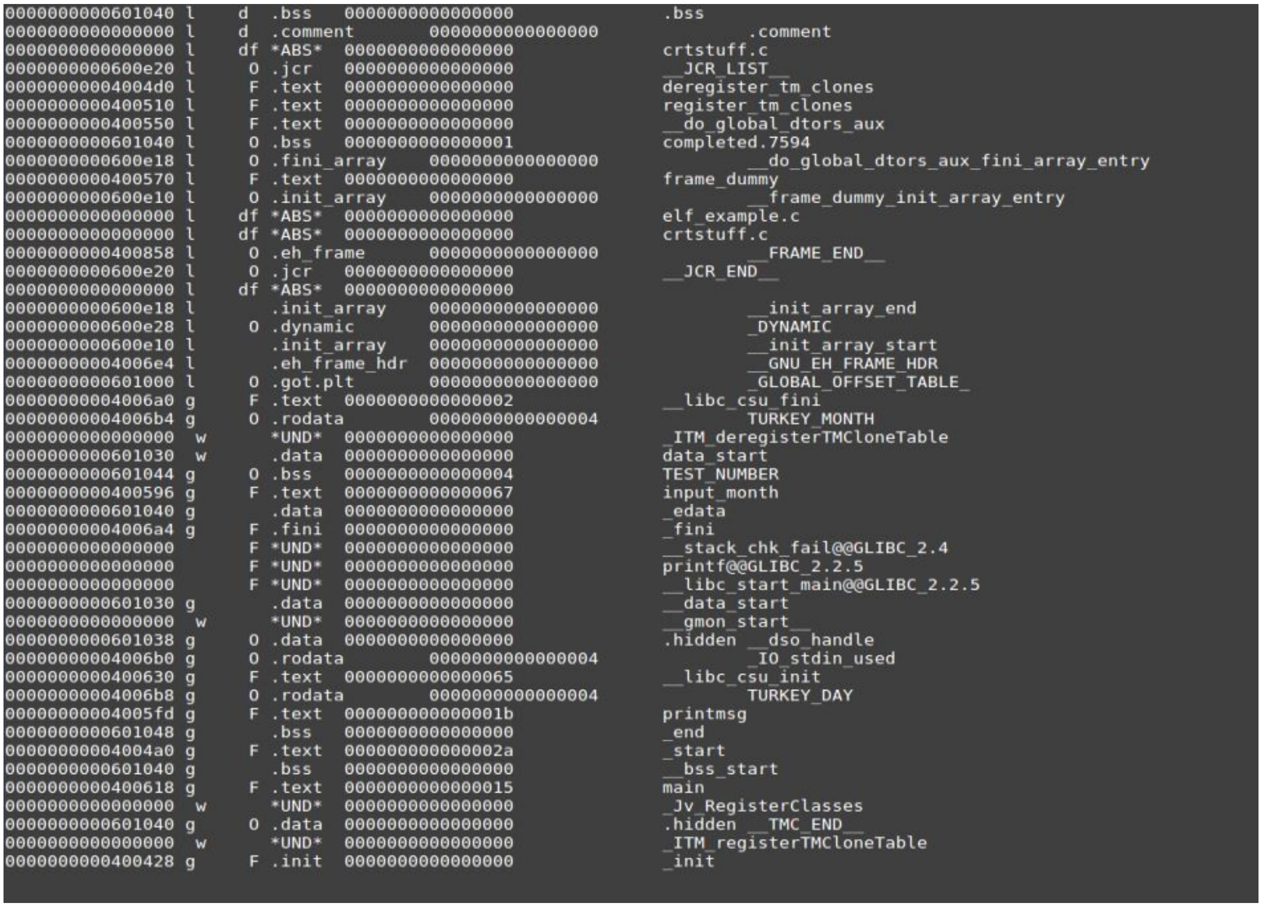
Hunter Damron
Graham McDonald
Reed Segars

ELF Analysis

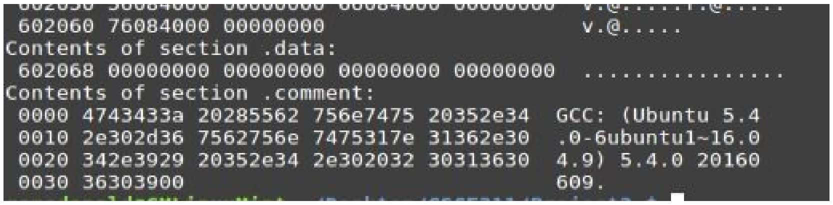
The Executable and Linkable Format (ELF) is the generic file format for executables in Linux based systems. It defines the structure for binaries, libraries, and core files. The three major components include the ELF header, sections, and segments. In our analysis, we created a simple program (elf_example.cc) with the binary executable ./elf. This program simply prompts the user for the correct month (11) and day (22) of Thanksgiving and does some error checking. The code was broken up into several functions and made use of constants in order to better illustrate the ELF file's sections and segments.

Sections contain the information needed for linking an object file to form an executable.

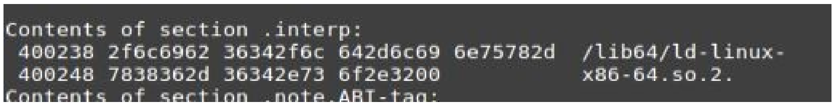
TEST_NUMBER from the elf_example.c program is stored in the .bss section. The variables input_month, TURKEY_MONTH, and TURKEY_DAY from the elf_example.cc program are stored in the .text section. The strings inside of the elf_example.cc program are stored in the .rodata section. The compiler and operating system information are stored in the .comment section. The .data section is empty. During runtime, global variables will be retrieved from the .text section into .data section. The .interp section stores the dynamic linker file path.



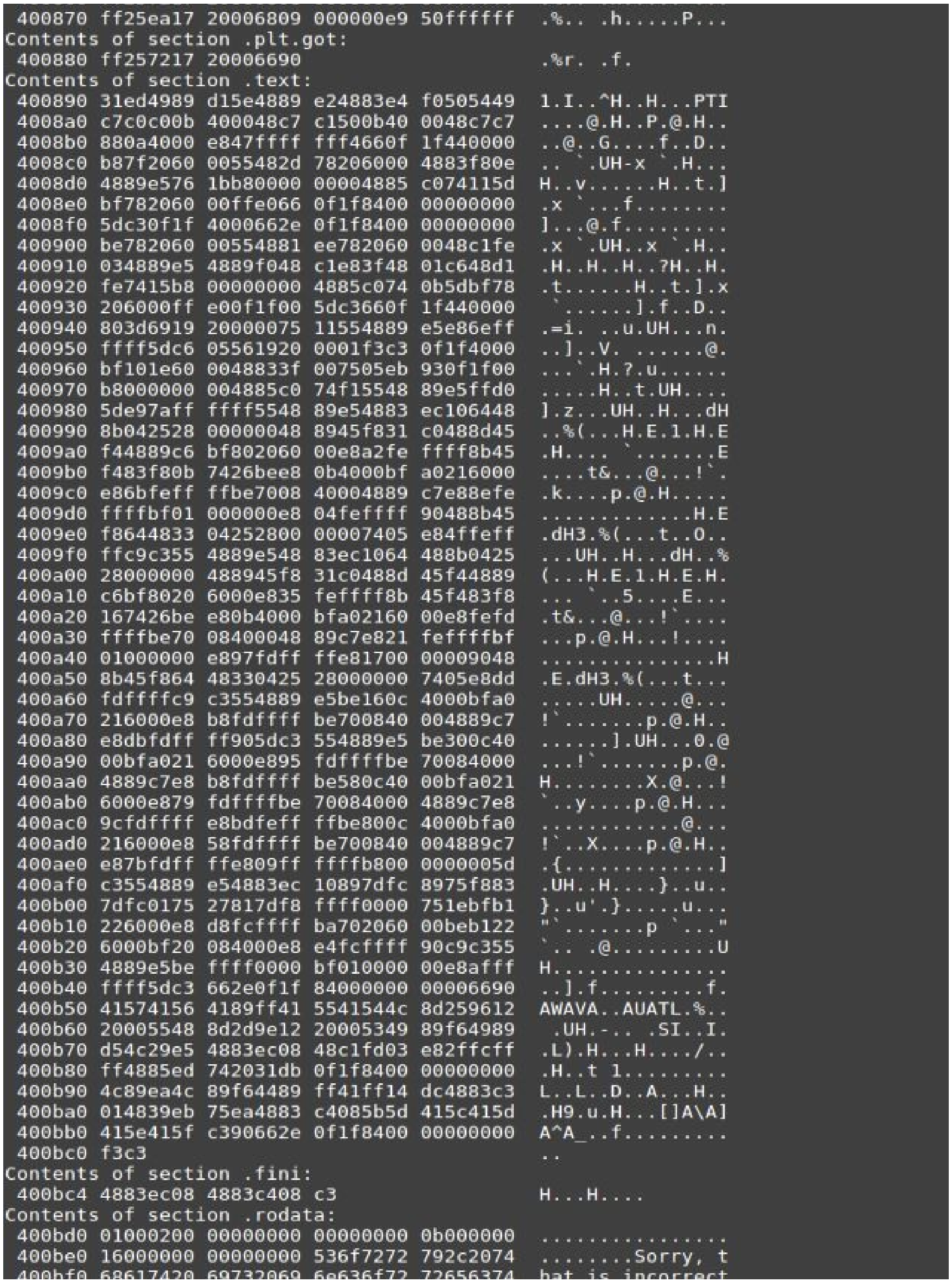
```
0000000000601040 l    d  .bss   0000000000000000              .bss
0000000000000000 l    d  .comment       0000000000000000              .comment
0000000000000000 l    df *ABS*  0000000000000000              crtstuff.c
0000000000600e20 l     O .jcr   0000000000000000              __JCR_LIST__
00000000004004d0 l     F .text  0000000000000000              deregister_tm_clones
0000000000400510 l     F .text  0000000000000000              register_tm_clones
0000000000400550 l     F .text  0000000000000000              __do_global_dtors_aux
0000000000601040 l     O .bss   0000000000000001              completed.7594
0000000000600e18 l     O .fini_array    0000000000000000              __do_global_dtors_aux_fini_array_entry
0000000000400570 l     F .text  0000000000000000              frame_dummy
0000000000600e10 l     O .init_array    0000000000000000              __frame_dummy_init_array_entry
0000000000000000 l    df *ABS*  0000000000000000              elf_example.c
0000000000000000 l    df *ABS*  0000000000000000              crtstuff.c
0000000000400858 l     O .eh_frame      0000000000000000              __FRAME_END__
0000000000600e20 l     O .jcr   0000000000000000              __JCR_END__
0000000000000000 l    df *ABS*  0000000000000000
0000000000600e18 l       .init_array    0000000000000000              __init_array_end
0000000000600e28 l     O .dynamic       0000000000000000              _DYNAMIC
0000000000600e10 l       .init_array    0000000000000000              __init_array_start
00000000004006e4 l       .eh_frame_hdr  0000000000000000              __GNU_EH_FRAME_HDR
0000000000601000 l     O .got.plt       0000000000000000              _GLOBAL_OFFSET_TABLE_
00000000004006a0 g     F .text  0000000000000002              __libc_csu_fini
00000000004006b4 g     O .rodata        0000000000000004              TURKEY_MONTH
0000000000000000  w      *UND*  0000000000000000              _ITM_deregisterTMCloneTable
0000000000601030  w      .data  0000000000000000              data_start
0000000000601044 g     O .bss   0000000000000004              TEST_NUMBER
0000000000400596 g     F .text  0000000000000067              input_month
0000000000601040 g       .data  0000000000000000              _edata
00000000004006a4 g     F .fini  0000000000000000              _fini
0000000000000000       F *UND*  0000000000000000              __stack_chk_fail@@GLIBC_2.4
0000000000000000       F *UND*  0000000000000000              printf@@GLIBC_2.2.5
0000000000000000       F *UND*  0000000000000000              __libc_start_main@@GLIBC_2.2.5
0000000000601030 g       .data  0000000000000000              __data_start
0000000000000000  w      *UND*  0000000000000000              __gmon_start__
0000000000601038 g     O .data  0000000000000000              .hidden __dso_handle
00000000004006b0 g     O .rodata        0000000000000004              _IO_stdin_used
0000000000400630 g     F .text  0000000000000065              __libc_csu_init
00000000004006b8 g     O .rodata        0000000000000004              TURKEY_DAY
00000000004005fd g     F .text  000000000000001b              printmsg
0000000000601048 g       .bss   0000000000000000              _end
00000000004004a0 g     F .text  000000000000002a              _start
0000000000601040 g       .bss   0000000000000000              __bss_start
0000000000400618 g     F .text  0000000000000015              main
0000000000000000  w      *UND*  0000000000000000              _Jv_RegisterClasses
0000000000601040 g     O .data  0000000000000000              .hidden __TMC_END__
0000000000000000  w      *UND*  0000000000000000              _ITM_registerTMCloneTable
0000000000400428 g     F .init  0000000000000000              _init
```



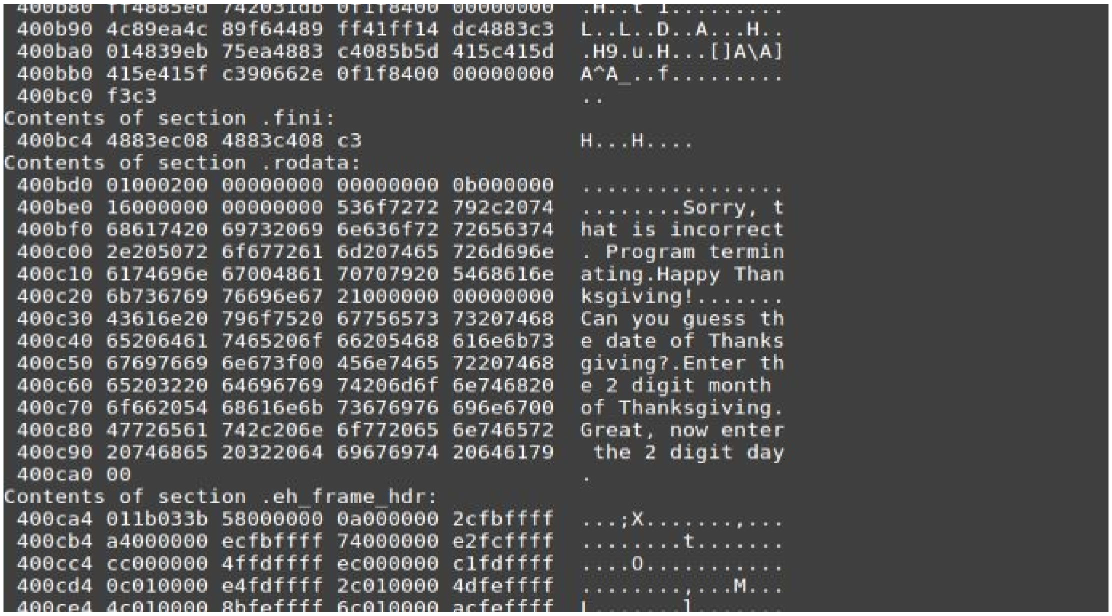
```
 602060 76084000 00000000                    v.@.....
Contents of section .data:
 602068 00000000 00000000 00000000 00000000  ................
Contents of section .comment:
 0000 4743433a 20285562 756e7475 20352e34  GCC: (Ubuntu 5.4
 0010 2e302d36 7562756e 7475317e 31362e30  .0-6ubuntu1~16.0
 0020 342e3929 20352e34 2e302032 30313630  4.9) 5.4.0 20160
 0030 36303900                             609.
```



```
Contents of section .interp:
 400238 2f6c6962 36342f6c 642d6c69 6e75782d  /lib64/ld-linux-
 400248 7838362d 36342e73 6f2e3200           x86-64.so.2.
Contents of section .note.ABI-tag:
```

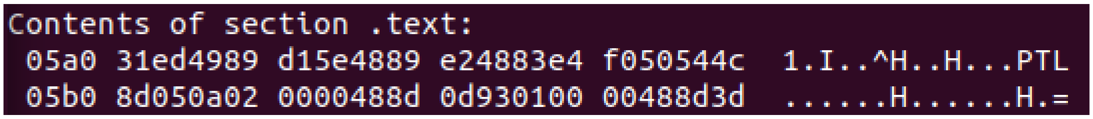


```
 400870 ff25ea17 20006809 000000e9 50ffffff  .%.. .h.....P...
Contents of section .plt.got:
 400880 ff257217 20006690                    .%r. .f.
Contents of section .text:
 400890 31ed4989 d15e4889 e24883e4 f0505449  1.I..^H..H...PTI
 4008a0 c7c0c00b 400048c7 c1500b40 0048c7c7  ....@.H..P.@.H..
 4008b0 880a4000 e847ffff fff4660f 1f440000  ..@..G....f..D..
 4008c0 b87f2060 0055482d 78206000 4883f80e  .. `.UH-x `.H...
 4008d0 4889e576 1bb80000 00004885 c074115d  H..v......H..t.]
 4008e0 bf782060 00ffe066 0f1f8400 00000000  .x `...f........
 4008f0 5dc30f1f 4000662e 0f1f8400 00000000  ]...@.f.........
 400900 be782060 00554881 ee782060 0048c1fe  .x `.UH..x `.H..
 400910 034889e5 4889f048 c1e83f48 01c648d1  .H..H..H..?H..H.
 400920 fe7415b8 00000000 4885c074 0b5dbf78  .t......H..t.].x
 400930 206000ff e00f1f00 5dc3660f 1f440000   `......].f..D..
 400940 803d6919 20000075 11554889 e5e86eff  .=i. ..u.UH...n.
 400950 ffff5dc6 05561920 0001f3c3 0f1f4000  ..]..V. ......@.
 400960 bf101e60 0048833f 007505eb 930f1f00  ...`.H.?.u......
 400970 b8000000 004885c0 74f15548 89e5ffd0  .....H..t.UH....
 400980 5de97aff ffff5548 89e54883 ec106448  ].z...UH..H...dH
 400990 8b042528 00000048 8945f831 c0488d45  ..%(...H.E.1.H.E
 4009a0 f44889c6 bf802060 00e8a2fe ffff8b45  .H.... `.......E
 4009b0 f483f80b 7426bee8 0b4000bf a0216000  ....t&...@...!`.
 4009c0 e86bfeff ffbe7008 40004889 c7e88efe  .k....p.@.H.....
 4009d0 ffffbf01 000000e8 04feffff 90488b45  .............H.E
 4009e0 f8644833 04252800 00007405 e84ffeff  .dH3.%(...t..O..
 4009f0 ffc9c355 4889e548 83ec1064 488b0425  ...UH..H...dH..%
 400a00 28000000 488945f8 31c0488d 45f44889  (...H.E.1.H.E.H.
 400a10 c6bf8020 6000e835 feffff8b 45f483f8  ... `..5....E...
 400a20 167426be e80b4000 bfa02160 00e8fefd  .t&...@...!`....
 400a30 ffffbe70 08400048 89c7e821 feffffbf  ...p.@.H...!....
 400a40 01000000 e897fdff ffe81700 00009048  ...............H
 400a50 8b45f864 48330425 28000000 7405e8dd  .E.dH3.%(...t...
 400a60 fdffffc9 c3554889 e5be160c 4000bfa0  .....UH.....@...
 400a70 216000e8 b8fdffff be700840 004889c7  !`.......p.@.H..
 400a80 e8dbfdff ff905dc3 554889e5 be300c40  ......].UH...0.@
 400a90 00bfa021 6000e895 fdffffbe 70084000  ...!`.......p.@.
 400aa0 4889c7e8 b8fdffff be580c40 00bfa021  H........X.@...!
 400ab0 6000e879 fdffffbe 70084000 4889c7e8  `..y....p.@.H...
 400ac0 9cfdffff e8bdfeff ffbe800c 4000bfa0  ............@...
 400ad0 216000e8 58fdffff be700840 004889c7  !`..X....p.@.H..
 400ae0 e87bfdff ffe809ff ffffb800 0000005d  .{.............]
 400af0 c3554889 e54883ec 10897dfc 8975f883  .UH..H....}..u..
 400b00 7dfc0175 27817df8 ffff0000 751ebfb1  }..u'.}.....u...
 400b10 226000e8 d8fcffff ba702060 00beb122  "`.......p `..."
 400b20 6000bf20 084000e8 e4fcffff 90c9c355  `.. .@.........U
 400b30 4889e5be ffff0000 bf010000 00e8afff  H...............
 400b40 ffff5dc3 662e0f1f 84000000 00006690  ..].f.........f.
 400b50 41574156 4189ff41 5541544c 8d259612  AWAVA..AUATL.%..
 400b60 20005548 8d2d9e12 20005349 89f64989   .UH.-.. .SI..I.
 400b70 d54c29e5 4883ec08 48c1fd03 e82ffcff  .L).H...H..../..
 400b80 ff4885ed 742031db 0f1f8400 00000000  .H..t 1.........
 400b90 4c89ea4c 89f64489 ff41ff14 dc4883c3  L..L..D..A...H..
 400ba0 014839eb 75ea4883 c4085b5d 415c415d  .H9.u.H...[]A\A]
 400bb0 415e415f c390662e 0f1f8400 00000000  A^A_..f.........
 400bc0 f3c3                                 ..
Contents of section .fini:
 400bc4 4883ec08 4883c408 c3                 H...H....
Contents of section .rodata:
 400bd0 01000200 00000000 00000000 0b000000  ................
 400be0 16000000 00000000 536f7272 792c2074  ........Sorry, t
 400bf0 68617420 69732069 6e636f72 72656374  hat is incorrect
```

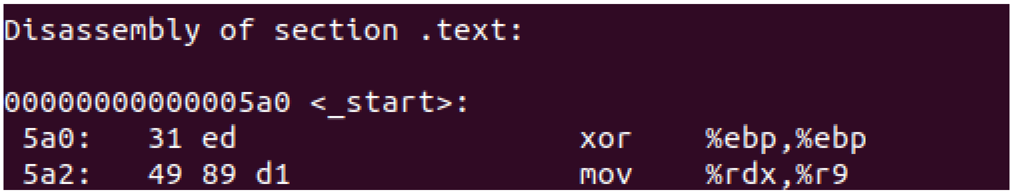


```
 400b80 ff4885ed 742031db 0f1f8400 00000000  .H..t 1.........
 400b90 4c89ea4c 89f64489 ff41ff14 dc4883c3  L..L..D..A...H..
 400ba0 014839eb 75ea4883 c4085b5d 415c415d  .H9.u.H...[]A\A]
 400bb0 415e415f c390662e 0f1f8400 00000000  A^A_..f.........
 400bc0 f3c3                                 ..
Contents of section .fini:
 400bc4 4883ec08 4883c408 c3                 H...H....
Contents of section .rodata:
 400bd0 01000200 00000000 00000000 0b000000  ................
 400be0 16000000 00000000 536f7272 792c2074  ........Sorry, t
 400bf0 68617420 69732069 6e636f72 72656374  hat is incorrect
 400c00 2e205072 6f677261 6d207465 726d696e  . Program termin
 400c10 6174696e 67004861 70707920 5468616e  ating.Happy Than
 400c20 6b736769 76696e67 21000000 00000000  ksgiving!.......
 400c30 43616e20 796f7520 67756573 73207468  Can you guess th
 400c40 65206461 7465206f 66205468 616e6b73  e date of Thanks
 400c50 67697669 6e673f00 456e7465 72207468  giving?.Enter th
 400c60 65203220 64696769 74206d6f 6e746820  e 2 digit month 
 400c70 6f662054 68616e6b 73676976 696e6700  of Thanksgiving.
 400c80 47726561 742c206e 6f772065 6e746572  Great, now enter
 400c90 20746865 20322064 69676974 20646179   the 2 digit day
 400ca0 00                                   .
Contents of section .eh_frame_hdr:
 400ca4 011b033b 58000000 0a000000 2cfbffff  ...;X.......,...
 400cb4 a4000000 ecfbffff 74000000 e2fcffff  ........t.......
 400cc4 cc000000 4ffdffff ec000000 c1fdffff  ....O...........
 400cd4 0c010000 e4fdffff 2c010000 4dfeffff  ........,...M...
 400ce4 4c010000 8bfeffff 6c010000 acfeffff  L.......l.......
```

Segments break down the structure of an ELF executable into small chunks to prepare it to be loaded into memory. In terms of the memory address space, the common segments include code (text), data, stack, and heap. In our example, the .text section of the ELF file is stored in the code segment of the memory address space. This can be seen by examining the disassembly of the .text section which shows the code text being translated to assembly language instructions.

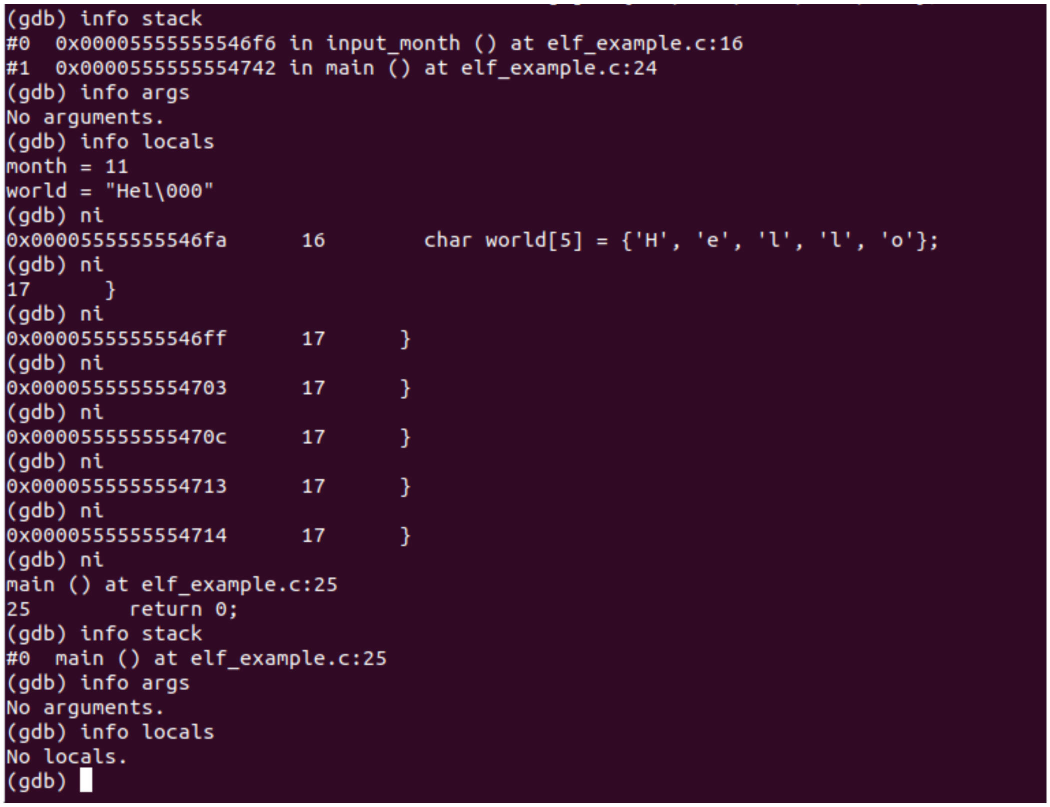


```
Contents of section .text:
 05a0 31ed4989 d15e4889 e24883e4 f050544c  1.I..^H..H...PTL
 05b0 8d050a02 0000488d 0d930100 00488d3d  ......H......H.=
```



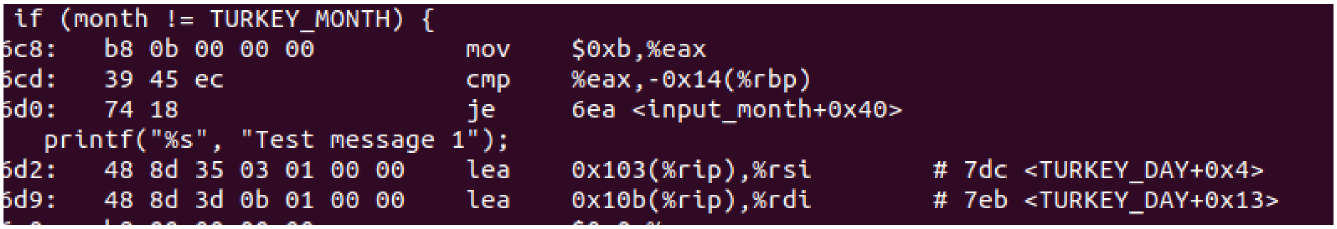
```
Disassembly of section .text:

00000000000005a0 <_start>:
 5a0:   31 ed                   xor    %ebp,%ebp
 5a2:   49 89 d1                mov    %rdx,%r9
```

Another segment is the call stack segment. The call stack stores function calls and local variables of those function calls for temporary use during program execution. Using gdb, we can see that the functions and local variables are pushed and popped off the stack during execution.



```
(gdb) info stack
#0  0x00005555555546f6 in input_month () at elf_example.c:16
#1  0x0000555555554742 in main () at elf_example.c:24
(gdb) info args
No arguments.
(gdb) info locals
month = 11
world = "Hel\000"
(gdb) ni
0x00005555555546fa      16        char world[5] = {'H', 'e', 'l', 'l', 'o'};
(gdb) ni
17      }
(gdb) ni
0x00005555555546ff      17      }
(gdb) ni
0x0000555555554703      17      }
(gdb) ni
0x000055555555470c      17      }
(gdb) ni
0x0000555555554713      17      }
(gdb) ni
0x0000555555554714      17      }
(gdb) ni
main () at elf_example.c:25
25        return 0;
(gdb) info stack
#0  main () at elf_example.c:25
(gdb) info args
No arguments.
(gdb) info locals
No locals.
(gdb) 
```

The last segment we examined was the data segment. The data segment contains the static (global) variables that exist throughout program execution. In our code, we had two initialized global variables, TURKEY_MONTH and TURKEY_DAY. In the object dump, we can see that this data is present through the assembly language. There is an lea operation which calculates the effective address of the constant as well as a compare operation which compares the value stored in %eax (0xb or 11) with that of TURKEY_MONTH which is stored in a stack frame register.



```
 if (month != TURKEY_MONTH) {
6c8:   b8 0b 00 00 00          mov    $0xb,%eax
6cd:   39 45 ec                cmp    %eax,-0x14(%rbp)
6d0:   74 18                   je     6ea <input_month+0x40>
   printf("%s", "Test message 1");
6d2:   48 8d 35 03 01 00 00    lea    0x103(%rip),%rsi        # 7dc <TURKEY_DAY+0x4>
6d9:   48 8d 3d 0b 01 00 00    lea    0x10b(%rip),%rdi        # 7eb <TURKEY_DAY+0x13>
```