OSTEP Chapter 13

ECE 3600, Fall 2022

Table of Contents

1. Address Spaces

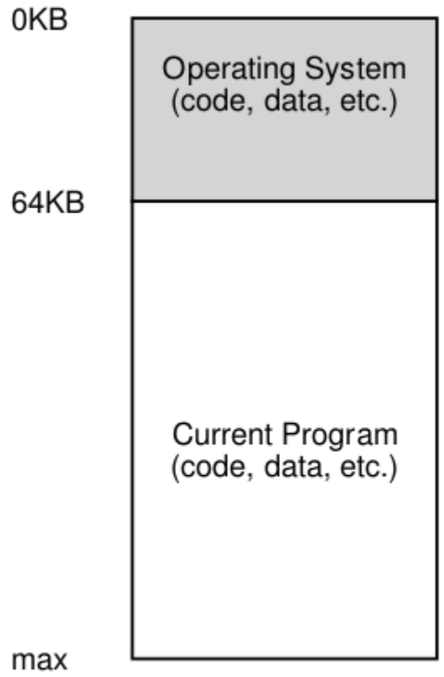

Figure 13.1: **Operating Systems: The Early Days**

2. Sharing Memory

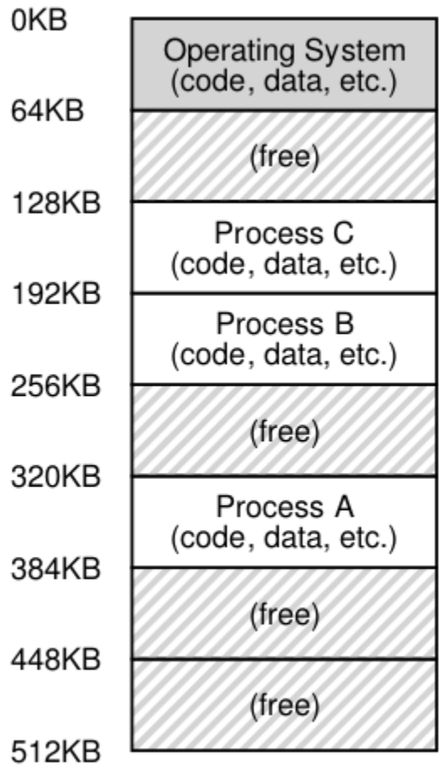

Figure 13.2: **Three Processes: Sharing Memory**

3. Code, Heap, Stack

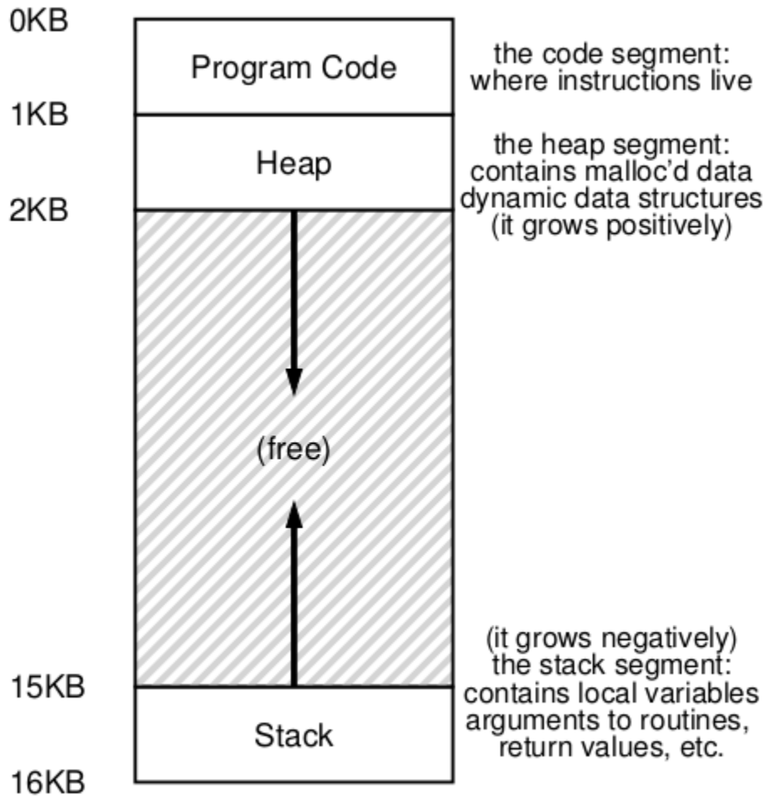

Figure 13.3: **An Example Address Space**

4. Experiment

va.c

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

int main(int argc, char *argv[]) {
    printf("location of code : %p\n", main);
    printf("location of heap : %p\n", malloc(100e6));
    int x = 3;
    printf("location of stack: %p\n", &x);
    return 0;
}
```

Sample output:

```
location of code : 0x55aa16a896fa
location of heap : 0x7f9c83599010
location of stack: 0x7ffddc28c204
```

note: stack/1e9 = 140728, i.e. 140 TB

try: static data

try commands: free, ps, pmap, top, lscpu; also see: /proc/<pid>/maps, /proc/cpuinfo

Homework Q3: Create a little program that uses a certain amount of memory, called memory-user.c. This program should take one command-line argument: the number of megabytes of memory it will use. When run, it should allocate an array, and constantly stream through the array, touching each entry. The program should do this indefinitely, or, perhaps, for a certain amount of time also specified at the command line.

alternative: see malloc() example