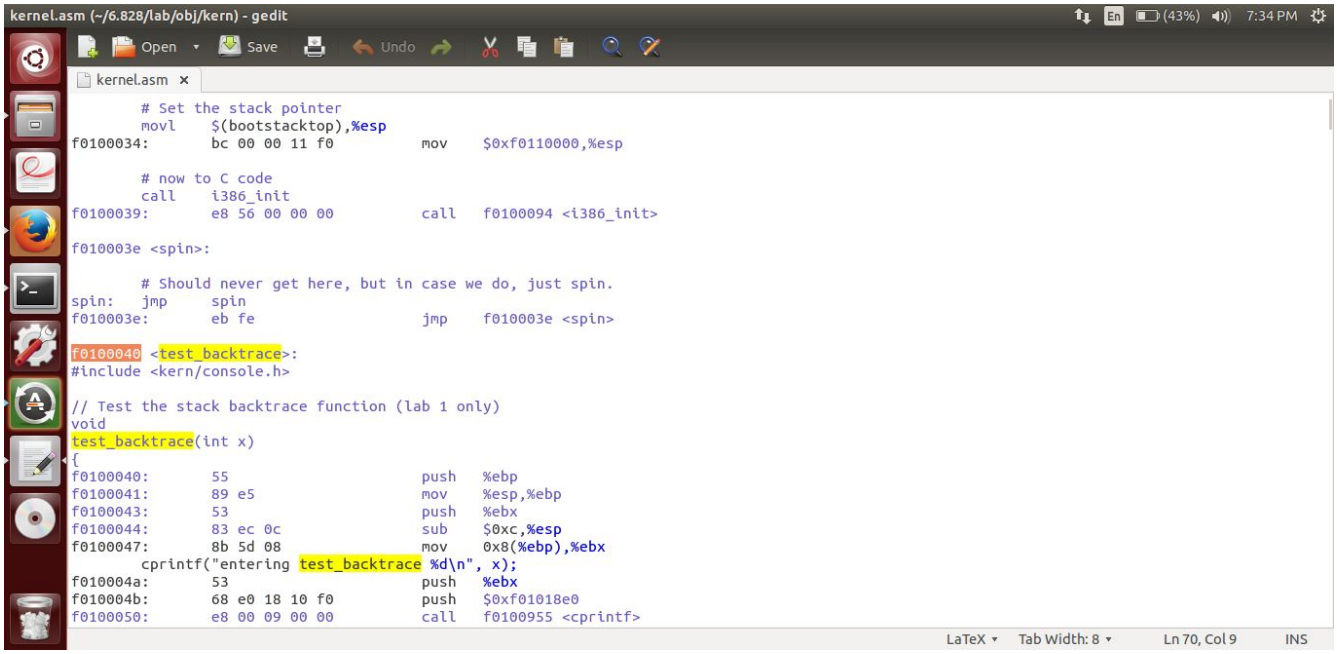
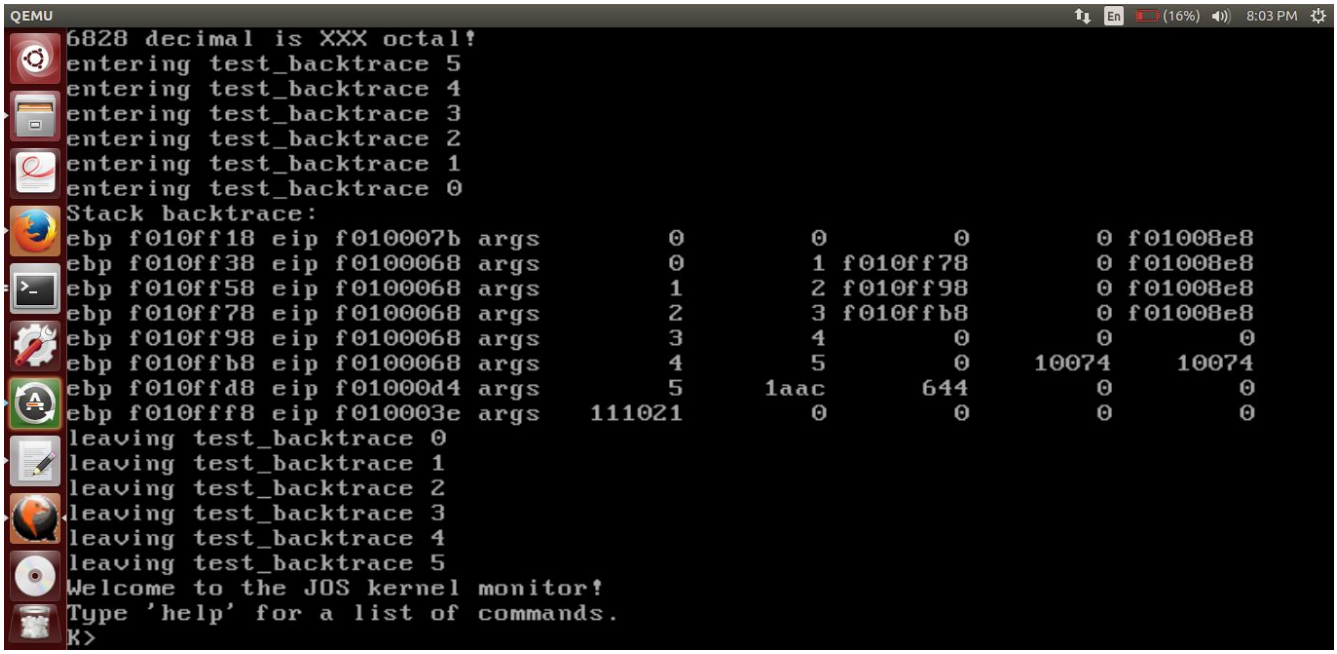
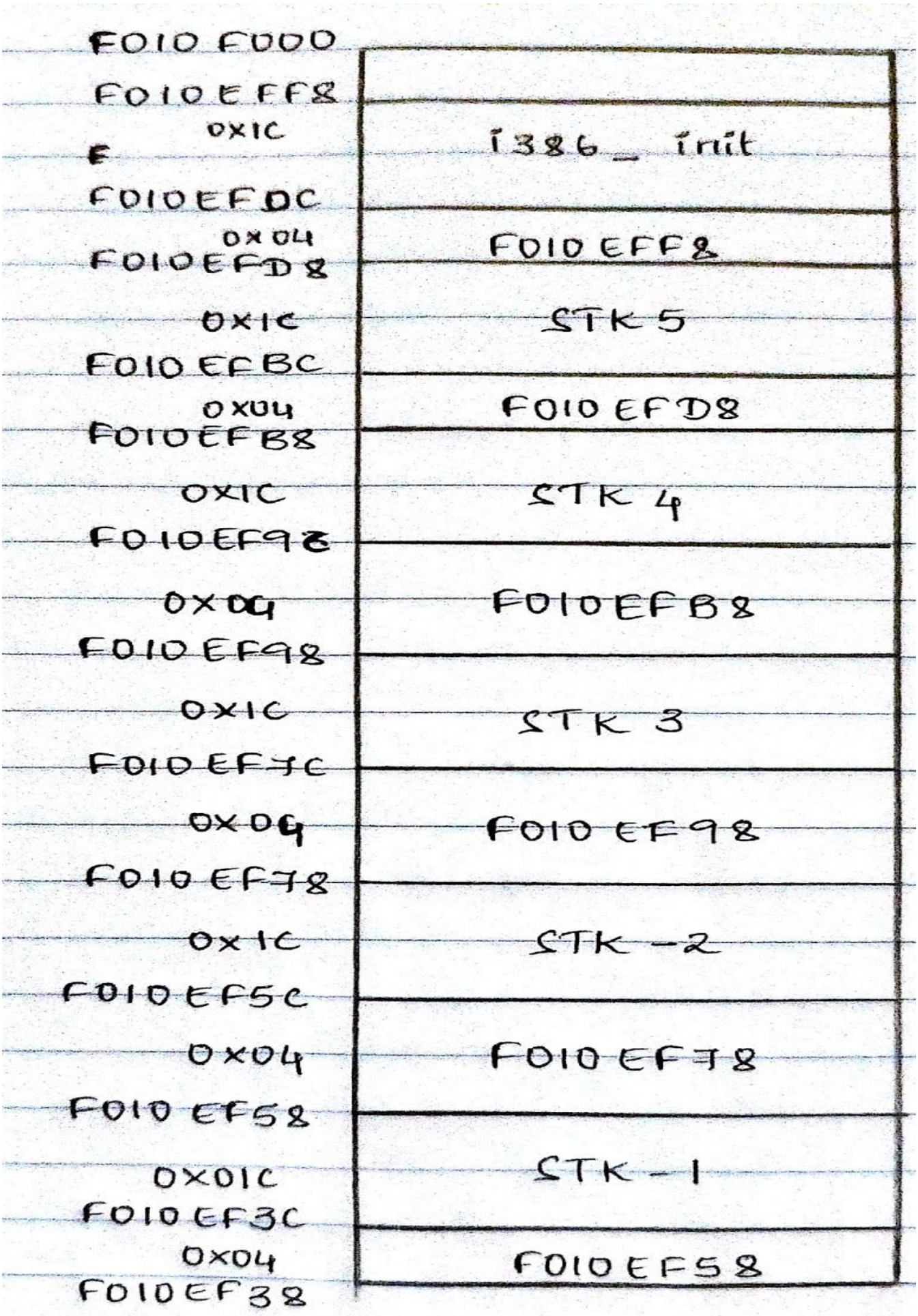
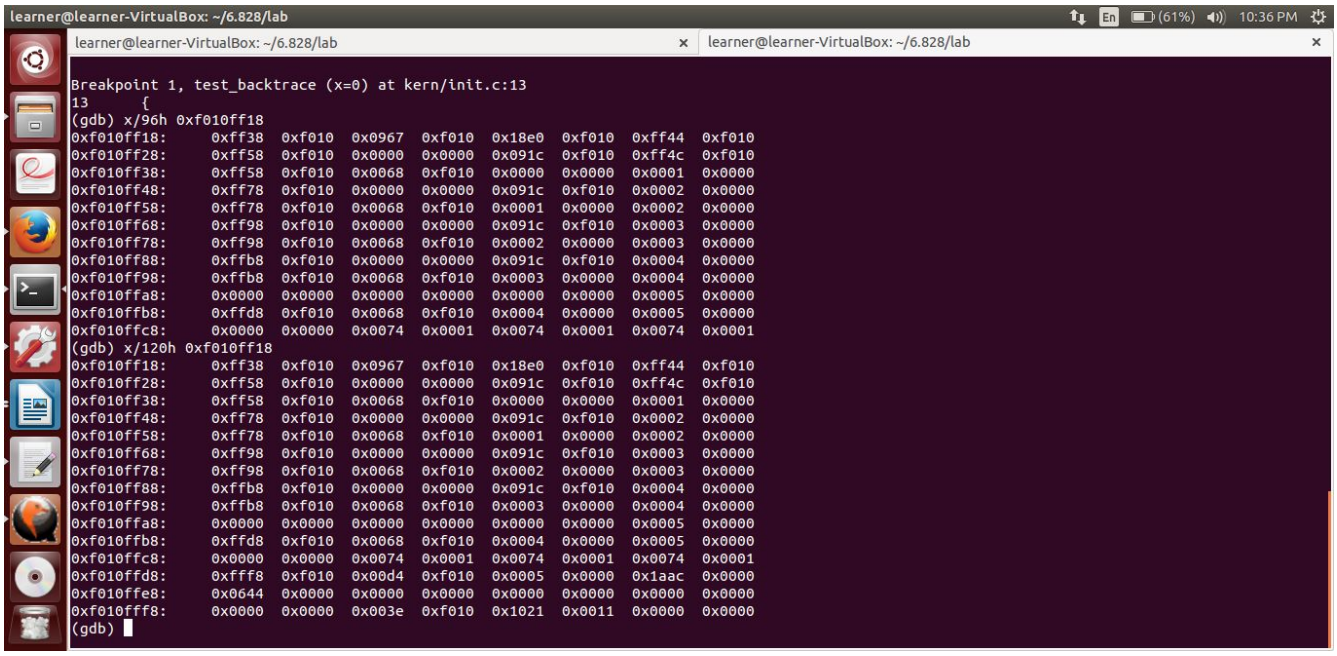


-Breakpoint at 0xf0100040
-Each time the function is called a new stack frame is initialized and the first two values of the initialised stack frame are the ebp and eip of the calling function.
-Size of each stack is 32 bytes.
-32 bits is 4 bytes-since each stack is 32 bytes therefore 32/4 is 8 words.
-ebp, eip, ebx of the current iteration,ebx of the previous iteration,ebp of the stack called before this.




F010F000
F010EFF8
0x1C
E
i386_init
F010EFDC
0x04
F010EFD8
F010EFF8
0x1C
STK 5
F010EFBC
0x04
F010EFB8
F010EFD8
0x1C
STK 4
F010EF9C
0x04
F010EF98
F010EFB8
0x1C
STK 3
F010EF7C
0x04
F010EF78
F010EF98
0x1C
STK -2
F010EF5C
0x04
F010EF58
F010EF78
0x01C
STK -1
F010EF3C
0x04
F010EF38
F010EF58



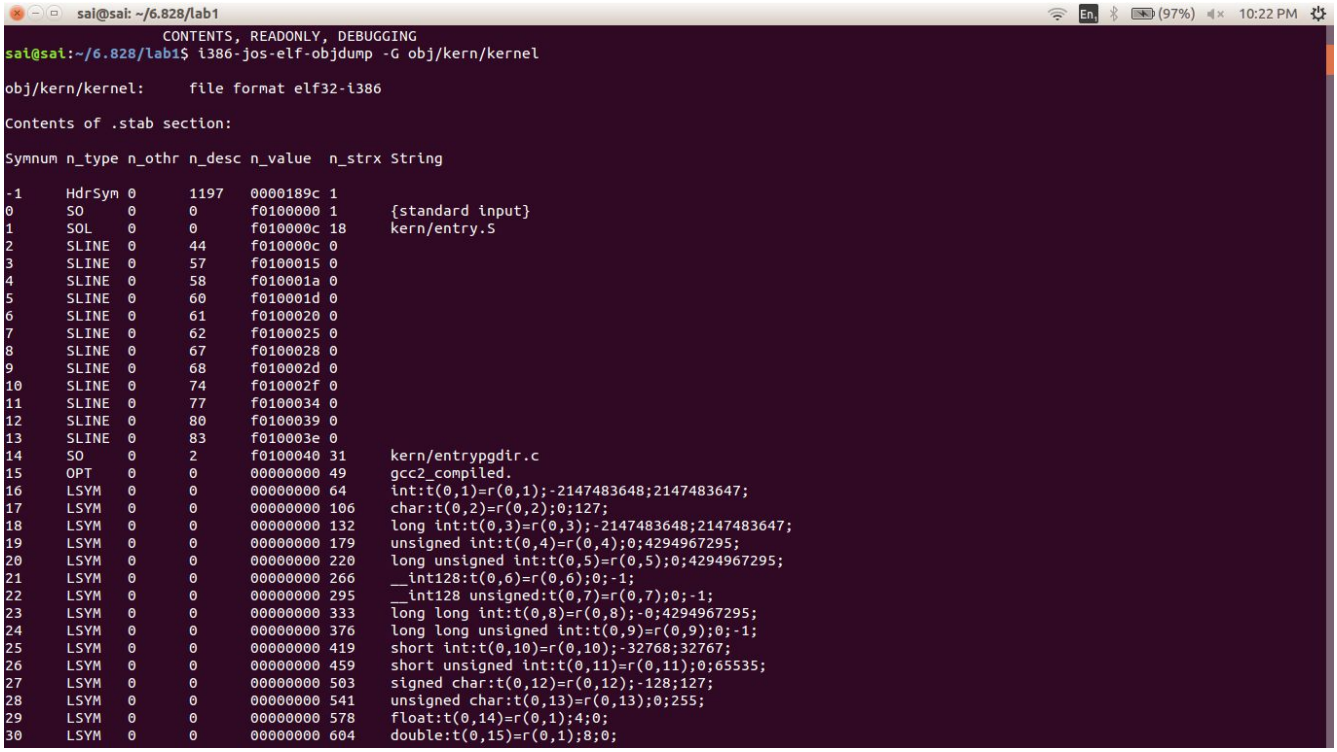
Exercise 11:

```
mon_backtrace(int argc, char **argv, struct Trapframe *tf)
{

uint32_t* ebp = (uint32_t*) read_ebp();       //This returns ebp address
    int i=2;                                  //To print the arguments

    cprintf("Stack backtrace:\n");
while(ebp)
{
cprintf("ebp %8x", ebp);                      //This gives the base pointer of the
stack
cprintf(" eip %8x", ebp[1]);                  //prints eip
cprintf(" args");
while(i<=6){                                  //prints rest of the arguments of the stack
cprintf(" %8x", *(ebp+i));
i++;
}
cprintf("\n");

ebp = (uint32_t*) *ebp;       //This pointer passes value of ebp present in the
current        stack to continue the while loop
i=2;
```

```
}
   return 0;
}
```

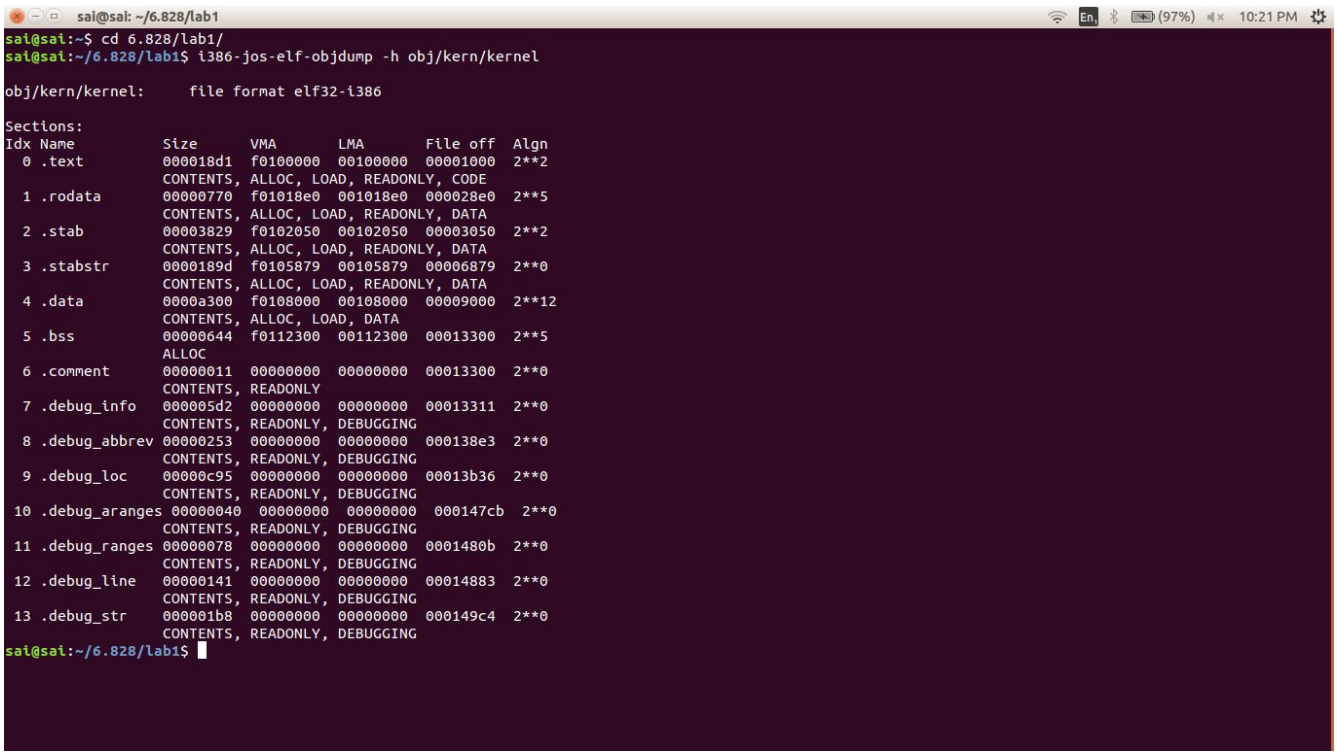
-

Exercise 12:



```
sai@sai: ~/6.828/lab1                                                    (98%)  10:28 PM
sai@sai: ~/6.828/lab1                                  sai@sai: ~/6.828/lab1
        add-auto-load-safe-path /home/sai/6.828/lab1/.gdbinit
line to your configuration file "/home/sai/.gdbinit".
To completely disable this security protection add
        set auto-load safe-path /
line to your configuration file "/home/sai/.gdbinit".
For more information about this security protection see the
"Auto-loading safe path" section in the GDB manual.  E.g., run from the shell:
        info "(gdb)Auto-loading safe path"
+ target remote localhost:26000
warning: A handler for the OS ABI "GNU/Linux" is not built into this configuration
of GDB.  Attempting to continue with the default i8086 settings.

The target architecture is assumed to be i8086
[f000:fff0]    0xffff0: ljmp   $0xf000,$0xe05b
0x0000fff0 in ?? ()
+ symbol-file obj/kern/kernel
(gdb) b *0xf0100040
Breakpoint 1 at 0xf0100040: file kern/init.c, line 13.
(gdb) c
Continuing.
The target architecture is assumed to be i386
=> 0xf0100040 <test_backtrace>: push   %ebp

Breakpoint 1, test_backtrace (x=5) at kern/init.c:13
13      {
(gdb) x/96h 0xf0102050
0xf0102050:     0x0001  0x0000  0x0000  0x04ad  0x189c  0x0000  0x0001  0x0000
0xf0102060:     0x0064  0x0000  0x0000  0xf010  0x0012  0x0000  0x0084  0x0000
0xf0102070:     0x000c  0xf010  0x0000  0x0000  0x0044  0x002c  0x000c  0xf010
0xf0102080:     0x0000  0x0000  0x0044  0x0039  0x0015  0xf010  0x0000  0x0000
0xf0102090:     0x0044  0x003a  0x001a  0xf010  0x0000  0x0000  0x0044  0x003c
0xf01020a0:     0x001d  0xf010  0x0000  0x0000  0x0044  0x003d  0x0020  0xf010
0xf01020b0:     0x0000  0x0000  0x0044  0x003e  0x0025  0xf010  0x0000  0x0000
0xf01020c0:     0x0044  0x0043  0x0028  0xf010  0x0000  0x0000  0x0044  0x0044
0xf01020d0:     0x002d  0xf010  0x0000  0x0000  0x0044  0x004a  0x002f  0xf010
0xf01020e0:     0x0000  0x0000  0x0044  0x004d  0x0034  0xf010  0x0000  0x0000
0xf01020f0:     0x0044  0x0050  0x0039  0xf010  0x0000  0x0000  0x0044  0x0053
0xf0102100:     0x003e  0xf010  0x001f  0x0000  0x0064  0x0002  0x0040  0xf010
(gdb)
```

```
sai@sai: ~/6.828/lab1                                                    (99%)  10:33 PM
sai@sai: ~/6.828/lab1                                  sai@sai: ~/6.828/lab1
line to your configuration file "/home/sai/.gdbinit".
For more information about this security protection see the
"Auto-loading safe path" section in the GDB manual.  E.g., run from the shell:
        info "(gdb)Auto-loading safe path"
+ target remote localhost:26000
warning: A handler for the OS ABI "GNU/Linux" is not built into this configuration
of GDB.  Attempting to continue with the default i8086 settings.

The target architecture is assumed to be i8086
[f000:fff0]    0xffff0: ljmp   $0xf000,$0xe05b
0x0000fff0 in ?? ()
+ symbol-file obj/kern/kernel
(gdb) b *0x0010000c
Breakpoint 1 at 0x10000c
(gdb) c
Continuing.
The target architecture is assumed to be i386
=> 0x10000c:    movw   $0x1234,0x472

Breakpoint 1, 0x0010000c in ?? ()
(gdb) x/8h 0x00102050
0x102050:       0x0001  0x0000  0x0000  0x04ad  0x189c  0x0000  0x0001  0x0000
(gdb) x/120h 0x00102050
0x102050:       0x0001  0x0000  0x0000  0x04ad  0x189c  0x0000  0x0001  0x0000
0x102060:       0x0064  0x0000  0x0000  0xf010  0x0012  0x0000  0x0084  0x0000
0x102070:       0x000c  0xf010  0x0000  0x0000  0x0044  0x002c  0x000c  0xf010
0x102080:       0x0000  0x0000  0x0044  0x0039  0x0015  0xf010  0x0000  0x0000
0x102090:       0x0044  0x003a  0x001a  0xf010  0x0000  0x0000  0x0044  0x003c
0x1020a0:       0x001d  0xf010  0x0000  0x0000  0x0044  0x003d  0x0020  0xf010
0x1020b0:       0x0000  0x0000  0x0044  0x003e  0x0025  0xf010  0x0000  0x0000
0x1020c0:       0x0044  0x0043  0x0028  0xf010  0x0000  0x0000  0x0044  0x0044
0x1020d0:       0x002d  0xf010  0x0000  0x0000  0x0044  0x004a  0x002f  0xf010
0x1020e0:       0x0000  0x0000  0x0044  0x004d  0x0034  0xf010  0x0000  0x0000
0x1020f0:       0x0044  0x0050  0x0039  0xf010  0x0000  0x0000  0x0044  0x0053
0x102100:       0x003e  0xf010  0x001f  0x0000  0x0064  0x0002  0x0040  0xf010
0x102110:       0x0031  0x0000  0x003c  0x0000  0x0000  0x0000  0x0040  0x0000
0x102120:       0x0080  0x0000  0x0000  0x0000  0x006a  0x0000  0x0080  0x0000
0x102130:       0x0000  0x0000  0x0084  0x0000  0x0080  0x0000  0x0000  0x0000
(gdb)
```

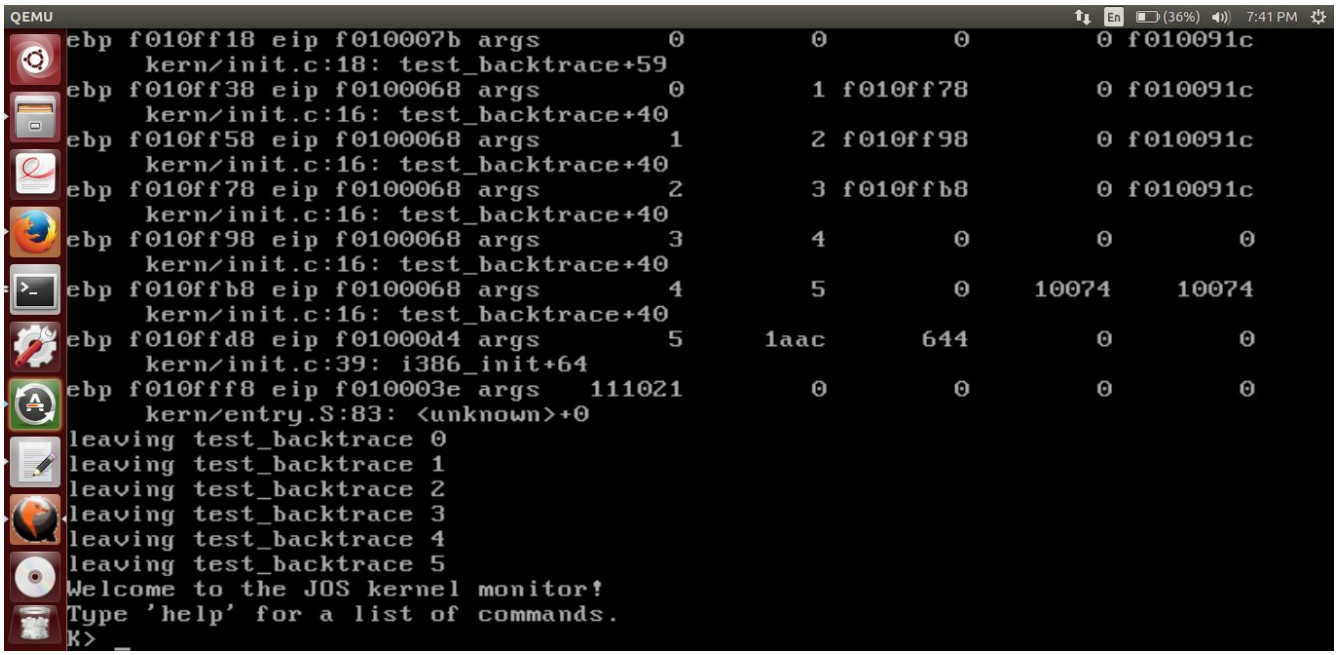


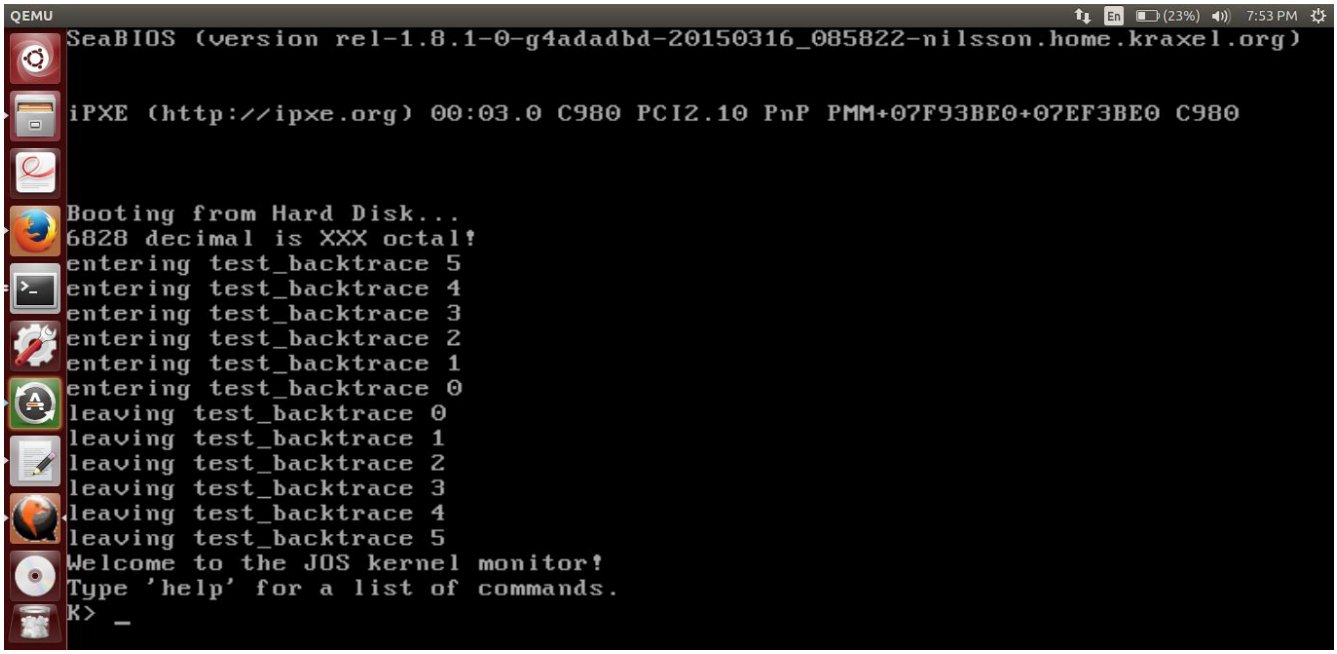

```
mon_backtrace(int argc, char **argv, struct Trapframe *tf)
{
    // Your code here.
uint32_t* ebp = (uint32_t*) read_ebp();
    int i=2;
    //ebp = read_ebp();
    cprintf("Stack backtrace:\n");
while(ebp)
{
cprintf("ebp %8x", ebp);
cprintf(" eip %8x", ebp[1]);
cprintf(" args");
while(i<=6){
cprintf(" %8x", *(ebp+i));
i++;
}
cprintf("\n");
struct Eipdebuginfo info; //This struct defines file,line no,func addr,fn name
debuginfo_eip (ebp[1], &info); //this function gives the info at eip address

// The below code in kdebug.c gives us info about the line number

cprintf("\t%s:%d: %.*s+%d\n",           info.eip_file, info.eip_line,
        info.eip_fn_namelen, info.eip_fn_name,
        ebp[1]-info.eip_fn_addr);

ebp = (uint32_t*) *ebp;
```

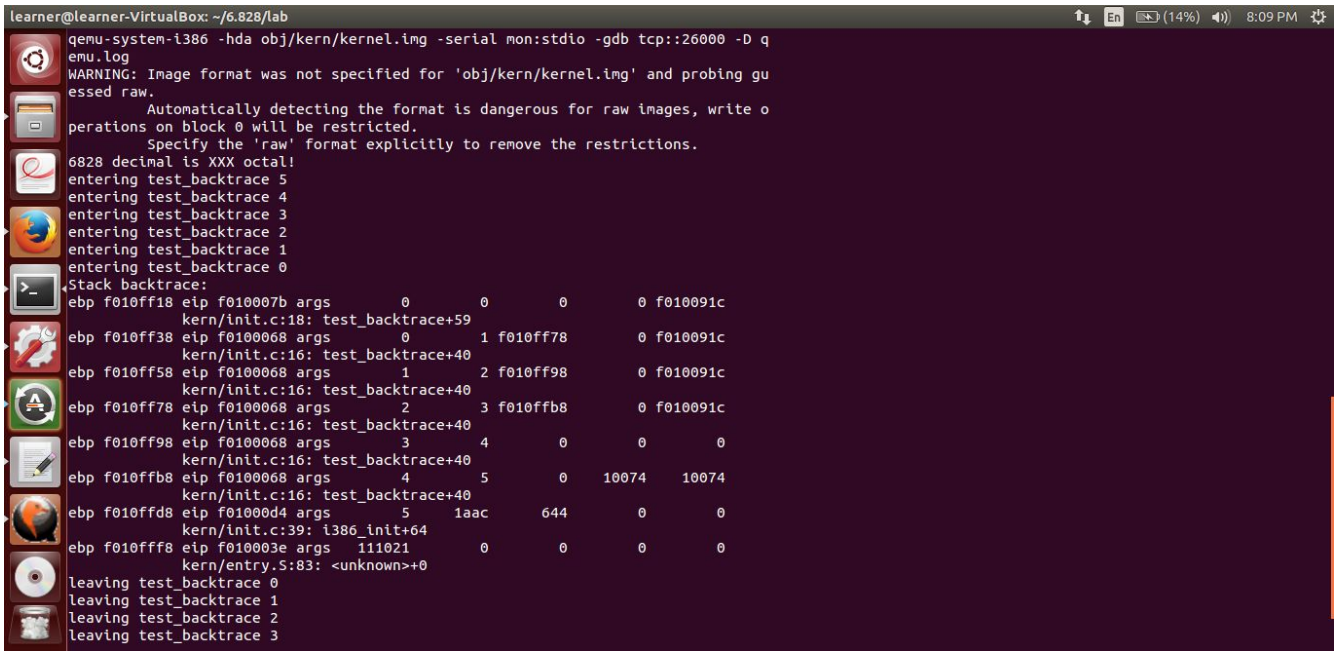
```
i=2;


}
   return 0;
}
```

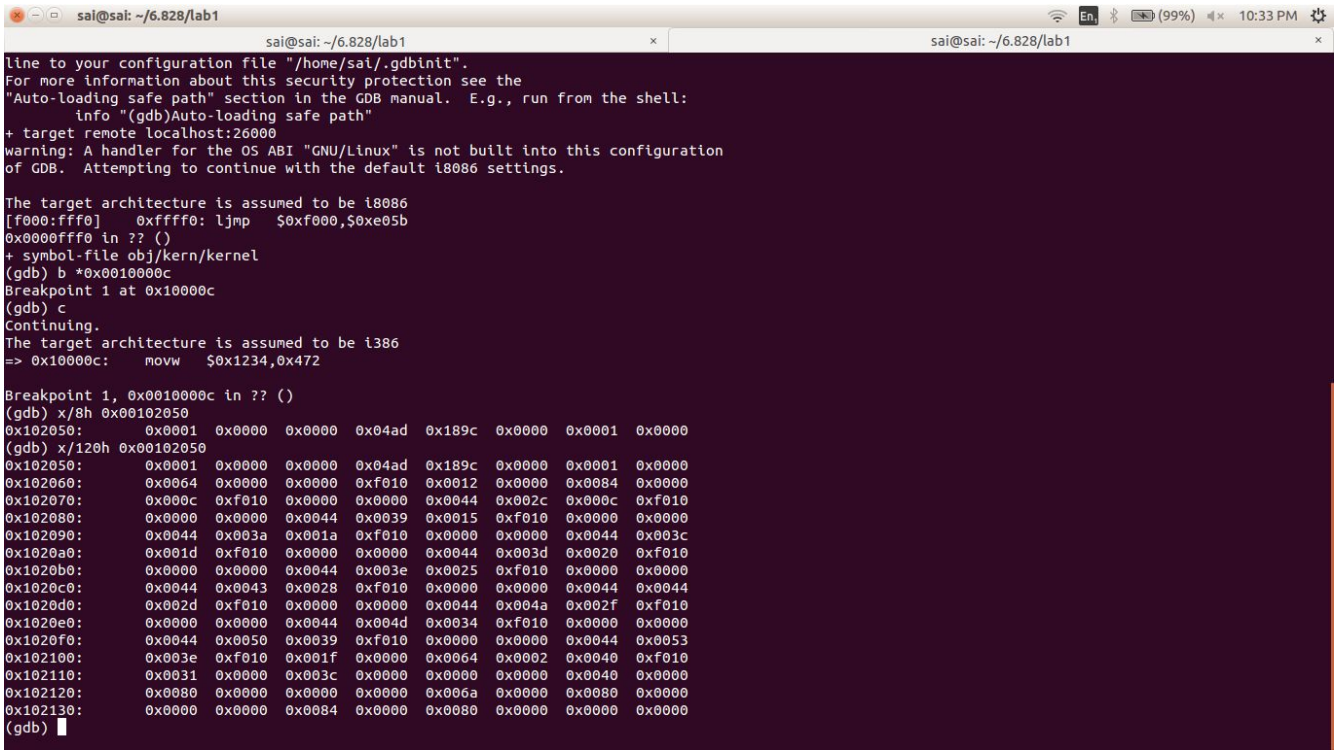
In kdebug.c and in debuginfofunction we have added these lines to print the line number

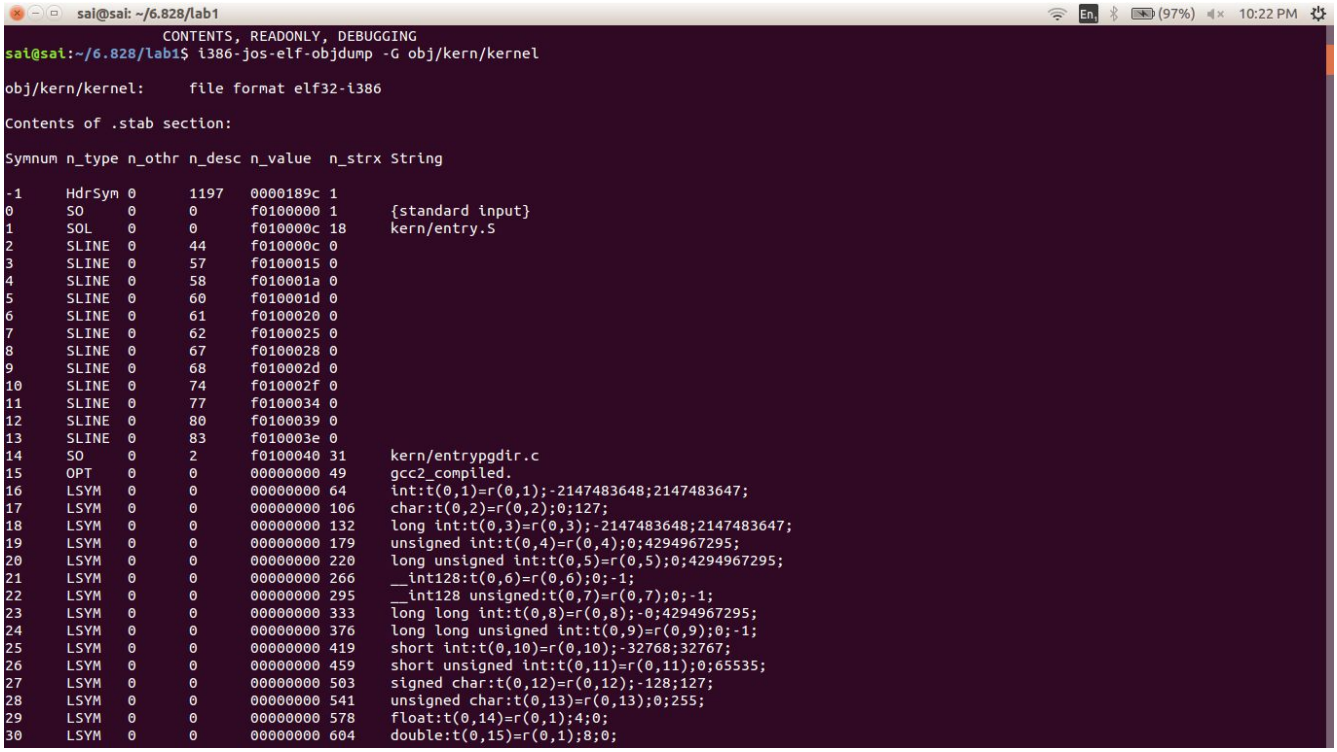
```
// Your code here.

        stab_binsearch (stabs, &lline, &rline, N_SLINE, addr);
        info->eip_line = stabs[lline].n_desc;
#define    N_SLINE       0x44    // text segment line number
```


QEMU
(36%) 7:41 PM
ebp f010ff18 eip f010007b args 0 0 0 0 f010091c
kern/init.c:18: test_backtrace+59
ebp f010ff38 eip f0100068 args 0 1 f010ff78 0 f010091c
kern/init.c:16: test_backtrace+40
ebp f010ff58 eip f0100068 args 1 2 f010ff98 0 f010091c
kern/init.c:16: test_backtrace+40
ebp f010ff78 eip f0100068 args 2 3 f010ffb8 0 f010091c
kern/init.c:16: test_backtrace+40
ebp f010ff98 eip f0100068 args 3 4 0 0 0
kern/init.c:16: test_backtrace+40
ebp f010ffb8 eip f0100068 args 4 5 0 10074 10074
kern/init.c:16: test_backtrace+40
ebp f010ffd8 eip f01000d4 args 5 1aac 644 0 0
kern/init.c:39: i386_init+64
ebp f010fff8 eip f010003e args 111021 0 0 0 0
kern/entry.S:83: <unknown>+0
leaving test_backtrace 0
leaving test_backtrace 1
leaving test_backtrace 2
leaving test_backtrace 3
leaving test_backtrace 4
leaving test_backtrace 5
Welcome to the JOS kernel monitor!
Type 'help' for a list of commands.
K> _


QEMU
(23%) 7:53 PM
SeaBIOS (version rel-1.8.1-0-g4adadbd-20150316_085822-nilsson.home.kraxel.org)
iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+07F93BE0+07EF3BE0 C980
Booting from Hard Disk...
6828 decimal is XXX octal!
entering test_backtrace 5
entering test_backtrace 4
entering test_backtrace 3
entering test_backtrace 2
entering test_backtrace 1
entering test_backtrace 0
leaving test_backtrace 0
leaving test_backtrace 1
leaving test_backtrace 2
leaving test_backtrace 3
leaving test_backtrace 4
leaving test_backtrace 5
Welcome to the JOS kernel monitor!
Type 'help' for a list of commands.
K> _


learner@learner-VirtualBox: ~/6.828/lab
qemu-system-i386 -hda obj/kern/kernel.img -serial mon:stdio -gdb tcp::26000 -D q
emu.log
WARNING: Image format was not specified for 'obj/kern/kernel.img' and probing gu
essed raw.
Automatically detecting the format is dangerous for raw images, write o
perations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.
6828 decimal is XXX octal!
entering test_backtrace 5
entering test_backtrace 4
entering test_backtrace 3
entering test_backtrace 2
entering test_backtrace 1
entering test_backtrace 0
Stack backtrace:
ebp f010ff18 eip f010007b args 0 0 0 0 f010091c
kern/init.c:18: test_backtrace+59
ebp f010ff38 eip f0100068 args 0 0 1 f010ff78 0 f010091c
kern/init.c:16: test_backtrace+40
ebp f010ff58 eip f0100068 args 1 1 2 f010ff98 0 f010091c
kern/init.c:16: test_backtrace+40
ebp f010ff78 eip f0100068 args 2 2 3 f010ffb8 0 f010091c
kern/init.c:16: test_backtrace+40
ebp f010ff98 eip f0100068 args 3 3 4 0 0 0
kern/init.c:16: test_backtrace+40
ebp f010ffb8 eip f0100068 args 4 4 5 0 10074 10074
kern/init.c:16: test_backtrace+40
ebp f010ffd8 eip f01000d4 args 5 5 1aac 644 0 0
kern/init.c:39: i386_init+64
ebp f010fff8 eip f010003e args 111021 0 0 0 0
kern/entry.S:83: <unknown>+0
leaving test_backtrace 0
leaving test_backtrace 1
leaving test_backtrace 2
leaving test_backtrace 3


sai@sai: ~/6.828/lab1
sai@sai: ~/6.828/lab1
sai@sai: ~/6.828/lab1
line to your configuration file "/home/sai/.gdbinit".
For more information about this security protection see the
"Auto-loading safe path" section in the GDB manual. E.g., run from the shell:
info "(gdb)Auto-loading safe path"
+ target remote localhost:26000
warning: A handler for the OS ABI "GNU/Linux" is not built into this configuration
of GDB. Attempting to continue with the default i8086 settings.
The target architecture is assumed to be i8086
[f000:fff0] 0xffff0: ljmp $0xf000,$0xe05b
0x0000fff0 in ?? ()
+ symbol-file obj/kern/kernel
(gdb) b *0x0010000c
Breakpoint 1 at 0x10000c
(gdb) c
Continuing.
The target architecture is assumed to be i386
=> 0x10000c: movw $0x1234,0x472
Breakpoint 1, 0x0010000c in ?? ()
(gdb) x/8h 0x00102050
0x102050: 0x0001 0x0000 0x0000 0x04ad 0x189c 0x0000 0x0001 0x0000
(gdb) x/120h 0x00102050
0x102050: 0x0001 0x0000 0x0000 0x04ad 0x189c 0x0000 0x0001 0x0000
0x102060: 0x0064 0x0000 0x0000 0xf010 0x0012 0x0000 0x0084 0x0000
0x102070: 0x000c 0xf010 0x0000 0x0000 0x0044 0x002c 0x000c 0xf010
0x102080: 0x0000 0x0000 0x0044 0x0039 0x0015 0xf010 0x0000 0x0000
0x102090: 0x0044 0x003a 0x001a 0xf010 0x0000 0x0000 0x0044 0x003c
0x1020a0: 0x001d 0xf010 0x0000 0x0000 0x0044 0x003d 0x0020 0xf010
0x1020b0: 0x0000 0x0000 0x0044 0x003e 0x0025 0xf010 0x0000 0x0000
0x1020c0: 0x0044 0x0043 0x0028 0xf010 0x0000 0x0000 0x0044 0x0044
0x1020d0: 0x002d 0xf010 0x0000 0x0000 0x0044 0x004a 0x002f 0xf010
0x1020e0: 0x0000 0x0000 0x0044 0x004d 0x0034 0xf010 0x0000 0x0000
0x1020f0: 0x0044 0x0050 0x0039 0xf010 0x0000 0x0000 0x0044 0x0053
0x102100: 0x003e 0xf010 0x001f 0x0000 0x0064 0x0002 0x0040 0xf010
0x102110: 0x0031 0x0000 0x003c 0x0000 0x0000 0x0000 0x0040 0x0000
0x102120: 0x0080 0x0000 0x0000 0x0000 0x006a 0x0000 0x0080 0x0000
0x102130: 0x0000 0x0000 0x0084 0x0000 0x0080 0x0000 0x0000 0x0000
(gdb)


sai@sai: ~/6.828/lab1
(97%) 10:22 PM
CONTENTS, READONLY, DEBUGGING
sai@sai:~/6.828/lab1$ i386-jos-elf-objdump -G obj/kern/kernel
obj/kern/kernel: file format elf32-i386
Contents of .stab section:
Symnum n_type n_othr n_desc n_value n_strx String
-1 HdrSym 0 1197 0000189c 1
0 SO 0 0 f0100000 1 {standard input}
1 SOL 0 0 f010000c 18 kern/entry.S
2 SLINE 0 44 f010000c 0
3 SLINE 0 57 f0100015 0
4 SLINE 0 58 f010001a 0
5 SLINE 0 60 f010001d 0
6 SLINE 0 61 f0100020 0
7 SLINE 0 62 f0100025 0
8 SLINE 0 67 f0100028 0
9 SLINE 0 68 f010002d 0
10 SLINE 0 74 f010002f 0
11 SLINE 0 77 f0100034 0
12 SLINE 0 80 f0100039 0
13 SLINE 0 83 f010003e 0
14 SO 0 2 f0100040 31 kern/entrypgdir.c
15 OPT 0 0 00000000 49 gcc2_compiled.
16 LSYM 0 0 00000000 64 int:t(0,1)=r(0,1);-2147483648;2147483647;
17 LSYM 0 0 00000000 106 char:t(0,2)=r(0,2);0;127;
18 LSYM 0 0 00000000 132 long int:t(0,3)=r(0,3);-2147483648;2147483647;
19 LSYM 0 0 00000000 179 unsigned int:t(0,4)=r(0,4);0;4294967295;
20 LSYM 0 0 00000000 220 long unsigned int:t(0,5)=r(0,5);0;4294967295;
21 LSYM 0 0 00000000 266 __int128:t(0,6)=r(0,6);0;-1;
22 LSYM 0 0 00000000 295 __int128 unsigned:t(0,7)=r(0,7);0;-1;
23 LSYM 0 0 00000000 333 long long int:t(0,8)=r(0,8);-0;4294967295;
24 LSYM 0 0 00000000 376 long long unsigned int:t(0,9)=r(0,9);0;-1;
25 LSYM 0 0 00000000 419 short int:t(0,10)=r(0,10);-32768;32767;
26 LSYM 0 0 00000000 459 short unsigned int:t(0,11)=r(0,11);0;65535;
27 LSYM 0 0 00000000 503 signed char:t(0,12)=r(0,12);-128;127;
28 LSYM 0 0 00000000 541 unsigned char:t(0,13)=r(0,13);0;255;
29 LSYM 0 0 00000000 578 float:t(0,14)=r(0,1);4;0;
30 LSYM 0 0 00000000 604 double:t(0,15)=r(0,1);8;0;

```
sai@sai: ~/6.828/lab1                                                    (97%) 10:21 PM
sai@sai:~$ cd 6.828/lab1/
sai@sai:~/6.828/lab1$ i386-jos-elf-objdump -h obj/kern/kernel

obj/kern/kernel:     file format elf32-i386

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         000018d1  f0100000  00100000  00001000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .rodata       00000770  f01018e0  001018e0  000028e0  2**5
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  2 .stab         00003829  f0102050  00102050  00003050  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  3 .stabstr      0000189d  f0105879  00105879  00006879  2**0
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .data         0000a300  f0108000  00108000  00009000  2**12
                  CONTENTS, ALLOC, LOAD, DATA
  5 .bss          00000644  f0112300  00112300  00013300  2**5
                  ALLOC
  6 .comment      00000011  00000000  00000000  00013300  2**0
                  CONTENTS, READONLY
  7 .debug_info   000005d2  00000000  00000000  00013311  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_abbrev 00000253  00000000  00000000  000138e3  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .debug_loc    00000c95  00000000  00000000  00013b36  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .debug_aranges 00000040  00000000  00000000  000147cb  2**0
                  CONTENTS, READONLY, DEBUGGING
 11 .debug_ranges 00000078  00000000  00000000  0001480b  2**0
                  CONTENTS, READONLY, DEBUGGING
 12 .debug_line   00000141  00000000  00000000  00014883  2**0
                  CONTENTS, READONLY, DEBUGGING
 13 .debug_str    000001b8  00000000  00000000  000149c4  2**0
                  CONTENTS, READONLY, DEBUGGING
sai@sai:~/6.828/lab1$
```

```
sai@sai: ~/6.828/lab1                                                    (98%) 10:28 PM
sai@sai: ~/6.828/lab1                                  sai@sai: ~/6.828/lab1
        add-auto-load-safe-path /home/sai/6.828/lab1/.gdbinit
line to your configuration file "/home/sai/.gdbinit".
To completely disable this security protection add
        set auto-load safe-path /
line to your configuration file "/home/sai/.gdbinit".
For more information about this security protection see the
"Auto-loading safe path" section in the GDB manual.  E.g., run from the shell:
        info "(gdb)Auto-loading safe path"
+ target remote localhost:26000
warning: A handler for the OS ABI "GNU/Linux" is not built into this configuration
of GDB.  Attempting to continue with the default i8086 settings.

The target architecture is assumed to be i8086
[f000:fff0]    0xffff0: ljmp   $0xf000,$0xe05b
0x0000fff0 in ?? ()
+ symbol-file obj/kern/kernel
(gdb) b *0xf0100040
Breakpoint 1 at 0xf0100040: file kern/init.c, line 13.
(gdb) c
Continuing.
The target architecture is assumed to be i386
=> 0xf0100040 <test_backtrace>: push   %ebp

Breakpoint 1, test_backtrace (x=5) at kern/init.c:13
13      {
(gdb) x/96h 0xf0102050
0xf0102050:     0x0001  0x0000  0x0000  0x04ad  0x189c  0x0000  0x0001  0x0000
0xf0102060:     0x0064  0x0000  0x0000  0xf010  0x0012  0x0000  0x0084  0x0000
0xf0102070:     0x000c  0xf010  0x0000  0x0000  0x0044  0x002c  0x000c  0xf010
0xf0102080:     0x0000  0x0000  0x0044  0x0039  0x0015  0xf010  0x0000  0x0000
0xf0102090:     0x0044  0x003a  0x001a  0xf010  0x0000  0x0000  0x0044  0x003c
0xf01020a0:     0x001d  0xf010  0x0000  0x0000  0x0044  0x003d  0x0020  0xf010
0xf01020b0:     0x0000  0x0000  0x0044  0x003e  0x0025  0xf010  0x0000  0x0000
0xf01020c0:     0x0044  0x0043  0x0028  0xf010  0x0000  0x0000  0x0044  0x0044
0xf01020d0:     0x002d  0xf010  0x0000  0x0000  0x0044  0x004a  0x002f  0xf010
0xf01020e0:     0x0000  0x0000  0x0044  0x004d  0x0034  0xf010  0x0000  0x0000
0xf01020f0:     0x0044  0x0050  0x0039  0xf010  0x0000  0x0000  0x0044  0x0053
0xf0102100:     0x003e  0xf010  0x001f  0x0000  0x0064  0x0002  0x0040  0xf010
(gdb)
```