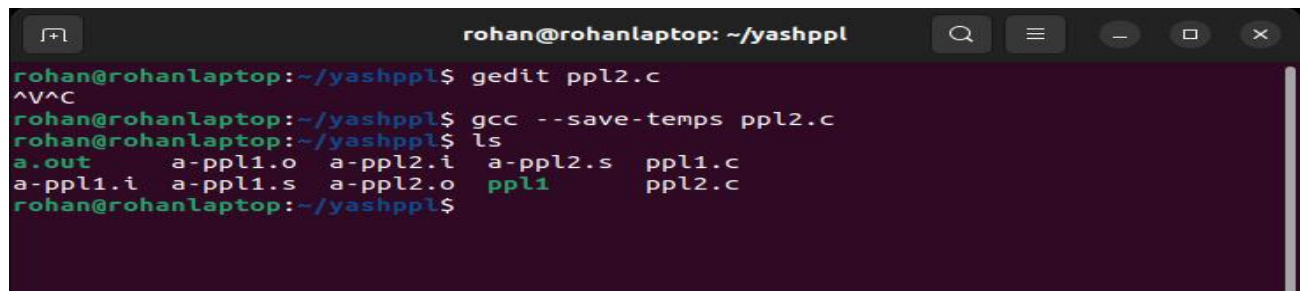


642303003

Assignment No:02

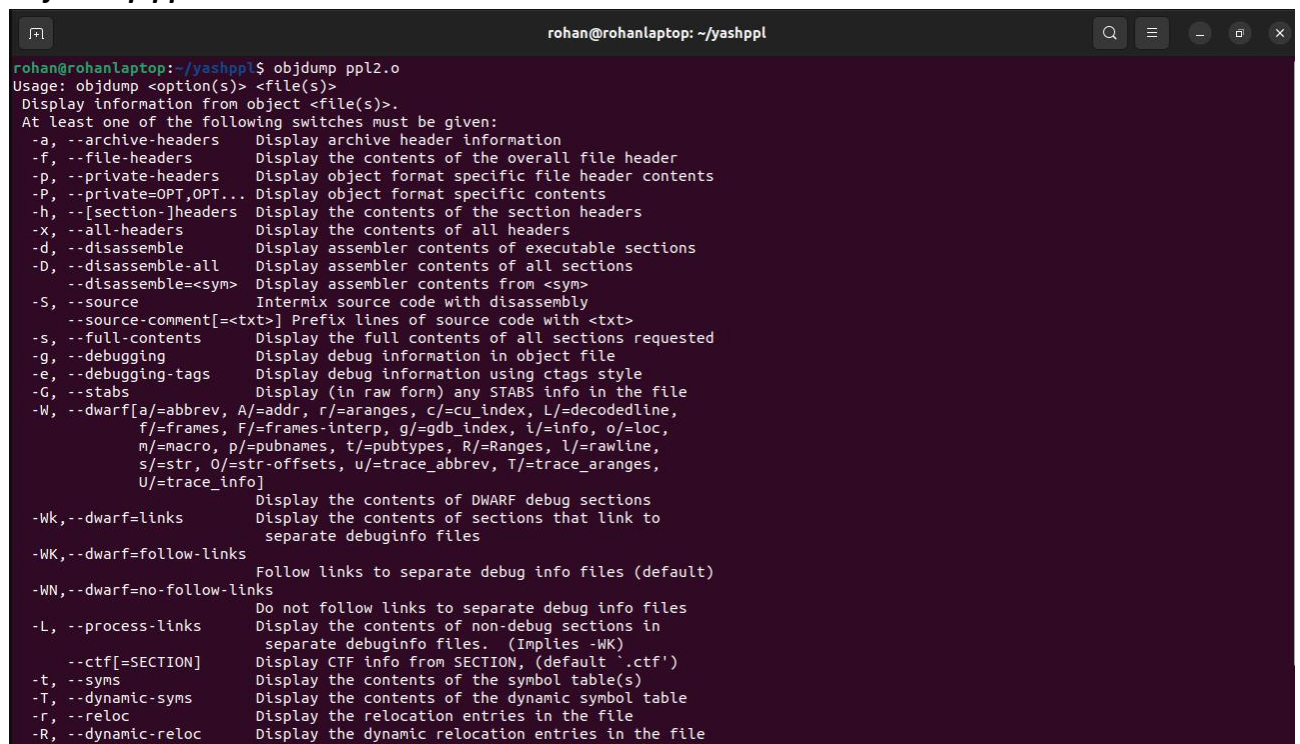
```
#include<stdio.h>
void function()
{ char buff[5];
    char buff1[16];
} int main(){ int a=1,
b=5,c=6; function();
return 0;
}
```



A terminal window titled 'rohan@rohanlaptop: ~/yashppl' showing the following commands and output:

```
rohan@rohanlaptop:~/yashppl$ gedit ppl2.c
^V^C
rohan@rohanlaptop:~/yashppl$ gcc --save-temps ppl2.c
rohan@rohanlaptop:~/yashppl$ ls
a.out      a-ppl1.o  a-ppl2.i  a-ppl2.s  ppl1.c
a-ppl1.i  a-ppl1.s  a-ppl2.o  ppl1      ppl2.c
rohan@rohanlaptop:~/yashppl$
```

Objedump ppl2.o



A terminal window titled 'rohan@rohanlaptop: ~/yashppl' showing the command 'objdump ppl2.o' and its help text:

```
rohan@rohanlaptop:~/yashppl$ objdump ppl2.o
Usage: objdump <option(s)> <file(s)>
Display information from object <file(s)>.
At least one of the following switches must be given:
-a, --archive-headers    Display archive header information
-f, --file-headers       Display the contents of the overall file header
-p, --private-headers    Display object format specific file header contents
-P, --private=OPT,OPT... Display object format specific contents
-h, --[section-]headers  Display the contents of the section headers
-x, --all-headers        Display the contents of all headers
-d, --disassemble        Display assembler contents of executable sections
-D, --disassemble-all   Display assembler contents of all sections
--disassemble=<sym>     Display assembler contents from <sym>
-S, --source             Intermix source code with disassembly
--source-comment[=<txt>] Prefix lines of source code with <txt>
-s, --full-contents      Display the full contents of all sections requested
-g, --debugging          Display debug information in object file
-e, --debugging-tags     Display debug information using ctags style
-G, --stabs              Display (in raw form) any STABS info in the file
-W, --dwarf[a=abbrev, A/=addr, r/=ranges, c/=cu_index, L/=decodedline,
f/=frames, F/=frames-interp, g/=gdb_index, i/=info, o/=loc,
n/=macro, p/=pubnames, t/=pubtypes, R/=Ranges, l/=rawline,
s/=str, O/=str-offsets, u/=trace_abbrev, T/=trace_ranges,
U/=trace_info]          Display the contents of DWARF debug sections
-Wk,--dwarf=links        Display the contents of sections that link to
                        separate debuginfo files
-WK,--dwarf=follow-links Follow links to separate debug info files (default)
-WN,--dwarf=no-follow-links Do not follow links to separate debug info files
-L, --process-links      Display the contents of non-debug sections in
                        separate debuginfo files. (Implies -WK)
--ctf[=SECTION]         Display CTF info from SECTION, (default '.ctf')
-t, --syms              Display the contents of the symbol table(s)
-T, --dynamic-syms      Display the contents of the dynamic symbol table
-r, --reloc             Display the relocation entries in the file
-R, --dynamic-reloc     Display the dynamic relocation entries in the file
```

objdump -f -a-ppl2.o:

642303003

```
rohan@rohanlaptop:~/yashppl$ objdump -f a-ppl2.o

a-ppl2.o:      file format elf64-x86-64
architecture: i386:x86-64, flags 0x00000011:
HAS_RELOC, HAS_SYMS
start address 0x0000000000000000

rohan@rohanlaptop:~/yashppl$
```

objdump -x a-ppl2.o:-

```
rohan@rohanlaptop:~/yashppl$ objdump -x a-ppl2.o

a-ppl2.o:      file format elf64-x86-64
a-ppl2.o
architecture: i386:x86-64, flags 0x00000011:
HAS_RELOC, HAS_SYMS
start address 0x0000000000000000

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
 0 .text          00000064  0000000000000000 0000000000000000 00000040 2**0
CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
 1 .data          00000000  0000000000000000 0000000000000000 000000a4 2**0
CONTENTS, ALLOC, LOAD, DATA
 2 .bss           00000000  0000000000000000 0000000000000000 000000a4 2**0
ALLOC
 3 .comment       0000002c  0000000000000000 0000000000000000 000000a4 2**0
CONTENTS, READONLY
 4 .note.GNU-stack 00000000  0000000000000000 0000000000000000 000000d0 2**0
CONTENTS, READONLY
 5 .note.gnu.property 00000020  0000000000000000 0000000000000000 000000d0 2**3
CONTENTS, ALLOC, LOAD, READONLY, DATA
 6 .eh_frame      00000058  0000000000000000 0000000000000000 000000f0 2**3
CONTENTS, ALLOC, LOAD, RELOC, READONLY, DATA

SYMBOL TABLE:
0000000000000000 l    df *ABS* 0000000000000000 ppl2.c
0000000000000000 l    d  .text 0000000000000000 .text
0000000000000000 g    F  .text 0000000000000032 func
0000000000000000 *UND* 0000000000000000 __stack_chk_fail
0000000000000032 g    F  .text 0000000000000032 main

RELOCATION RECORDS FOR [.text]:
OFFSET              TYPE          VALUE
000000000000002c R_X86_64_PLT32 __stack_chk_fail-0x0000000000000004
0000000000000059 R_X86_64_PLT32 func-0x0000000000000004
```

objdump --disassemble-all ppl2.o:

```
rohan@rohanlaptop:~/yashppl$ objdump --disassemble-all a-ppl2.o

a-ppl2.o:      file format elf64-x86-64

Disassembly of section .text:

0000000000000000 <func>:
 0:  f3 0f 1e fa          endbr64
 4:  55                   push    %rbp
 5:  48 89 e5             mov     %rsp,%rbp
 8:  48 83 ec 20          sub     $0x20,%rsp
 c:  64 48 8b 04 25 28 00 mov     %fs:0x28,%rax
13:  00 00
15:  48 89 45 f8          mov     %rax,-0x8(%rbp)
19:  31 c0                xor     %eax,%eax
1b:  90                   nop
1c:  48 8b 45 f8          mov     -0x8(%rbp),%rax
20:  64 48 2b 04 25 28 00 sub     %fs:0x28,%rax
27:  00 00
29:  74 05                je      30 <func+0x30>
2b:  e8 00 00 00 00      call   30 <func+0x30>
30:  c9                   leave
31:  c3                   ret

0000000000000032 <main>:
32:  f3 0f 1e fa          endbr64
36:  55                   push    %rbp
37:  48 89 e5             mov     %rsp,%rbp
3a:  48 83 ec 10          sub     $0x10,%rsp
3e:  c7 45 f4 05 00 00 00 movl    $0x5,-0xc(%rbp)
45:  c7 45 f8 07 00 00 00 movl    $0x7,-0x8(%rbp)
4c:  c7 45 fc 08 00 00 00 movl    $0x8,-0x4(%rbp)
53:  b8 00 00 00 00      mov     $0x0,%eax
58:  e8 00 00 00 00      call   5d <main+0x2b>
5d:  b8 00 00 00 00      mov     $0x0,%eax
62:  c9                   leave
63:  c3                   ret
```

642303003

```
rohan@rohanlaptop: ~/yashppl

Disassembly of section .comment:

0000000000000000 <.comment>:
0: 00 47 43          add    %al,0x43(%rdi)
3: 43 3a 20          rex.XB cmp    (%r8),%spl
6: 28 55 62          sub    %dl,0x62(%rbp)
9: 75 0e             jne    79 <main+0x47>
b: 74 75             je     82 <main+0x50>
d: 20 31            and    %dh,(%rcx)
f: 31 2e            xor    %ebp,(%rsi)
11: 34 2e            xor    $0x2e,%al
13: 30 2d 31 75 62 75 xor    %ch,0x75627531(%rip)    # 7562754a <main+0x75627518>
19: 6e              outsb  %ds:(%rsi),(%dx)
1a: 74 75             je     91 <main+0x5f>
1c: 31 7e 32          xor    %edi,0x32(%rsi)
1f: 32 2e            xor    (%rsi),%ch
21: 30 34 29          xor    %dh,(%rcx,%rbp,1)
24: 20 31            and    %dh,(%rcx)
26: 31 2e            xor    %ebp,(%rsi)
28: 34 2e            xor    $0x2e,%al
2a: 30 00            xor    %al,(%rax)

Disassembly of section .note.gnu.property:

0000000000000000 <.note.gnu.property>:
0: 04 00            add    $0x0,%al
2: 00 00            add    %al,(%rax)
4: 10 00            adc    %al,(%rax)
6: 00 00            add    %al,(%rax)
8: 05 00 00 00 47    add    $0x47000000,%eax
d: 4e 55            rex.WRX push %rbp
f: 00 02            add    %al,(%rdx)
11: 00 00            add    %al,(%rax)
13: c0 04 00 00       rolb   $0x0,(%rax,%rax,1)
17: 00 03            add    %al,(%rbx)
19: 00 00            add    %al,(%rax)
1b: 00 00            add    %al,(%rax)
1d: 00 00            add    %al,(%rax)
```

```
Disassembly of section .eh_frame:

0000000000000000 <.eh_frame>:
0: 14 00            adc    $0x0,%al
2: 00 00            add    %al,(%rax)
4: 00 00            add    %al,(%rax)
6: 00 00            add    %al,(%rax)
8: 01 7a 52          add    %edi,0x52(%rdx)
b: 00 01            add    %al,(%rcx)
d: 78 10            js     1f <.eh_frame+0x1f>
f: 01 1b            add    %ebx,(%rbx)
11: 0c 07            or     $0x7,%al
13: 08 90 01 00 00 1c or     %dl,0x1c000001(%rax)
19: 00 00            add    %al,(%rax)
1b: 00 1c 00          add    %bl,(%rax,%rax,1)
1e: 00 00            add    %al,(%rax)
20: 00 00            add    %al,(%rax)
22: 00 00            add    %al,(%rax)
24: 32 00            xor    (%rax),%al
26: 00 00            add    %al,(%rax)
28: 00 45 0e          add    %al,0xe(%rbp)
2b: 10 86 02 43 0d 06 adc    %al,0x60d4302(%rsi)
31: 69 0c 07 08 00 00 00 imul   $0x8,(%rdi,%rax,1),%ecx
38: 1c 00            sbb    $0x0,%al
3a: 00 00            add    %al,(%rax)
3c: 3c 00            cmp    $0x0,%al
3e: 00 00            add    %al,(%rax)
40: 00 00            add    %al,(%rax)
42: 00 00            add    %al,(%rax)
44: 32 00            xor    (%rax),%al
46: 00 00            add    %al,(%rax)
48: 00 45 0e          add    %al,0xe(%rbp)
4b: 10 86 02 43 0d 06 adc    %al,0x60d4302(%rsi)
51: 69 0c 07 08 00 00 00 imul   $0x8,(%rdi,%rax,1),%ecx

rohan@rohanlaptop:~/yashppl$
```