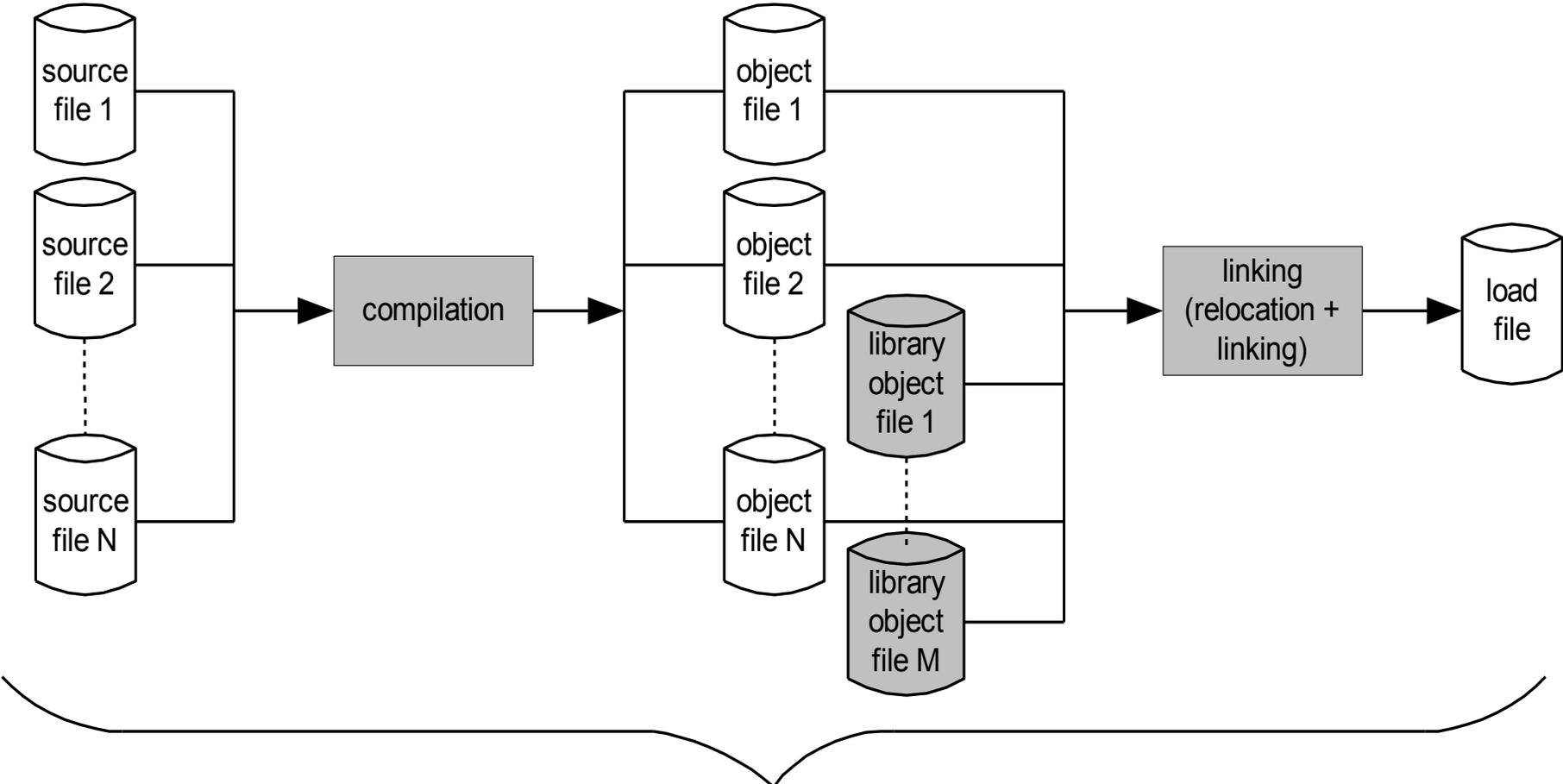
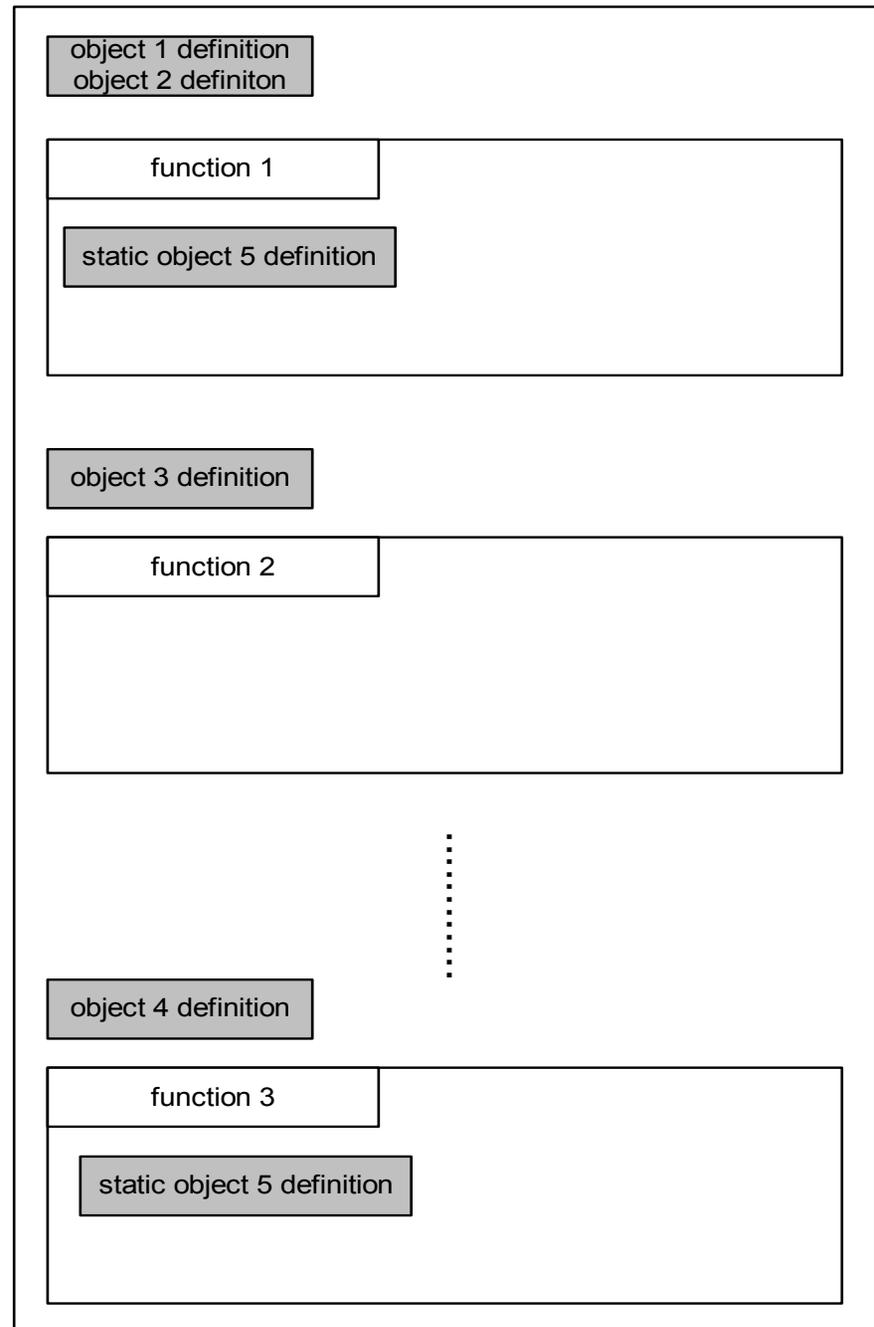


Compilation, Linking, Execution of C/C++ programs: compilation and linking

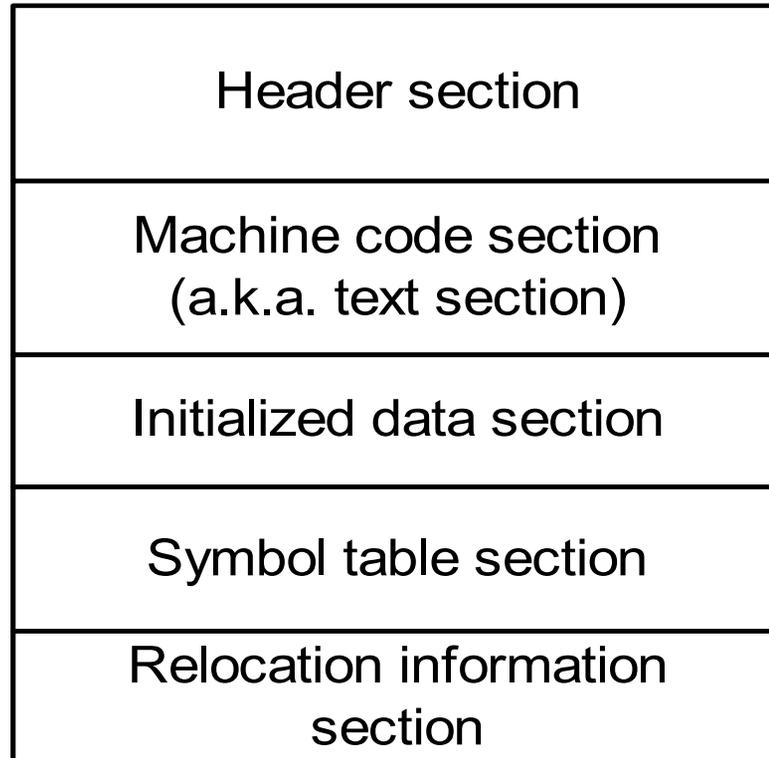


usually performed by a compiler, usually in one uninterrupted sequence

Layout of C/C++ programs:



Object module structure:



A sample C program:

```
#include <stdio.h>

int a[10]={0,1,2,3,4,5,6,7,8,9};
int b[10];

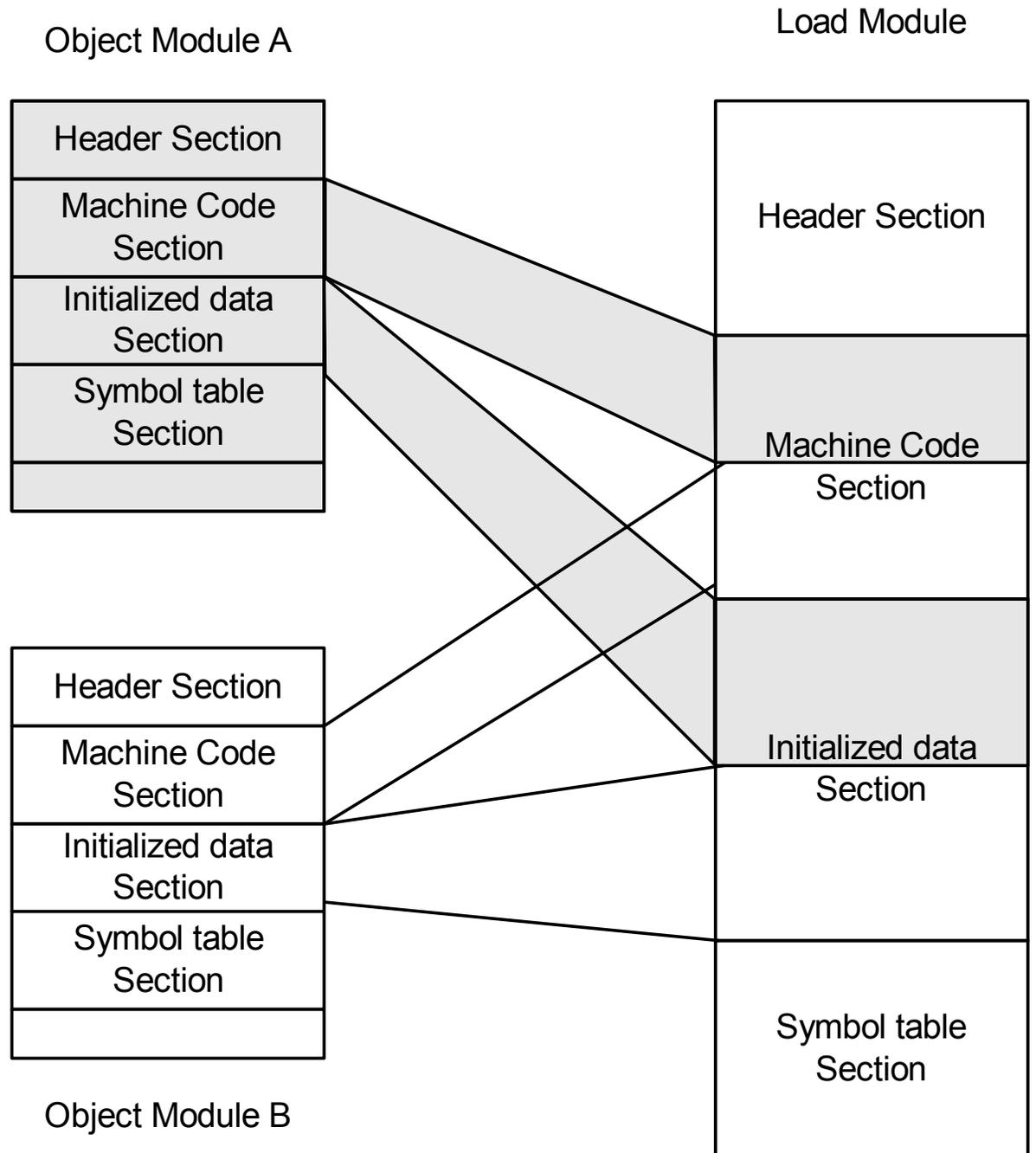
void main()
{
    int i;
    static int k = 3;

    for(i = 0; i < 10; i++) {
        printf("%d\n",a[i]);
        b[i] = k*a[i];
    }
}
```

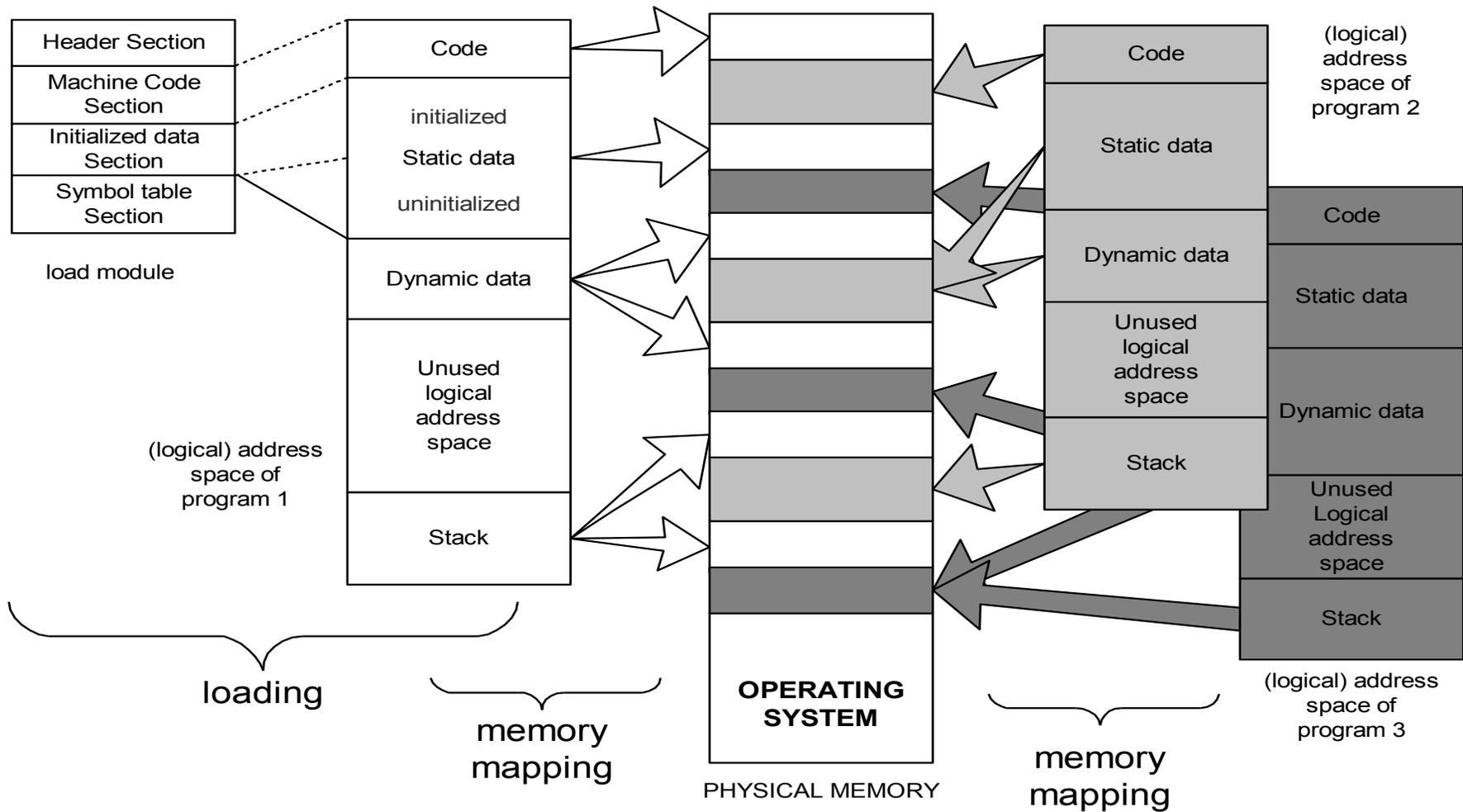
Object module of the sample C program:

<i>Offset</i>	<i>Contents</i>	<i>Comment</i>
Header section		
0	124	number of bytes of Machine code section
4	44	number of bytes of initialized data section
8	40	number of bytes of Uninitialized data section (array <code>b []</code>) (<i>not part of this object module</i>)
12	60	number of bytes of Symbol table section
16	44	number of bytes of Relocation information section
Machine code section (124 bytes)		
20	X	code for the top of the <code>for</code> loop (36 bytes)
56	X	code for call to <code>printf()</code> (22 bytes)
68	X	code for the assignment statement (10 bytes)
88	X	code for the bottom of the <code>for</code> loop (4 bytes)
92	X	code for exiting <code>main()</code> (52 bytes)
Initialized data section (44 bytes)		
144	0	beginning of array <code>a []</code>
148	1	
:		
176	8	
180	9	end of array <code>a []</code> (40 bytes)
184	3	variable <code>k</code> (4 bytes)
Symbol table section (60 bytes)		
188	X	array <code>a []</code> : offset 0 in Initialized data section (12 bytes)
200	X	variable <code>k</code> : offset 40 in Initialized data section (10 bytes)
210	X	array <code>b []</code> : offset 0 in Uninitialized data section (12 bytes)
222	X	<code>main</code> : offset 0 in Machine code section (12 bytes)
234	X	<code>printf</code> : external, used at offset 56 of Machine code section (14 bytes)
Relocation information section (44 bytes)		
248	X	relocation information

Creation of load module

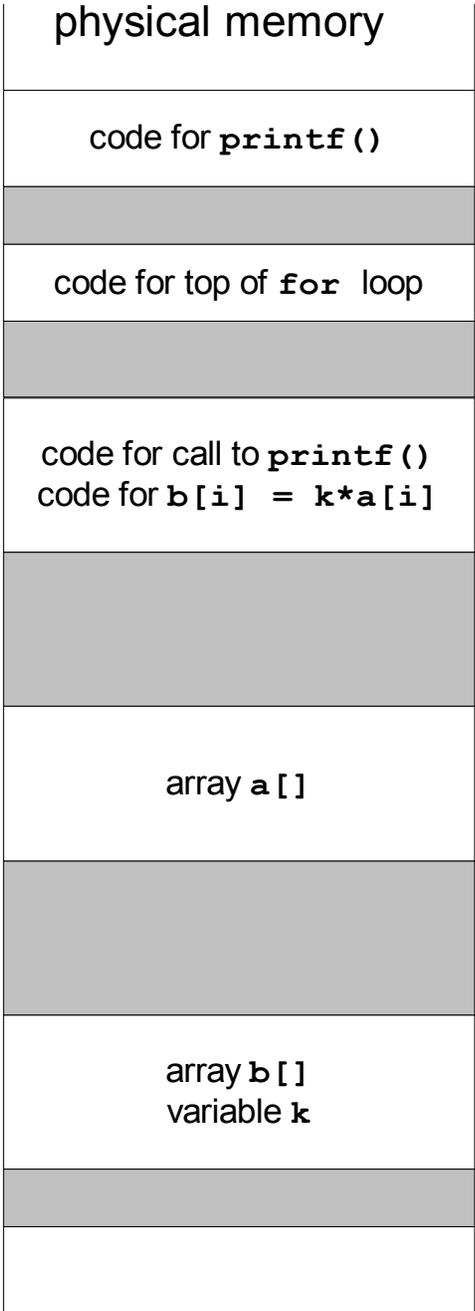
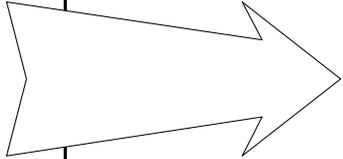


Loading and memory mapping:

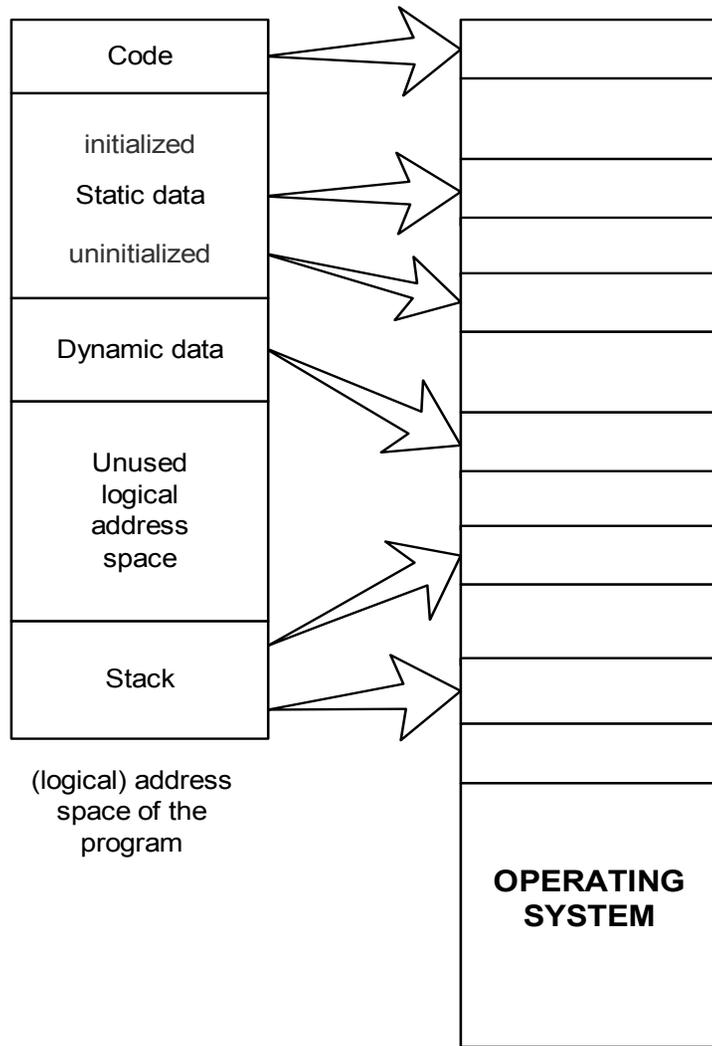


From source program
to “placement” in
memory during source program
execution

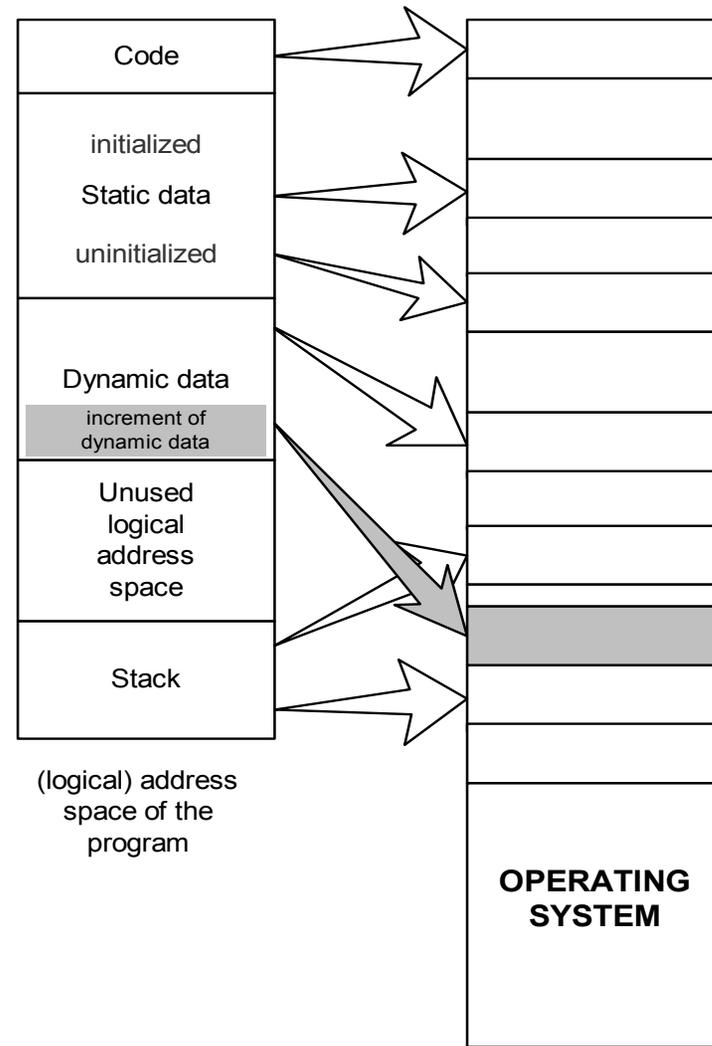
```
int a[10]={0,1,2,3,4,5,6,7,8,9};  
int b[10];  
  
void main()  
{  
  int i;  
  static int k = 3;  
  
  for(i = 0; i < 10; i++) {  
    printf("%d\n",a[i]);  
    b[i] = k*a[i];  
  }/*endfor*/  
}/*end main*/
```



Dynamic memory allocation:



Before dynamic memory allocation



After dynamic memory allocation