

# Princeton University

## COS 217: Introduction to Programming Systems

### Example of Using the gdb Debugger

### For an IA-32 Assembly Language Program

```
% cat junk.txt
one
two
three
four
% gcc -g -Wall -ansi -pedantic -o mysort mysort.c quicksort.s partition.s swap.s
% xemacs
<Esc> x gdb <Enter> mysort <Enter>
(gdb) break main
(gdb) break swap
(gdb) run < junk.txt
(gdb) continue
```

*At this point, execution is paused after the prolog of swap.*

```
(gdb) print/a $ebp
bfff8ad8
(gdb) x/a 0xbfff8ad8
bfff8f18
(gdb) x/a 0xbfff8adc
080486af
(gdb) x/a 0xbfff8ae0
095b6048
(gdb) x/d 0xbfff8ae4
0
(gdb) x/d 0xbfff8ae8
3
(gdb) x/a 0x095b6048
095b6018
(gdb) x/a 0x095b604c
095b6028
(gdb) x/a 0x095b6050
095b6038
(gdb) x/a 0x095b6054
095b6008
(gdb) x/c 0x095b6018
o
(gdb) x/s 0x095b6018
one
(gdb) x/c 0x095b6028
t
(gdb) x/s 0x095b6028
two
(gdb) x/c 0x095b6038
t
(gdb) x/s 0x095b6028
three
(gdb) x/c 0x095b6008
f
(gdb) x/s 0x095b6028
four
```

## Registers

Register	Contents
...	
EBP	bfff8ad8
...	

## Memory

Address	Contents	Explanation
...		
095b6008	66	'f'
095b6009	6f	'o'
095b600a	75	'u'
095b600b	72	'r'
095b600c	00	'\0'
...		
095b6018	6f	'o'
095b6019	6e	'n'
095b601a	65	'e'
095b601b	00	'\0'
...		
095b6028	74	't'
095b6029	77	'w'
095b602a	6f	'o'
095b602b	00	'\0'
...		
095b6038	74	't'
095b6039	68	'h'
095b603a	72	'r'
095b603b	65	'e'
095b603c	65	'e'
095b603d	00	'\0'
...		
095b6048	18	ppcArray[0]
095b6049	60	
095b604a	5b	
095b604b	09	
095b604c	28	ppcArray[1]
095b604d	60	
095b604e	5b	
095b604f	09	
095b6050	38	ppcArray[2]
095b6051	60	
095b6052	5b	
095b6053	09	
095b6054	08	ppcArray[3]
095b6055	60	
095b6056	5b	
095b6057	09	
...		
bfff8ad8	18	old EBP
bfff8ad9	8f	
bfff8ada	ff	
bfff8adb	bf	
bfff8adc	af	return address
bfff8add	86	
bfff8ade	04	
bfff8adf	08	
bfff8ae0	48	ppcArray
bfff8ae1	60	
bfff8ae2	5b	
bfff8ae3	09	
bfff8ae4	00	iOne
bfff8ae5	00	
bfff8ae6	00	
bfff8ae7	00	
bfff8ae8	03	iTwo
bfff8ae9	00	
bfff8aea	00	
bfff8aeb	00	
...		