

A photograph of a campus scene. In the foreground, there are several tulips in shades of red and yellow. Behind them is a paved area, possibly a walkway or road. In the background, there are green lawns, trees with sparse leaves, and a white building. The right side of the image is overlaid with a dark red semi-transparent rectangle containing text.

CS197C

Intro to C

Nicolas Scarci - *October 5, 2016*

Segmentation

Code

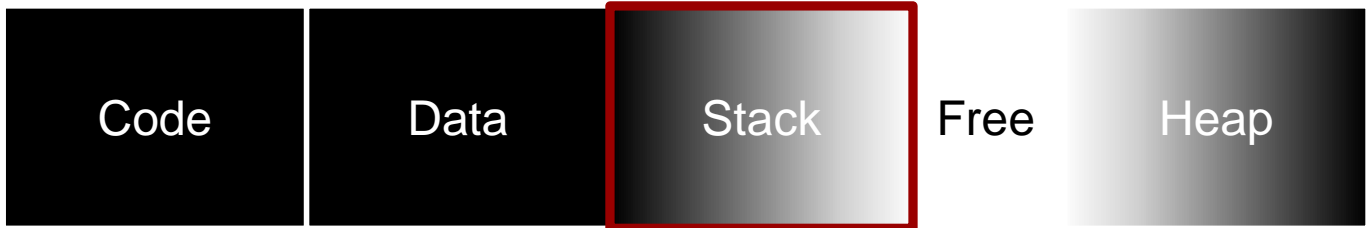
Data

Stack

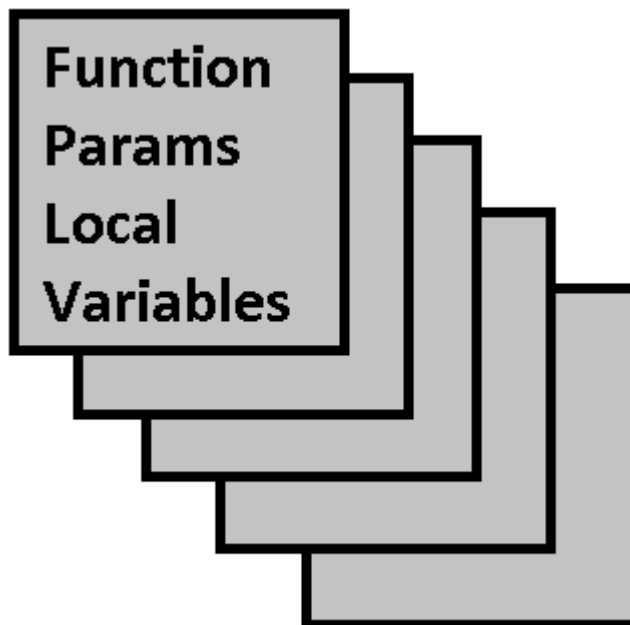
Free

Heap

Segmentation



Stack Segment



Frames

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int main(int argc, char * argv[]){
    int op1=5;
    int op2=6;
    int answer = add(op1,op2);
    printf ("%i+%i=%i\n", op1, op2, answer);
    return 0;
}

int add(int op1, int op2){
    return op1+op2;
}

"stack.c" 13L, 204C      1,1      A

```

main

argc	1
argv	0xbf825794

Frames

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int main(int argc, char * argv[]){
    int op1=5;
    int op2=6;
    int answer = add(op1,op2);
    printf("%i+%i=%i\n",op1,op2,answer);
    return 0;
}

int add(int op1, int op2){
    return op1+op2;
}

"stack.c" 13L, 204C      1,1      A

```

main

argc	1
argv	0xbf825794
op1	5

Frames

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int main(int argc, char * argv[]){
    int op1=5;
    int op2=6;
    int answer = add(op1,op2);
    printf("%i+%i=%i\n",op1,op2,answer);
    return 0;
}

int add(int op1, int op2){
    return op1+op2;
}

"stack.c" 13L, 204C      1,1      A
  
```

main

argc	1
argv	0xbf825794
op1	5
op2	6

Frames

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int main(int argc, char * argv[]){
    int op1=5;
    int op2=6;
    int answer = add(op1,op2);
    printf("%i+%i=%i\n",op1,op2,answer);
    return 0;
}

int add(int op1, int op2){
    return op1+op2;
}
"stack.c" 13L, 204C      1,1      A

```

main

argc	1
argv	0xbf825794
op1	5
op2	6
answer	

Frames

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int main(int argc, char * argv[]){
    int op1=5;
    int op2=6;
    int answer = add(op1,op2);
    printf("%i+%i=%i\n",op1,op2,answer);
    return 0;
}

```

```

int add(int op1, int op2){
    return op1+op2;
}

```

"stack.c" 13L, 204C 1,1

add		
op1	5	
op2	6	
		94
A:	answer	

Frames

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int main(int argc, char * argv[]){
    int op1=5;
    int op2=6;
    int answer = add(op1,op2);
    printf("%i+%i=%i\n",op1,op2,answer);
    return 0;
}

int add(int op1, int op2){
    return op1+op2;
}
"stack.c" 13L, 204C      1,1      A

```

main

argc	1
argv	0xbf825794
op1	5
op2	6
answer	11

Passing By Reference

```
elinux.cs.umass.edu - PuTTY  
#include<stdio.h>  
  
int main(int argc, char * argv[]){  
    int op1=5;  
    int op2=6;  
    int answer;  
    add(op1,op2,&answer);  
    printf("%i+%i=%i\n",op1,op2,answer);  
    return 0;  
}  
  
int add(int op1, int op2, int * ansRef){  
    *ansRef=op1+op2;  
    return 0;  
}  
"ref.c" 15L, 238C      1,1      All
```

main	
argc	1
argv	0xbf825794
op1	5
op2	6
answer	

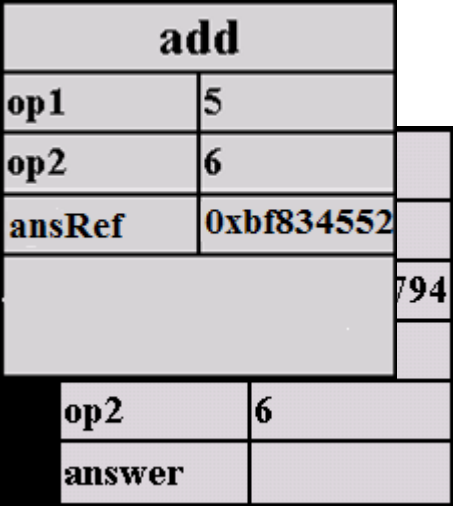
Passing By Reference

```
elinux.cs.umass.edu - PuTTY
#include<stdio.h>

int main(int argc, char * argv[]){
    int op1=5;
    int op2=6;
    int answer;
    add(op1,op2,&answer);
    printf("%i+%i=%i\n",op1,op2,answer);
    return 0;
}

int add(int op1, int op2, int * ansRef){
    *ansRef=op1+op2;
    return 0;
}

"ref.c" 15L, 238C      1,1      All
```



Passing By Reference

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

int main(int argc, char * argv[]){
    int op1=5;
    int op2=6;
    int answer;
    add(op1,op2,&answer);
    printf("%i+%i=%i\n",op1,op2,answer);
    return 0;
}

int add(int op1, int op2, int * ansRef){
    *ansRef=op1+op2;
    return 0;
}

"ref.c" 15L, 238C          1,1          All

```

add	
op1	5
op2	6
ansRef	0xbf834552
	794
op2	6
answer	11

Passing By Reference

```
elinux.cs.umass.edu - PuTTY
#include<stdio.h>

int main(int argc, char * argv[]){
    int op1=5;
    int op2=6;
    int answer;
    add(op1,op2,&answer);
    printf("%i+%i=%i\n",op1,op2,answer);
    return 0;
}

int add(int op1, int op2, int * ansRef){
    *ansRef=op1+op2;
    return 0;
}

"ref.c" 15L, 238C      1,1      All
```

main	
argc	1
argv	0xbf825794
op1	5
op2	6
answer	11

Local Address

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

int * localAddress() {
    int a =5;
    return &a;
}

int main(int argc, char * argv[]) {
    int * aAddr = localAddress();
    printf("%p:%i\n", aAddr, *aAddr);
    printf("%p:%i\n", aAddr, *aAddr);
    return 0;
}
"localAddress.c" 13L, 212C 1,

```

main	
argc	1
argv	0xbf825

Local Address

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

int * localAddress() {
    int a =5;
    return &a;
}

int main(int argc, char * argv[]) {
    int * aAddr = localAddress();
    printf("%p:%i\n", aAddr, *aAddr);
    printf("%p:%i\n", aAddr, *aAddr);
    return 0;
}
"localAddress.c" 13L, 212C 1,

```

main	
argc	1
argv	0xbf825
aAddr	

Local Address

```
elinux.cs.umass.edu - PuTTY
#include<stdio.h>

int * localAddress() {
    int a =5;
    return &a;
}

int main(int argc, char * argv[]){
    int * aAddr = localAddress();
    printf("%p:%i\n",aAddr,*aAddr);
    printf("%p:%i\n",aAddr,*aAddr);
    return 0;
}
"localAddress.c" 13L, 212C
```

localAddress	
a	5

Local Address

elinux.cs.umass.edu - PuTTY

```
#include<stdio.h>
```

```
int * localAddress() {
    int a =5;
    return &a;
}
```

```
int main(int argc, char * argv[]) {
    int * aAddr = localAddress();
    printf("%p:%i\n", aAddr, *aAddr);
    printf("%p:%i\n", aAddr, *aAddr);
    return 0;
}
"localAddress.c" 13L, 212C
```

main	
argc	1
argv	0xbf825
aAddr	0

Local Address

elinux7 experimental) - PuTTY

```
elinux7 experimental) > gcc localAddress.c
localAddress.c: In function 'localAddress':
localAddress.c:5:9: warning: function returns address of local variable [-Wreturn-local-addr]
    return &a;
           ^
elinux7 experimental) > ./a.out
Segmentation fault (core dumped)
elinux7 experimental) > █
```

Segmentation Fault

elinux.cs.umass.edu - PuTTY

```
#include<stdio.h>

int main(int argc, char * argv[]){
    int * anAddr;
    printf("%i\n",anAddr);
    *anAddr=5;
    printf("%i\n",*anAddr);
    return 0;
}
```

"segfault.c" 9L, 143C

1,1

main	
argc	1
argv	0xbf825
anAddr	0

NullPointerException

elinux.cs.umass.edu - PuTTY

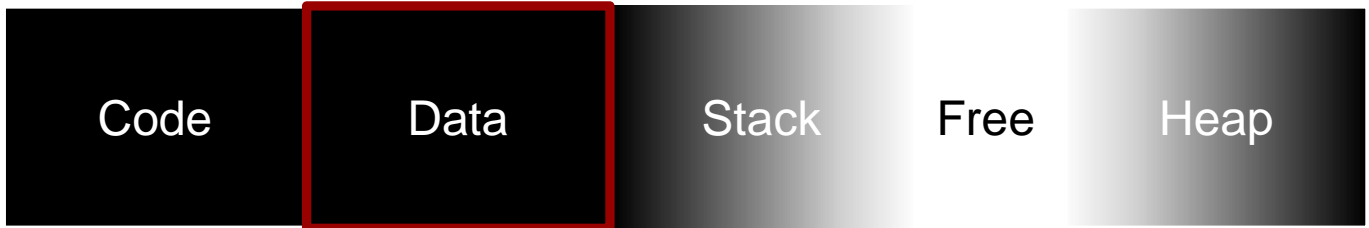
```

elinux7 experimental) > gcc segfault.c
segfault.c: In function 'main':
segfault.c:5:9: warning: format '%i' expects argument of type
e 'int', but argument 2 has type 'int *' [-Wformat=]
    printf("%i\n", anAddr);
        ^
elinux7 experimental) > ./a.out
0
Segmentation fault (core dumped)
elinux7 experimental) >

```

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEF (backspace)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g

Segmentation



Static Variables

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int main(int argc, char * argv[]){
    printf("%i\n",incStaticVar());
    printf("%i\n",incStaticVar());
    return 0;
}

int incStaticVar(){
    static int i=5;
    return i++;
}
"static.c" 12L, 184C
  
```

main	
argc	1
argv	0xbf825

Static Variables

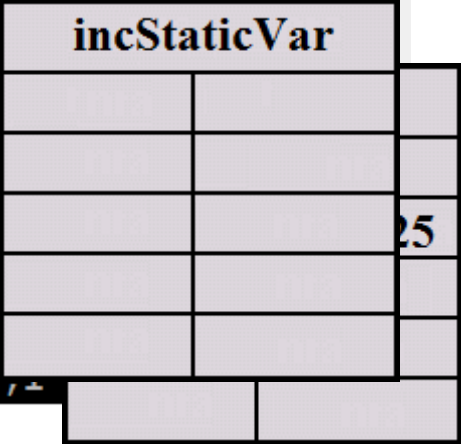
elinux.cs.umass.edu - PuTTY

```
#include<stdio.h>

int main(int argc, char * argv[]){
    printf("%i\n",incStaticVar());
    printf("%i\n",incStaticVar());
    return 0;
}

int incStaticVar(){
    static int i=5;
    return i++;
}

"static.c" 12L, 184C
```



Static Variables

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int main(int argc, char * argv[]){
    printf("%i\n",incStaticVar());
    printf("%i\n",incStaticVar());
    return 0;
}

int incStaticVar(){
    static int i=5;
    return i++;
}
"static.c" 12L, 184C

```

data	
i	5

incStaticVar		
		25

Static Variables

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int main(int argc, char * argv[]){
    printf("%i\n",incStaticVar());
    printf("%i\n",incStaticVar());
    return 0;
}

int incStaticVar(){
    static int i=5;
    return i++;
}
"static.c" 12L, 184C

```

data	
i	6

incStaticVar		
		25

Static Variables

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int main(int argc, char * argv[]){
    printf("%i\n",incStaticVar());
    printf("%i\n",incStaticVar());
    return 0;
}

int incStaticVar(){
    static int i=5;
    return i++;
}
"static.c" 12L, 184C 1,1

```

data	
i	6

main	
argc	1
argv	0xbf825

Static Variables

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int main(int argc, char * argv[]){
    printf("%i\n",incStaticVar());
    printf("%i\n",incStaticVar());
    return 0;
}

int incStaticVar(){
    static int i=5;
    return i++;
}
"static.c" 12L, 184C

```

data	
i	6

incStaticVar		
		25

Static Variables

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int main(int argc, char * argv[]){
    printf("%i\n",incStaticVar());
    printf("%i\n",incStaticVar());
    return 0;
}

int incStaticVar(){
    static int i=5;
    return i++;
}
"static.c" 12L, 184C

```

data	
i	7

incStaticVar		
		25

Static Variables

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int main(int argc, char * argv[]){
    printf("%i\n",incStaticVar());
    printf("%i\n",incStaticVar());
    return 0;
}

int incStaticVar(){
    static int i=5;
    return i++;
}
"static.c" 12L, 184C 1,1

```

data	
i	7

main	
argc	1
argv	0xbf825

Static Variables

elinux.cs.umass.edu - PuTTY

— □ ×

```
elinux7 experimental) > gcc static.c
static.c: In function 'main':
static.c:4:16: warning: implicit declaration of function
incStaticVar' [-Wimplicit-function-declaration]
    printf("%i\n",incStaticVar());
                   ^
elinux7 experimental) > ./a.out
5
6
elinux7 experimental) > █
```

Global Variables

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int i=5;

int main(int argc, char * argv[]){
    printf("%i\n",i);
    incGlobalVar();
    printf("%i\n",i);
    return 0;
}

int incGlobalVar(){
    i++;
    return 0;
}
"global.c" 15L, 172C

```

data	
i	5

main	
argc	1
argv	0xbf825

Global Variables

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int i=5;

int main(int argc, char * argv[]){
    printf("%i\n",i);
    incGlobalVar();
    printf("%i\n",i);
    return 0;
}

int incGlobalVar(){
    i++;
    return 0;
}

```

data	
i	5

main	
argc	1
argv	0xbf825

"global.c" 15L, 172C

1,1

All

Global Variables

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int i=5;

int main(int argc, char * argv[]){
    printf("%i\n",i);
    incGlobalVar();
    printf("%i\n",i);
    return 0;
}

int incGlobalVar(){
    i++;
    return 0;
}

```



"global.c" 15L, 172C

1,1

All

data	
i	5

incGlobalVar		
		f825

Global Variables

elinux.cs.umass.edu - PuTTY

```

#include<stdio.h>

int i=5;

int main(int argc, char * argv[]){
    printf("%i\n",i);
    incGlobalVar();
    printf("%i\n",i);
    return 0;
}

int incGlobalVar(){
    i++;
    return 0;
}
"global.c" 15L, 172C

```

data	
i	6

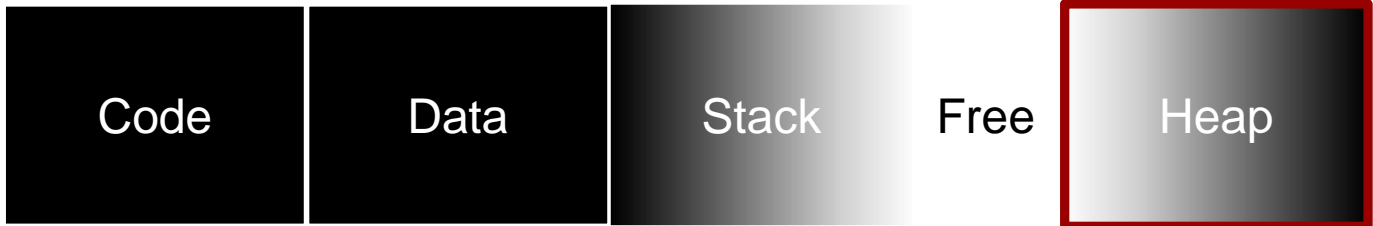
main	
argc	1
argv	0xbf825

Global Variables


elinux.cs.umass.edu - PuTTY

```
elinux7 experimental) > gcc global.c
global.c: In function 'main':
global.c:7:2: warning: implicit declaration of function 'incGlobalVar' [-Wimplicit-function-declaration]
    incGlobalVar();
    ^
elinux7 experimental) > ./a.out
5
6
elinux7 experimental) > █
```

Segmentation



Heap segment

 elnux.cs.umass.edu - PuTTY


```
#include<stdio.h>
#include<stdlib.h>

int main(int argc, char * argv[]){
    void * startAddr = malloc(sizeof(int)+sizeof(char));
    int * intAddr = (int *) startAddr;
    char * charAddr = (char *) (intAddr+1);
    *intAddr=5;
    *charAddr='N';
    printf("%p:\t%i\n",intAddr,*intAddr);
    printf("%p:\t%c\n",charAddr,*charAddr);
    free(startAddr);
    return 0;
}
"malloc.c" 14L, 343C
```

heap	
0x237c0	
0x237c1	
0x237c2	
0x237c3	
0x237c4	

main	
argc	1
argv	0xbf825
startAddr	0x237c0

Heap segment

 elnux.cs.umass.edu - PuTTY

```
#include<stdio.h>
#include<stdlib.h>

int main(int argc, char * argv[]){
    void * startAddr = malloc(sizeof(int)+sizeof(char));
    int * intAddr = (int *) startAddr;
    char * charAddr = (char *) (intAddr+1);
    *intAddr=5;
    *charAddr='N';
    printf("%p:\t%i\n",intAddr,*intAddr);
    printf("%p:\t%c\n",charAddr,*charAddr);
    free(startAddr);
    return 0;
}
```

"malloc.c" 14L, 343C

heap

0x237c0

0x237c1

0x237c2

0x237c3

0x237c4

main

argc

1

argv

0xbf825


startAddr

0x237c0

intAddr

0x237c0

Heap segment

 elnux.cs.umass.edu - PuTTY

```
#include<stdio.h>
#include<stdlib.h>


int main(int argc, char * argv[]){
    void * startAddr = malloc(sizeof(int)+sizeof(char));
    int * intAddr = (int *) startAddr;
    char * charAddr = (char *) (intAddr+1);
    *intAddr=5;
    *charAddr='N';
    printf("%p:\t%i\n",intAddr,*intAddr);
    printf("%p:\t%c\n",charAddr,*charAddr);
    free(startAddr);
    return 0;
}
```

"malloc.c" 14L, 343C

heap	
0x237c0	
0x237c1	
0x237c2	
0x237c3	
0x237c4	

main	
argc	1
argv	0xbf825
startAddr	0x237c0
intAddr	0x237c0
charAddr	0x237c4

Heap segment

 elnux.cs.umass.edu - PuTTY


```
#include<stdio.h>
#include<stdlib.h>

int main(int argc, char * argv[]){
    void * startAddr = malloc(sizeof(int)+sizeof(char));
    int * intAddr = (int *) startAddr;
    char * charAddr = (char *) (intAddr+1);
    *intAddr=5;
    *charAddr='N';
    printf("%p:\t%i\n",intAddr,*intAddr);
    printf("%p:\t%c\n",charAddr,*charAddr);
    free(startAddr);
    return 0;
}
"malloc.c" 14L, 343C
```

heap	
0x237c0	5
0x237c1	
0x237c2	
0x237c3	
0x237c4	

main	
argc	1
argv	0xbf825
startAddr	0x237c0
intAddr	0x237c0
charAddr	0x237c4

Heap segment

 elnux.cs.umass.edu - PuTTY

```
#include<stdio.h>
#include<stdlib.h>

int main(int argc, char * argv[]){
    void * startAddr = malloc(sizeof(int)+sizeof(char));
    int * intAddr = (int *) startAddr;
    char * charAddr = (char *) (intAddr+1);
    *intAddr=5;
    *charAddr='N';
    printf("%p:\t%i\n",intAddr,*intAddr);
    printf("%p:\t%c\n",charAddr,*charAddr);
    free(startAddr);
    return 0;
}
"malloc.c" 14L, 343C
```

heap	
0x237c0	5
0x237c1	
0x237c2	
0x237c3	
0x237c4	N

main	
argc	1
argv	0xbf825
startAddr	0x237c0
intAddr	0x237c0
charAddr	0x237c4

Heap segment

elinux.cs.umass.edu - PuTTY

```
elinux7 experimental) > gcc malloc.c  
elinux7 experimental) > ./a.out  
0x237c010:      5  
0x237c014:      N  
elinux7 experimental) > █
```

Your assignment

```

XXXXXXXXXXXX
X      *    XX
X  X^XX  XX
XPX  :X  XX
X\ XXXX:  XX
XX/   *X  D
XXX^ XXX* X
XXX^ XX   X
X/                XX
XXXXXXXXXXXX

```

Segmentation



A horizontal diagram showing five memory segments: Code, Data, Stack, Free, and Heap. The Code segment is highlighted with a red border. The Stack and Heap segments have a vertical gradient from light to dark. The Free segment is a white gap between the Stack and Heap.

Code

Data

Stack

Free

Heap

Function Pointers

1. JAVA abstract method
 - `int triple(int arg);`
2. C Function prototype
 - `int triple(int);`
3. Compatible C Function Pointer
 - `int (*fptr)(int);`

Function pointers

code

triple	0x406a2
third	0x406b4

main

argc	1
argv	0xbf825
array	0x7ffdb
0x7ffdb	1
0x7ffe0	2

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C                                1,1                                All

```

Function pointers

code	
triple	0x406a2
third	0x406b4
map	
func	0x406a2
array	0x7ffdb
length	2
i	0
0x/ffe0	2

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C
1,1 All

```


Function pointers

code	
triple	0x406a2
third	0x406b4
map	
func	0x406a2
array	0x7ffdb
length	2
i	0
0x/ffe0	2

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C                                1,1                                All

```

Function pointers

code	
triple	0x406a2
third	0x406b4
map	
main	
argc	1
argv	0xbf825
array	0x7ffdb
0x7ffdb	1
0x7ffe0	2

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C                                1,1                                All

```

Function pointers

code	
triple	0x406a2
third	0x406b4

triple	
a	1
0x/ffe0	2

```
elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C      1,1      All
```

Function pointers

code	
triple	0x406a2
third	0x406b4

map	

main	
argc	1
argv	0xbf825
array	0x7ffdb
0x7ffdb	3
0x7ffe0	2

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C                               1,1          All

```

Function pointers

code	
triple	0x406a2
third	0x406b4



map	
func	0x406a2
a array	0x7ffdb
a length	2
a i	1
0	
0x/neo	2

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C
1,1 All

```

Function pointers

code	
triple	0x406a2
third	0x406b4



map	
main	
argc	1
argv	0xbf825
array	0x7ffdb
0x7ffdb	3
0x7ffe0	2

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C
1,1 All

```

Function pointers

code	
triple	0x406a2
third	0x406b4

triple	
a	2
f	
a	a
a	l
a	i
0	
0x/nev	2

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C      1,1      All

```

Function pointers

code	
triple	0x406a2
third	0x406b4

map	

main	
argc	1
argv	0xbf825
array	0x7ffdb
0x7ffdb	3
0x7ffe0	6

```
elinux.cs.umass.edu - PuTTY
```

```
#include<stdio.h>
```

```
void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}
```

```
void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}
```

```
int triple(int a){ return a*3; }
```

```
int third (int a){ return a/3; }
```

```
int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}
```

```
"fptr.c" 26L, 481C
```

```
1,1
```

```
All
```


Function pointers

code	
triple	0x406a2
third	0x406b4

map	
func	0x406a2
array	0x7ffdb
length	2
i	2
0x7ffe0	6

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C                               1,1                               All

```

Function pointers

code

triple	0x406a2
third	0x406b4

main

argc	1
argv	0xbf825
array	0x7ffdb
0x7ffdb	3
0x7ffe0	6

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C                               1,1                               All

```

Function pointers

code

triple	0x406a2
--------	---------

third	0x406b4
-------	---------

printArray

array	0x7ffdb
-------	---------

length	2
--------	---

i	0
---	---

0x	
----	--

0x/neu	0
--------	---

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C          1,1          All

```

Function pointers

code

triple	0x406a2
third	0x406b4

main

argc	1
argv	0xbf825
array	0x7ffdb
0x7ffdb	3
0x7ffe0	6

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}
"fptr.c" 26L, 481C                                1,1                                All

```

Function pointers

code	
triple	0x406a2
third	0x406b4

map	
func	0x406b4
array	0x7ffdb
length	2
i	0
0x7ffe0	6

```
elinux.cs.umass.edu - PuTTY
```

```
#include<stdio.h>
```

```
void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}
```

```
void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}
```

```
int triple(int a){ return a*3; }
int third (int a){ return a/3; }
```

```
int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}
```

```
"fptr.c" 26L, 481C
```

```
1,1
```

```
All
```

Function pointers

code	
triple	0x406a2
third	0x406b4

map	
func	0x406b4
array	0x7ffdb
length	2
i	0
0x7ffe0	6

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C                                1,1                                All

```

Function pointers

code	
triple	0x406a2
third	0x406b4

map	

main	
argc	1
argv	0xbf825
array	0x7ffdb
0x7ffdb	3
0x7ffe0	6

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C                                1,1                                All

```

Function pointers

code	
triple	0x406a2
third	0x406b4

third	
a	3
0x7ffe0	6

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C                                1,1                                All

```


Function pointers

code	
triple	0x406a2
third	0x406b4

map	
argc	1
argv	0xbf825
array	0x7ffdb
0x7ffdb	1
0x7ffe0	6

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C      1,1      All

```

Function pointers

code	
triple	0x406a2
third	0x406b4

map	
func	0x406b4
array	0x7ffdb
length	2
i	1
0x7ffe0	6

```
elnux.cs.umass.edu - PuTTY
```

```
#include<stdio.h>
```

```
void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}
```

```
void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}
```

```
int triple(int a){ return a*3; }
```

```
int third (int a){ return a/3; }
```

```
int main(int argc, char * argv[]){
```

```
    int array[2]={1,2};
```

```
    map(triple,array,2);
```

```
    printArray(array,2);
```

```
    map(third,array,2);
```

```
    printArray(array,2);
```

```
    return 0;
```

```
}
```

```
"fptr.c" 26L, 481C
```

```
1,1
```

```
All
```

Function pointers

code	
triple	0x406a2
third	0x406b4

map	

main	
argc	1
argv	0xbf825
array	0x7ffdb
0x7ffdb	1
0x7ffe0	6

```
elinux.cs.umass.edu - PuTTY
```

```
#include<stdio.h>
```

```
void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}
```

```
void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}
```

```
int triple(int a){ return a*3; }
```

```
int third (int a){ return a/3; }
```

```
int main(int argc, char * argv[]){
```

```
    int array[2]={1,2};
```

```
    map(triple,array,2);
```

```
    printArray(array,2);
```

```
    map(third,array,2);
```

```
    printArray(array,2);
```

```
    return 0;
```

```
}
```

```
"fptr.c" 26L, 481C
```

```
1,1
```

```
All
```

Function pointers

code	
triple	0x406a2
third	0x406b4

third	
a	6
f	
a	a
a	l
a	i
0	
0x7ffe0	6

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}
"fptr.c" 26L, 481C                               1,1                               All

```

Function pointers

code	
triple	0x406a2
third	0x406b4

map	
argc	1
argv	0xbf825
array	0x7ffdb
0x7ffdb	1
0x7ffe0	2

```
elinux.cs.umass.edu - PuTTY
```

```
#include<stdio.h>
```

```
void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}
```

```
void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}
```

```
int triple(int a){ return a*3; }
```

```
int third (int a){ return a/3; }
```

```
int main(int argc, char * argv[]){
```

```
    int array[2]={1,2};
```

```
    map(triple,array,2);
```

```
    printArray(array,2);
```

```
    map(third,array,2);
```

```
    printArray(array,2);
```

```
    return 0;
```

```
}
```

```
"fptr.c" 26L, 481C
```

```
1,1
```

```
All
```

Function pointers

code	
triple	0x406a2
third	0x406b4

map	
func	0x406b4
a array	0x7ffdb
a length	2
a i	2
0	
0x7ffe0	2

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}

"fptr.c" 26L, 481C                                1,1                                All

```

Function pointers

code

triple	0x406a2
third	0x406b4

main

argc	1
argv	0xbf825
array	0x7ffdb
0x7ffdb	1
0x7ffe0	2

```

elinux.cs.umass.edu - PuTTY
#include<stdio.h>

void map(int (*func)(int), int * array, int length){
    int i=0;
    for(i=0;i<length;i++){
        array[i]= func(array[i]);
    }
}

void printArray(int * array, int length){
    int i=0;
    for(i=0;i<length;i++){printf("%i ",array[i]);}
    printf("\n");
}

int triple(int a){ return a*3; }
int third (int a){ return a/3; }

int main(int argc, char * argv[]){
    int array[2]={1,2};
    map(triple,array,2);
    printArray(array,2);
    map(third,array,2);
    printArray(array,2);
    return 0;
}
"fptr.c" 26L, 481C          1,1          All

```

Your Assignment

```
XXXXXXXXXX
X  *  XX
X X^XX XX
XPX :X XX
X\XXX: XX
XX/  *X D
XXX^XXX*X
XXX^XX  X
X/      XX
XXXXXXXXXX
```

```
XXXXXXXXXX
X   ^  XX
X X*XX XX
X X :X XX
XPXXX: XX
XX/   ^X D
XXX*XXX^X
XXX*XX  X
X/      XX
XXXXXXXXXX
```

```
XXXXXXXXXX
X   ^  XX
X X*XX XX
X XP:X XX
X/XXX: XX
XX/   ^X D
XXX*XXX^X
XXX*XX  X
X/      XX
XXXXXXXXXX
```

```
XXXXXXXXXX
X   ^  XX
X X*XX XX
X X PX XX
X/XXX: XX
XX/   ^X I
XXX*XXX^X
XXX*XX  X
X/      XX
XXXXXXXXXX
```

```
XXXXXXXXXX
X  P^  XX
X X*XX XX
X X .X XX
X/XXX: XX
XX/   ^X I
XXX*XXX^X
XXX*XX  X
X/      XX
XXXXXXXXXX
```

```
XXXXXXXXXX
X  P  XX
X X*XX XX
X X .X XX
X/XXX: XX
XX/   ^X I
XXX*XXX^X
XXX*XX  X
X/      XX
XXXXXXXXXX
You fell onto the spikes and died.
```


A photograph of a UMass Amherst campus scene, featuring a large tree on the left, a modern building in the background, and a field of tulips in the foreground. The image is overlaid with a semi-transparent dark red filter. The text is centered in the middle of the image.

UMassAmherst
The Commonwealth's Flagship Campus