Compiling Overview (0A)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

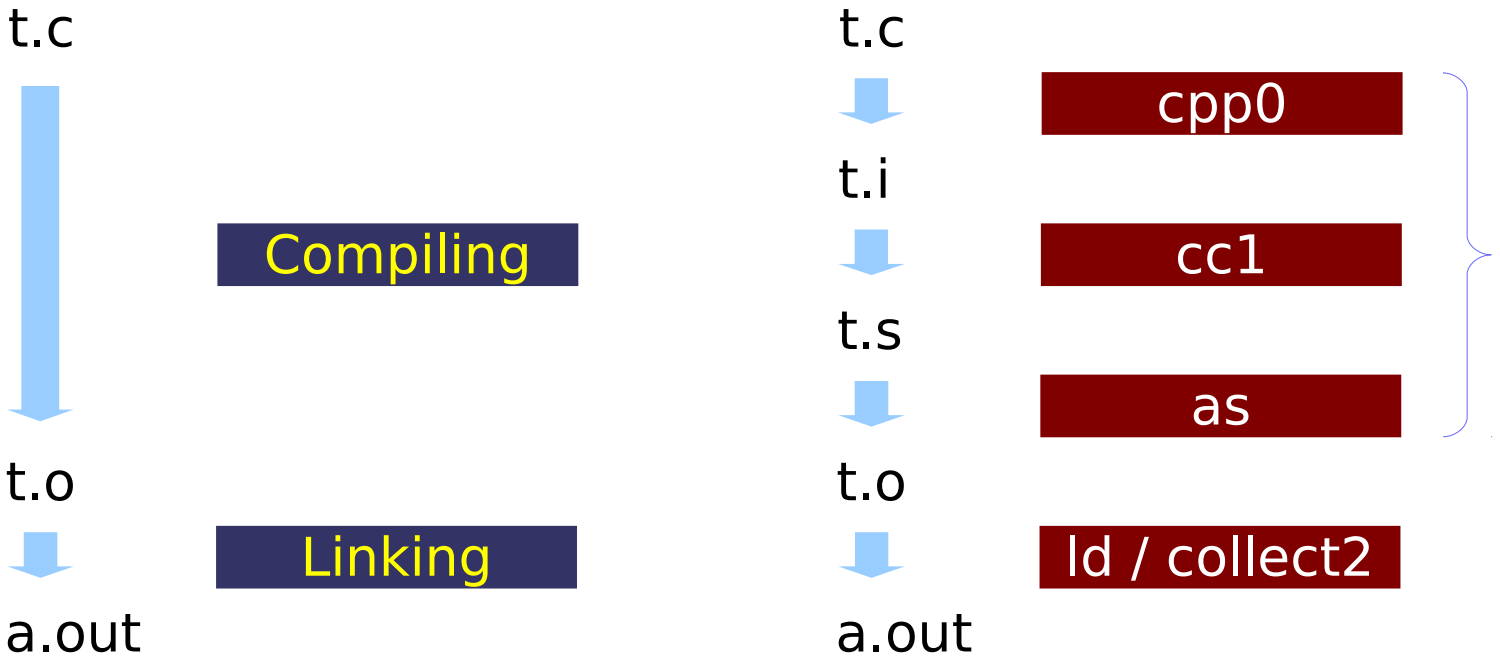
This document was produced by using OpenOffice.

Young Won Lim
10/20/16

GNU Compiler

- cpp0 pre processor
- cc1 **c** compiler
- cc1obj **objective-c** compiler
- cc1plus **c++** compiler
- f771 **fortran** compiler
- jc1 **java** compiler
- collect2 linker

GCC Compile Process



Example Files

main.h

```
int psum (int n) ;
```

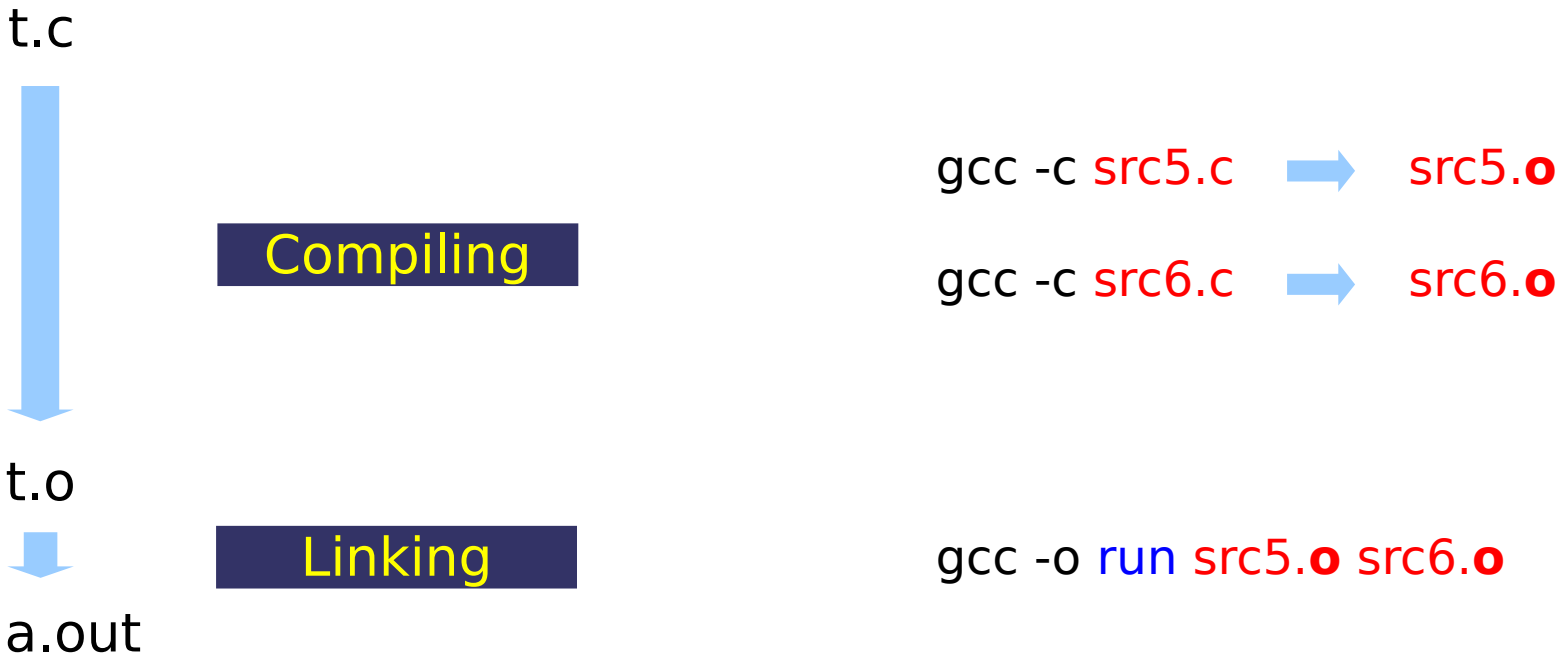
main.c

```
#include “main.h”

int main (void)
{
        int S1, S2, S3;

        S1 = psum ( 1 );
        printf(“S1 = %d \n”, S1);
        S2 = psum ( 2 );
        printf(“S2 = %d \n”, S2);
        S3 = psum ( 3 );
        printf(“S3 = %d \n”, S3);

        return 0;
}
```

psum.c

```
int psum (int n)
{
        int k, S = 0;
        for (k=1; k<=n; ++k) S += k;
        return S;
}
```

Compiling and Linking Examples



Preprocessor : cpp

To print preprocessed result on the screen

- gcc -E main.c
- cpp main.c

To preserve intermediate results
main.i preprocessor output file
main.s assembler output file

- gcc -c –-save-temps main.c

Preprocessor Output Example (1)

```
young@USBMTSYS2 ~ $ more main.c
// #include <stdio.h>
#include "main.h"


int main (void) {
  int S1, S2, S3;

  S1 = psum(1);
  printf("S1=%d \n", S1);
  S2 = psum(2);
  printf("S2=%d \n", S2);
  S3 = psum(3);
  printf("S3=%d \n", S3);

  return 0;
}
```

gcc -E main.c

```
young@USBMTSYS2 ~ $ gcc -E main.c
# 1 "main.c"
# 1 "<built-in>"
# 1 "<command-line>"
# 1 "/usr/include/stdc-predef.h" 1 3 4
# 1 "<command-line>" 2
# 1 "main.c"

# 1 "main.h" 1
int psum (int n);
# 3 "main.c" 2


int main (void) {
  int S1, S2, S3;

  S1 = psum(1);
  printf("S1=%d \n", S1);
  S2 = psum(2);
  printf("S2=%d \n", S2);
  S3 = psum(3);
  printf("S3=%d \n", S3);

  return 0;
}
```

Preprocessor Output Example (2)

```c
#include <stdio.h>
#include "main.h"


int main (void) {
  int S1, S2, S3;

  S1 = psum(1);
  printf("S1=%d \n", S1);
  S2 = psum(2);
  printf("S2=%d \n", S2);
  S3 = psum(3);
  printf("S3=%d \n", S3);

  return 0;
}
```

gcc -E main.c

```c
# 1 "<command-line>" 2
# 1 "main.c"
# 1 "/usr/include/stdio.h" 1 3 4
...
printf (_IO_FILE *__restrict, const char *__restrict,
    __gnuc_va_list);
...
extern int printf (const char *__restrict __format, ...);
...
extern int scanf (const char *__restrict __format, ...) ;
...
# 1 "main.h"
int psum (int n);
# 3 "main.c" 2


int main (void) {
  int S1, S2, S3;

  S1 = psum(1);
  printf("S1=%d \n", S1);
  S2 = psum(2);
  printf("S2=%d \n", S2);
  S3 = psum(3);
  printf("S3=%d \n", S3);

  return 0;
}
```

Generating Assembly Source

cc1 generates an assembly source

as generates an ELF object file

From source code (*.c)

- gcc -S -c main.c
- gcc -O2 -S -c main.c

Readable assembly listing with the source code intermixing

- gcc -c -g -Wa,-a,-ad main.c > main.lst
- gcc -c -g -Wa,-a,-ad [other GCC options] main.c > main.lst

From object code (*.o) with debug information

- objdump -d main.o > main.lst
- objdump -S -d main.o > main.lst (intermix with the source)

main.s listing : gcc -S -c main.c

```
        .file   "main.c"
        .section        .rodata
.LC0:
        .string "S1=%d \n"
.LC1:
        .string "S2=%d \n"
.LC2:
        .string "S3=%d \n"
        .text
        .globl  main
        .type   main, @function
main:
.LFB0:
        .cfi_startproc
        pushq  %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq   %rsp, %rbp
        .cfi_def_cfa_register 6
        subq    $16, %rsp
        movl    $1, %edi
        call    psum
        movl    %eax, -12(%rbp)
        movl    -12(%rbp), %eax
        movl    %eax, %esi
        movl    $.LC0, %edi
        movl    $0, %eax
        call    printf
        movl    $2, %edi
        call    psum
        movl    %eax, -8(%rbp)
        movl    -8(%rbp), %eax
        movl    %eax, %esi
        movl    $.LC1, %edi
        movl    $0, %eax
        call    printf
        movl    $3, %edi
        call    psum
        movl    %eax, -4(%rbp)
        movl    -4(%rbp), %eax
        movl    %eax, %esi
        movl    $.LC2, %edi
        movl    $0, %eax
        call    printf
        movl    $0, %eax
        leave
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
.LFE0:
        .size   main, .-main
        .ident  "GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.2)
5.4.0 20160609"
        .section        .note.GNU-stack,"",@progbits
```

psum.s listing : gcc -S -c main.c

```
.file   "psum.c"
.text
.globl  psum
.type   psum, @function
psum:
.LFB0:
        .cfi_startproc
        pushq  %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq   %rsp, %rbp
        .cfi_def_cfa_register 6
        movl   %edi, -20(%rbp)
        movl   $0, -4(%rbp)
        movl   $1, -8(%rbp)
        jmp    .L2
.L3:
        movl   -8(%rbp), %eax
        addl   %eax, -4(%rbp)
        addl   $1, -8(%rbp)
.L2:
        movl   -8(%rbp), %eax
        cmpl   -20(%rbp), %eax
        jle    .L3
        movl   -4(%rbp), %eax
        popq   %rbp
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
.LFE0:
        .size   psum, .-psum
        .ident  "GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.2)
5.4.0 20160609"
        .section        .note.GNU-stack,"",@progbits
```

listing : gcc -c -g -Wa,-a,-ad main.c (1)

```
GAS LISTING /tmp/ccFifv3g.s                             page 1


 1                      .file   "main.c"
 2                      .text
 3              .Ltext0:
 4                      .section        .rodata
 5              .LC0:
 6 0000 53313D25                .string "S1=%d \n"
 6      64200A00
 7              .LC1:
 8 0008 53323D25                .string "S2=%d \n"
 8      64200A00
 9              .LC2:
10 0010 53333D25                .string "S3=%d \n"
10      64200A00
11                      .text
12                      .globl  main
14              main:
15              .LFB0:
16                      .file 1 "main.c"
 1:main.c       **** #include <stdio.h>
 2:main.c       **** #include "main.h"
 3:main.c       ****
 4:main.c       ****
 5:main.c       **** int main (void) {
11:main.c       ****   printf("S2=%d \n", S2);
17                      .loc 1 5 0
18                      .cfi_startproc
19 0000 55                      pushq   %rbp
20                      .cfi_def_cfa_offset 16
21                      .cfi_offset 6, -16
22 0001 4889E5                  movq    %rsp, %rbp
23                      .cfi_def_cfa_register 6
24 0004 4883EC10                subq    $16, %rsp
 6:main.c       ****   int S1, S2, S3;
 7:main.c       ****
 8:main.c       ****   S1 = psum(1);
25                      .loc 1 8 0
26 0008 BF010000                movl    $1, %edi
26      00
27 000d E8000000                call    psum
27      00
28 0012 8945F4                  movl    %eax, -12(%rbp)
 9:main.c       ****   printf("S1=%d \n", S1);
29                      .loc 1 9 0
30 0015 8B45F4                  movl    -12(%rbp), %eax
31 0018 89C6                    movl    %eax, %esi
32 001a BF000000                movl    $.LC0, %edi
32      00
33 001f B8000000                movl    $0, %eax
33      00
34 0024 E8000000                call    printf
34      00
10:main.c       ****   S2 = psum(2);
35                      .loc 1 10 0
36 0029 BF020000                movl    $2, %edi
36      00
37 002e E8000000                call    psum
37      00
38 0033 8945F8                  movl    %eax, -8(%rbp)
```

listing : gcc -c -g -Wa,-a,-ad main.c (2)

```
GAS LISTING /tmp/ccFifv3g.s                     page 2


 11:main.c      ****  printf("S2=%d \n", S2);
 39                   .loc 1 11 0
 40 0036 8B45F8       movl   -8(%rbp), %eax
 41 0039 89C6         movl   %eax, %esi
 42 003b BF000000     movl   $.LC1, %edi
 42      00
 43 0040 B8000000     movl   $0, %eax
 43      00
 44 0045 E8000000     call   printf
 44      00
 12:main.c      ****  S3 = psum(3);
 45                   .loc 1 12 0
 46 004a BF030000     movl   $3, %edi
 46      00
 47 004f E8000000     call   psum
 47      00
 48 0054 8945FC       movl   %eax, -4(%rbp)
 13:main.c      ****  printf("S3=%d \n", S3);
 49                   .loc 1 13 0
 50 0057 8B45FC       movl   -4(%rbp), %eax
 51 005a 89C6         movl   %eax, %esi
 52 005c BF000000     movl   $.LC2, %edi
 52      00
 53 0061 B8000000     movl   $0, %eax
 53      00
 54 0066 E8000000     call   printf
 54      00
 14:main.c      ****
 15:main.c      ****  return 0;
 55                   .loc 1 15 0
 56 006b B8000000     movl   $0, %eax
 56      00
 16:main.c      **** }
 57                   .loc 1 16 0
 58 0070 C9           leave
 59                   .cfi_def_cfa 7, 8
 60 0071 C3           ret
 61                   .cfi_endproc
 62             .LFE0:
 64             .Letext0:

GAS LISTING /tmp/ccFifv3g.s                     page 3


DEFINED SYMBOLS
                     *ABS*:0000000000000000 main.c
    /tmp/ccFifv3g.s:14     .text:0000000000000000 main

UNDEFINED SYMBOLS
psum
printf
```

listing : gcc -c -g -Wa,-a,-ad psum.c (2)

```
GAS LISTING /tmp/cc69qP8b.s                     page 1


 1                      .file   "psum.c"
 2                      .text
 3              .Ltext0:
 4                      .globl  psum
 6              psum:
 7              .LFB0:
 8                      .file 1 "psum.c"
 1:psum.c       **** int psum(int n)
 2:psum.c       **** {
 9                      .loc 1 2 0
10                      .cfi_startproc
11 0000 55                      pushq   %rbp
12                      .cfi_def_cfa_offset 16
13                      .cfi_offset 6, -16
14 0001 4889E5                  movq    %rsp, %rbp
15                      .cfi_def_cfa_register 6
16 0004 897DEC                  movl    %edi, -20(%rbp)
 3:psum.c       ****   int k, S=0;
17                      .loc 1 3 0
18 0007 C745FC00                movl    $0, -4(%rbp)
18      000000
 4:psum.c       ****   for (k=1; k<=n; ++k) S += k;
19                      .loc 1 4 0
20 000e C745F801                movl    $1, -8(%rbp)
20      000000
21 0015 EB0A                    jmp     .L2
22              .L3:
23                      .loc 1 4 0 is_stmt 0 discriminator 3
24 0017 8B45F8                  movl    -8(%rbp), %eax
25 001a 0145FC                  addl    %eax, -4(%rbp)
26 001d 8345F801                addl    $1, -8(%rbp)
27              .L2:
28                      .loc 1 4 0 discriminator 1
29 0021 8B45F8                  movl    -8(%rbp), %eax
30 0024 3B45EC                  cmpl    -20(%rbp), %eax
31 0027 7EEE                    jle     .L3
 5:psum.c       ****   return S;
32                      .loc 1 5 0 is_stmt 1
33 0029 8B45FC                  movl    -4(%rbp), %eax
 6:psum.c       **** }
34                      .loc 1 6 0
35 002c 5D                      popq    %rbp
36                      .cfi_def_cfa 7, 8
37 002d C3                      ret
38                      .cfi_endproc
39              .LFE0:
41              .Letext0:

GAS LISTING /tmp/cc69qP8b.s                     page 2


DEFINED SYMBOLS
                    *ABS*:0000000000000000 psum.c
    /tmp/cc69qP8b.s:6      .text:0000000000000000 psum

NO UNDEFINED SYMBOLS
```

listing : objdump -d main.o

main.o: file format elf64-x86-64

Disassembly of section .text:

```
0000000000000000 <main>:
   0:   55                      push   %rbp
   1:   48 89 e5                mov    %rsp,%rbp
   4:   48 83 ec 10             sub    $0x10,%rsp
   8:   bf 01 00 00 00          mov    $0x1,%edi
   d:   e8 00 00 00 00          callq  12 <main+0x12>
  12:   89 45 f4                mov    %eax,-0xc(%rbp)
  15:   8b 45 f4                mov    -0xc(%rbp),%eax
  18:   89 c6                   mov    %eax,%esi
  1a:   bf 00 00 00 00          mov    $0x0,%edi
  1f:   b8 00 00 00 00          mov    $0x0,%eax
  24:   e8 00 00 00 00          callq  29 <main+0x29>
  29:   bf 02 00 00 00          mov    $0x2,%edi
  2e:   e8 00 00 00 00          callq  33 <main+0x33>
  33:   89 45 f8                mov    %eax,-0x8(%rbp)
  36:   8b 45 f8                mov    -0x8(%rbp),%eax
  39:   89 c6                   mov    %eax,%esi
  3b:   bf 00 00 00 00          mov    $0x0,%edi
  40:   b8 00 00 00 00          mov    $0x0,%eax
  45:   e8 00 00 00 00          callq  4a <main+0x4a>
  4a:   bf 03 00 00 00          mov    $0x3,%edi
  4f:   e8 00 00 00 00          callq  54 <main+0x54>
  54:   89 45 fc                mov    %eax,-0x4(%rbp)
  57:   8b 45 fc                mov    -0x4(%rbp),%eax
  5a:   89 c6                   mov    %eax,%esi
  5c:   bf 00 00 00 00          mov    $0x0,%edi
  61:   b8 00 00 00 00          mov    $0x0,%eax
  66:   e8 00 00 00 00          callq  6b <main+0x6b>
  6b:   b8 00 00 00 00          mov    $0x0,%eax
  70:   c9                      leaveq
  71:   c3                      retq
```

listing : objdump -d psum.o

```
psum.o:     file format elf64-x86-64


Disassembly of section .text:

0000000000000000 <psum>:
   0:   55                      push   %rbp
   1:   48 89 e5                mov    %rsp,%rbp
   4:   89 7d ec                mov    %edi,-0x14(%rbp)
   7:   c7 45 fc 00 00 00 00            movl   $0x0,-0x4(%rbp)
   e:   c7 45 f8 01 00 00 00            movl   $0x1,-0x8(%rbp)
  15:   eb 0a                   jmp    21 <psum+0x21>
  17:   8b 45 f8                mov    -0x8(%rbp),%eax
  1a:   01 45 fc                add    %eax,-0x4(%rbp)
  1d:   83 45 f8 01             addl   $0x1,-0x8(%rbp)
  21:   8b 45 f8                mov    -0x8(%rbp),%eax
  24:   3b 45 ec                cmp    -0x14(%rbp),%eax
  27:   7e ee                   jle    17 <psum+0x17>
  29:   8b 45 fc                mov    -0x4(%rbp),%eax
  2c:   5d                      pop    %rbp
  2d:   c3                      retq
```

listing : objdump -S -d main.o

```
main.o:     file format elf64-x86-64


Disassembly of section .text:

0000000000000000 <main>:
#include <stdio.h>
#include "main.h"


int main (void) {
   0:   55                      push   %rbp
   1:   48 89 e5                mov    %rsp,%rbp
   4:   48 83 ec 10             sub    $0x10,%rsp
 int S1, S2, S3;

  S1 = psum(1);
   8:   bf 01 00 00 00          mov    $0x1,%edi
   d:   e8 00 00 00 00          callq  12 <main+0x12>
  12:   89 45 f4                mov    %eax,-0xc(%rbp)
  printf("S1=%d \n", S1);
  15:   8b 45 f4                mov    -0xc(%rbp),%eax
  18:   89 c6                   mov    %eax,%esi
  1a:   bf 00 00 00 00          mov    $0x0,%edi
  1f:   b8 00 00 00 00          mov    $0x0,%eax
  24:   e8 00 00 00 00          callq  29 <main+0x29>
  S2 = psum(2);
  29:   bf 02 00 00 00          mov    $0x2,%edi
  2e:   e8 00 00 00 00          callq  33 <main+0x33>
  33:   89 45 f8                mov    %eax,-0x8(%rbp)
  printf("S2=%d \n", S2);
  36:   8b 45 f8                mov    -0x8(%rbp),%eax
  39:   89 c6                   mov    %eax,%esi
  3b:   bf 00 00 00 00          mov    $0x0,%edi
  40:   b8 00 00 00 00          mov    $0x0,%eax
  45:   e8 00 00 00 00          callq  4a <main+0x4a>
  S3 = psum(3);
  4a:   bf 03 00 00 00          mov    $0x3,%edi
  4f:   e8 00 00 00 00          callq  54 <main+0x54>
  54:   89 45 fc                mov    %eax,-0x4(%rbp)
  printf("S3=%d \n", S3);
  57:   8b 45 fc                mov    -0x4(%rbp),%eax
  5a:   89 c6                   mov    %eax,%esi
  5c:   bf 00 00 00 00          mov    $0x0,%edi
  61:   b8 00 00 00 00          mov    $0x0,%eax
  66:   e8 00 00 00 00          callq  6b <main+0x6b>

  return 0;
  6b:   b8 00 00 00 00          mov    $0x0,%eax
}
  70:   c9                      leaveq
  71:   c3                      retq
```

listing : objdump -S -d psum.o

```
psum.o:     file format elf64-x86-64


Disassembly of section .text:

0000000000000000 <psum>:
int psum(int n)
{
   0:   55                      push   %rbp
   1:   48 89 e5                mov    %rsp,%rbp
   4:   89 7d ec                mov    %edi,-0x14(%rbp)
  int k, S=0;
   7:   c7 45 fc 00 00 00 00          movl   $0x0,-0x4(%rbp)
  for (k=1; k<=n; ++k) S += k;
   e:   c7 45 f8 01 00 00 00          movl   $0x1,-0x8(%rbp)
  15:   eb 0a                   jmp    21 <psum+0x21>
  17:   8b 45 f8                mov    -0x8(%rbp),%eax
  1a:   01 45 fc                add    %eax,-0x4(%rbp)
  1d:   83 45 f8 01             addl   $0x1,-0x8(%rbp)
  21:   8b 45 f8                mov    -0x8(%rbp),%eax
  24:   3b 45 ec                cmp    -0x14(%rbp),%eax
  27:   7e ee                   jle    17 <psum+0x17>
  return S;
  29:   8b 45 fc                mov    -0x4(%rbp),%eax
}
  2c:   5d                      pop    %rbp
  2d:   c3                      retq
```

collect2

Static Library

- mkdir libttt
- cd libttt
- cp /usr/lib/libc.a ./
- ar -x libttt.a

Dynamic Library

-

ELF

gcc -g t t.c

objdump -S t

gcc -v –save-temps -o t t.c

- t.i (the preprocessed output)
- t.s (the assembly file)

Collect2, ld Options

-L
-l
-shared
-static
-nostdlib
-nostartfiles
-Wl
-s
-x
-n
-r
-e
-M
-oformat

ELF

gcc -da t.c

as -V -Qy -o t.o t.s

References

[1] An Introduction to GCC, B. Gough, http://www.network-theory.co.uk/docs/gccintro/
[2] Unix, Linux Programming Indispensable Utilities, CW Paik