Assignment #1

Wes McEvoy
01/27/20
ECEN 1310

Problems: 1.4, 1.8, 1.11, 1.14

1.4)

int* x	1004	1008
int b	1	1004
int a	2	1000

1.8)

int b	3	1016
int a	3	1012
int **z	1004	1008
int * y	1012	1004
int * x	1012	1000

line 3 - int **
line 4 - int *
line 5 - int *
line 6 - int
line 7 - int
line 8 - int
line 9 - int

1.11) The code wouldn't crash the program but it would result in garbage in the memory locations of y, x, and b.
The variation will cause a segmentation fault on line 8 since *y tries to point to memory address 'NULL' which is outside of the range.

1.14) after line 3:

int *x	1000	1004
int a	?	1060

after main 6:

int *x	1000	1004
int a	18	1000

before first return:

int sum	6	1028
void *pc	line 4+	1024
int rv	6	1020
int c	3	1016
int b	2	1012
int a	1	1008
int * x	1000	1004
int a	?	1000
void* pc	"system"	996
int rv	⊗	992

before second return:

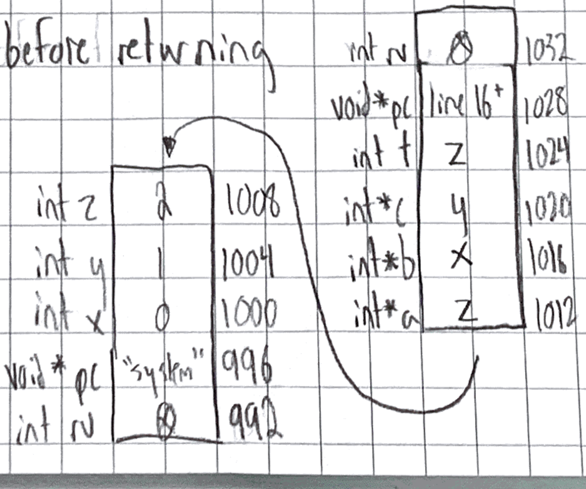
int sum	18	1028
void *pc	line 5+	1024
int rv	18	1020
int c	6	1016
int b	6	1012
int a	6	1008
int *x	1000	1004
int a	6	1000
void *pc	"system"	996
int rv	⊗	992

1.20) before first call:

int z	2	1008
int y	1	1004
int x	0	1000

continued on back

before returning



int rv	⊗	1032
void*pc	line 16+	1028
int t	z	1024
int*c	y	1020
int*b	x	1016
int*a	z	1012

int z	2	1008
int y	1	1004
int x	0	1000
void* pc	"system"	996
int rv	⊗	992

The swap function can be called to switch 2 values while the third remains the same. Since the return values of the function will be slightly different, the arguments must be modified. In order to swap only two values, the same address is passed in twice.

Original:

```
int main(){
  int x=0, y=1, z=2;
  swap(&x,&y,&z);
  assert(x==2);
  assert(y==0);
  assert(z==1);
  return 0;
}
```

New:

```
int main(){
int x=0, y=1, z=2;
swap(&x, &x, &z);
assert(x==2);
assert(y==1);
assert(z==0);
return 0;
}
```

Note: -&x passed in twice
- y retains its value
- No change to the function itself