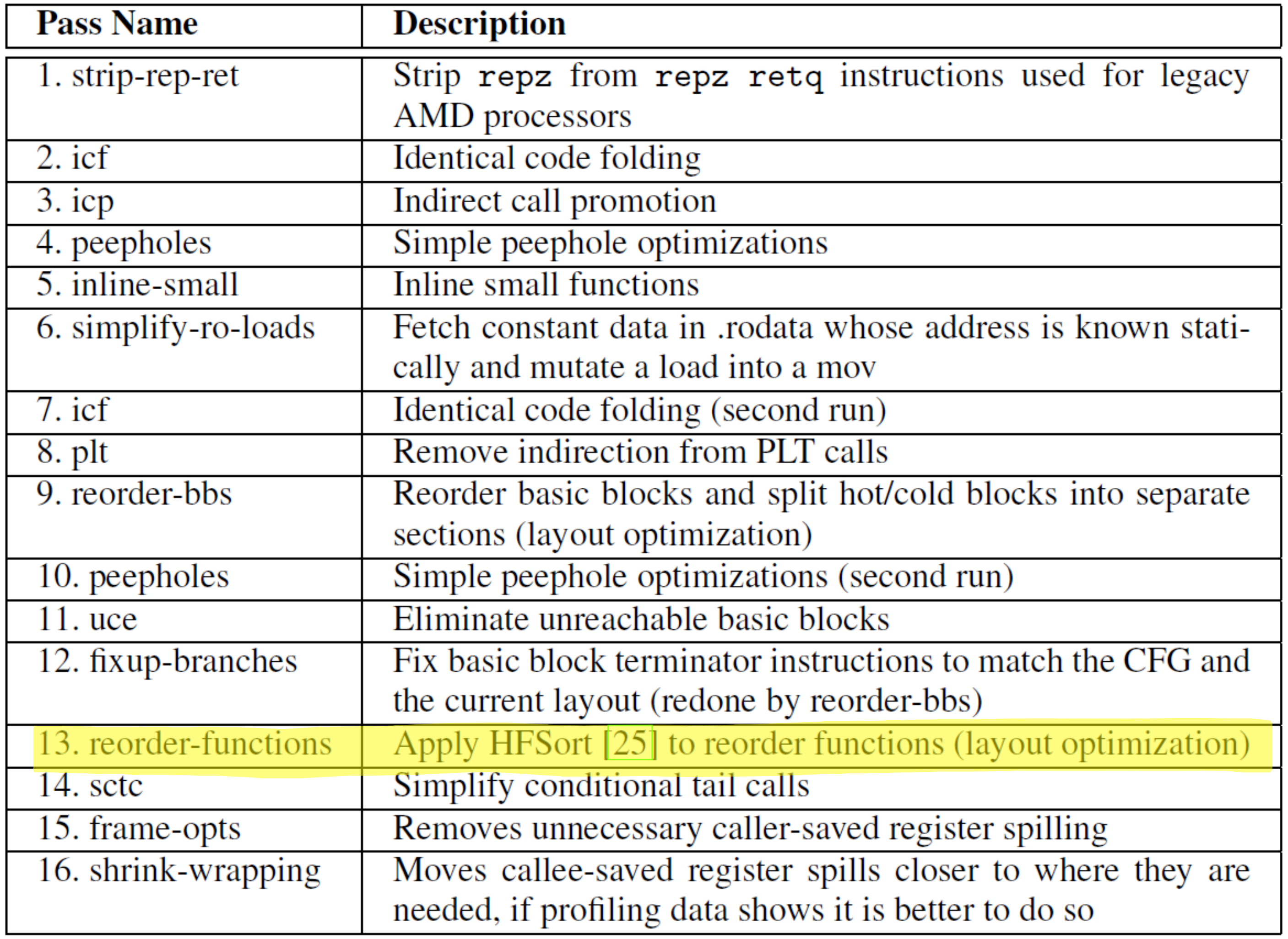


Pass Name	Description
1. strip-rep-ret	Strip `repz` from `repz retq` instructions used for legacy AMD processors
2. icf	Identical code folding
3. icp	Indirect call promotion
4. peepholes	Simple peephole optimizations
5. inline-small	Inline small functions
6. simplify-ro-loads	Fetch constant data in .rodata whose address is known statically and mutate a load into a mov
7. icf	Identical code folding (second run)
8. plt	Remove indirection from PLT calls
9. reorder-bbs	Reorder basic blocks and split hot/cold blocks into separate sections (layout optimization)
10. peepholes	Simple peephole optimizations (second run)
11. uce	Eliminate unreachable basic blocks
12. fixup-branches	Fix basic block terminator instructions to match the CFG and the current layout (redone by reorder-bbs)
13. reorder-functions	Apply HFSort [25] to reorder functions (layout optimization)
14. sctc	Simplify conditional tail calls
15. frame-opts	Removes unnecessary caller-saved register spilling
16. shrink-wrapping	Moves callee-saved register spills closer to where they are needed, if profiling data shows it is better to do so

BOLT

Optimization pipeline

Uses a weighted call graph to do the reordering

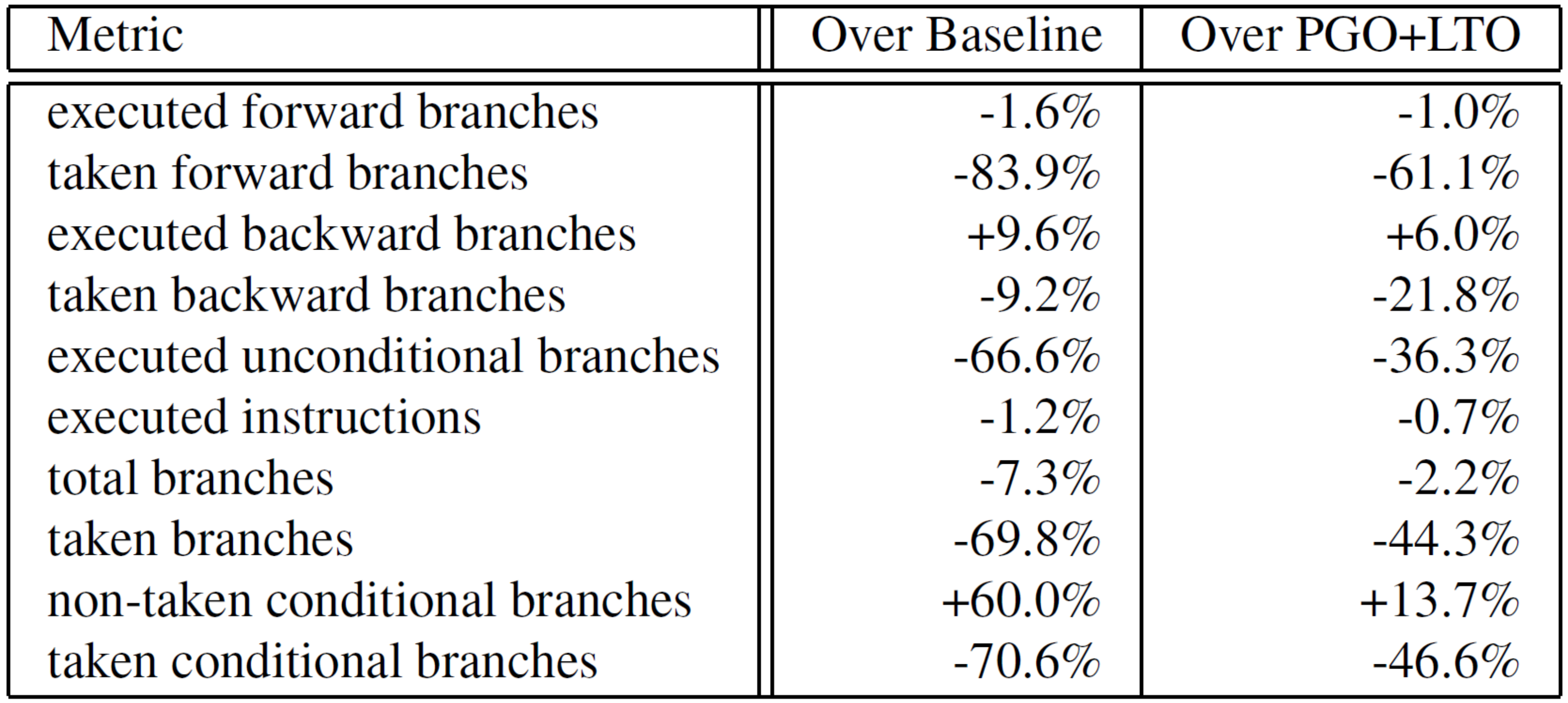


Pass Name	Description
1. strip-rep-ret	Strip `repz` from `repz retq` instructions used for legacy AMD processors
2. icf	Identical code folding
3. icp	Indirect call promotion
4. peepholes	Simple peephole optimizations
5. inline-small	Inline small functions
6. simplify-ro-loads	Fetch constant data in .rodata whose address is known statically and mutate a load into a mov
7. icf	Identical code folding (second run)
8. plt	Remove indirection from PLT calls
9. reorder-bbs	Reorder basic blocks and split hot/cold blocks into separate sections (layout optimization)
10. peepholes	Simple peephole optimizations (second run)
11. uce	Eliminate unreachable basic blocks
12. fixup-branches	Fix basic block terminator instructions to match the CFG and the current layout (redone by reorder-bbs)
13. reorder-functions	Apply HFSort [25] to reorder functions (layout optimization)
14. sctc	Simplify conditional tail calls
15. frame-opts	Removes unnecessary caller-saved register spilling
16. shrink-wrapping	Moves callee-saved register spills closer to where they are needed, if profiling data shows it is better to do so

Workflow

0) Have a binary that does not fit into the instruction cache
1) Build a copy of it with `-Wl,-q` (or pass `--emit-relocs` to the linker)
2) Collect execution profiles with `perf`
3) Run `llvm-bolt` with the output of `perf`
4) Diff running times
5) See reduced number of instruction cache misses with `perf`

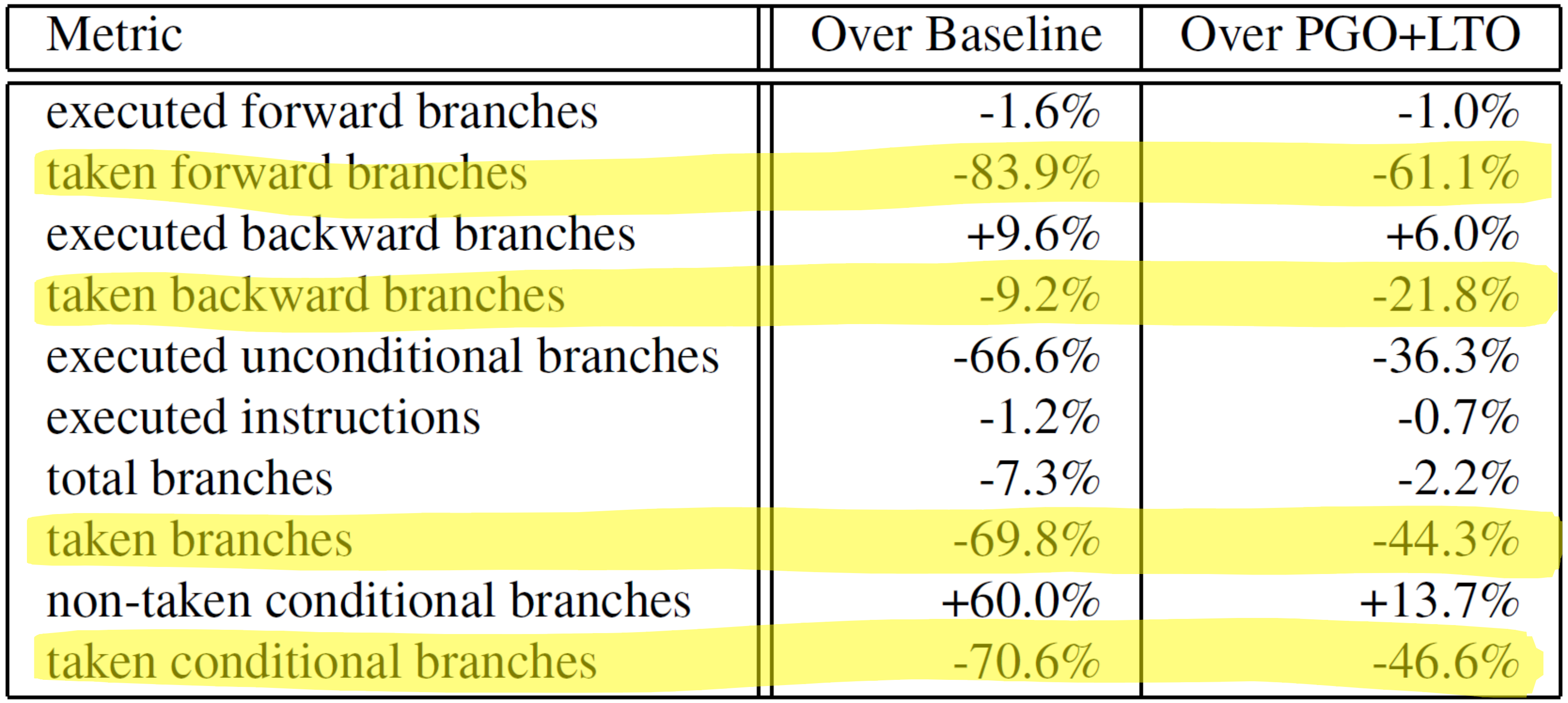
Clang binaries up to 20% faster with PGO+LTO+BOLT than with PGO+LTO



Metric	Over Baseline	Over PGO+LTO
executed forward branches	-1.6%	-1.0%
taken forward branches	-83.9%	-61.1%
executed backward branches	+9.6%	+6.0%
taken backward branches	-9.2%	-21.8%
executed unconditional branches	-66.6%	-36.3%
executed instructions	-1.2%	-0.7%
total branches	-7.3%	-2.2%
taken branches	-69.8%	-44.3%
non-taken conditional branches	+60.0%	+13.7%
taken conditional branches	-70.6%	-46.6%

Stats reported by BOLT when applied to Clang’s baseline and PGO+LTO binaries

Clang binaries up to 20% faster with PGO+LTO+BOLT than with PGO+LTO



Metric	Over Baseline	Over PGO+LTO
executed forward branches	-1.6%	-1.0%
taken forward branches	-83.9%	-61.1%
executed backward branches	+9.6%	+6.0%
taken backward branches	-9.2%	-21.8%
executed unconditional branches	-66.6%	-36.3%
executed instructions	-1.2%	-0.7%
total branches	-7.3%	-2.2%
taken branches	-69.8%	-44.3%
non-taken conditional branches	+60.0%	+13.7%
taken conditional branches	-70.6%	-46.6%

Stats reported by BOLT when applied to Clang's baseline and PGO+LTO binaries

Thanks!