Chapter 13: Address Spaces

Adam Disney

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Early Systems

- OS was just a library of functions.
- A single program used the rest of memory.

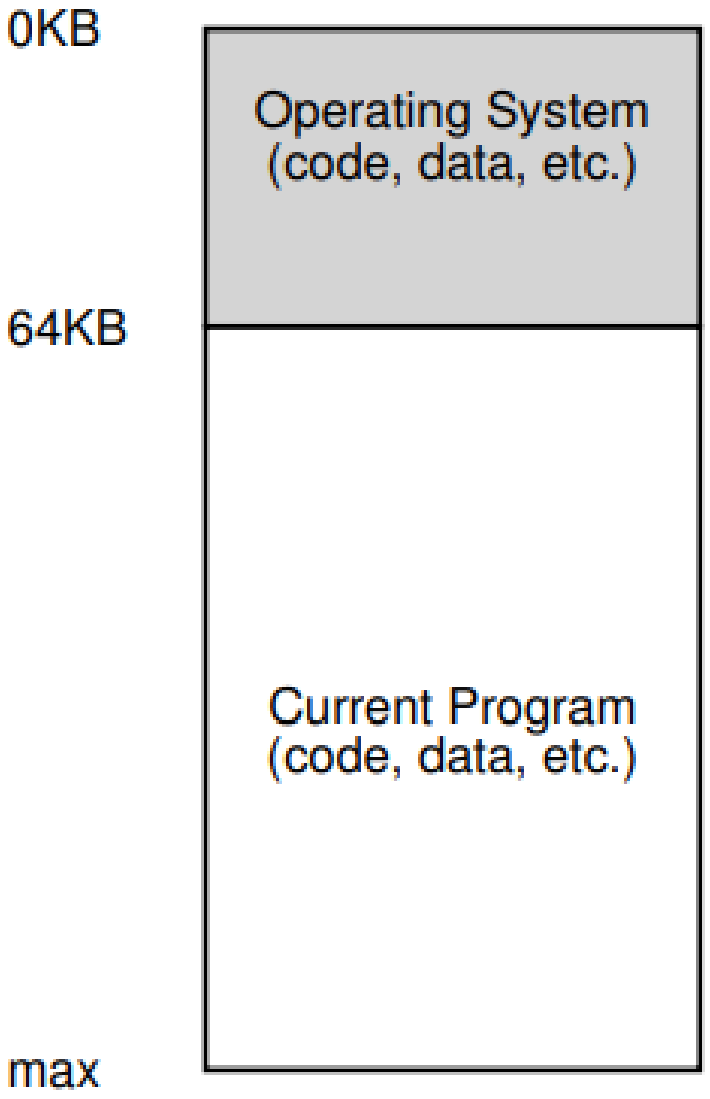

Figure 13.1: **Operating Systems: The Early Days**

Multiprogramming and Time Sharing

- Multiprogramming:
  - Multiple processes were ready to run, and we wanted to maximize CPU usage because machines were very expensive.
- Time Sharing:
  - Realized limitations of batch computing, particularly on programmers.
  - Interactivity became important
- https://history-computer.com/dec-pdp-11-computer/
- https://en.wikipedia.org/wiki/PDP-11

Time Sharing

- One way to implement the context switch (given the model from Figure 13.1) is to save/restore all state, including all physical memory, to/from disk.
- This is brutally slow especially as memory grows, thus we want to leave processes in memory and only save/load the register-level state which is fast.

Time Sharing

- One way to implement the context switch (given the model from Figure 13.1) is to save/restore all state, including all physical memory, to/from disk.
- This is brutally slow especially as memory grows, thus we want to leave processes in memory and only save/load the register-level state which is fast.
- With multiple programs in memory, protection becomes an important issue.

Time Sharing

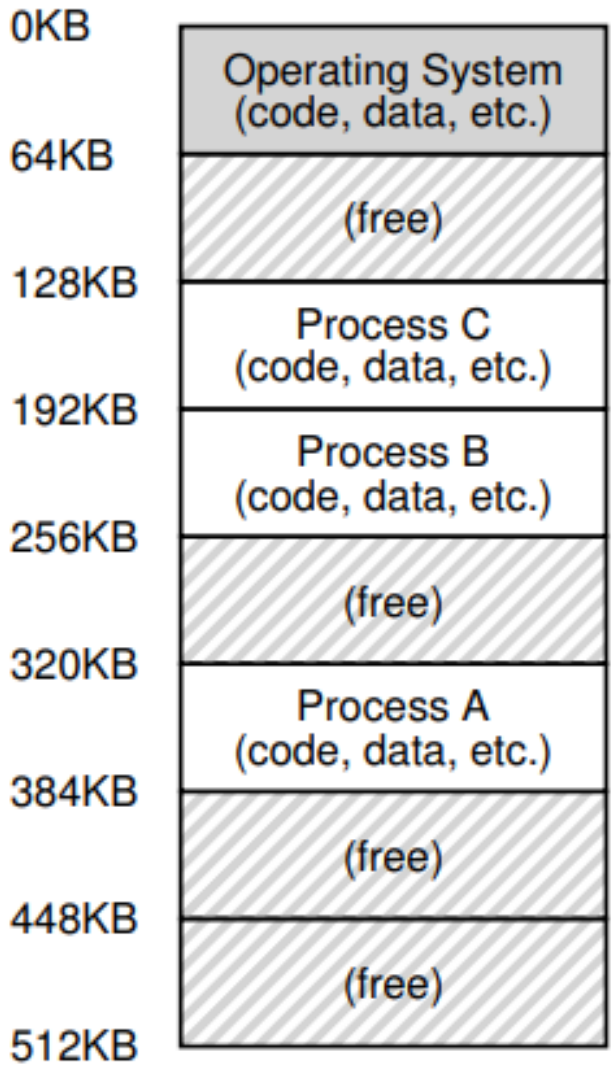

Figure 13.2: **Three Processes: Sharing Memory**

The Address Space

- We need an easy to use abstraction of physical memory called an address space.
- The address space of a process contains all the memory state.
  - Code, stack, heap, others we're going to ignore for now

The Address Space

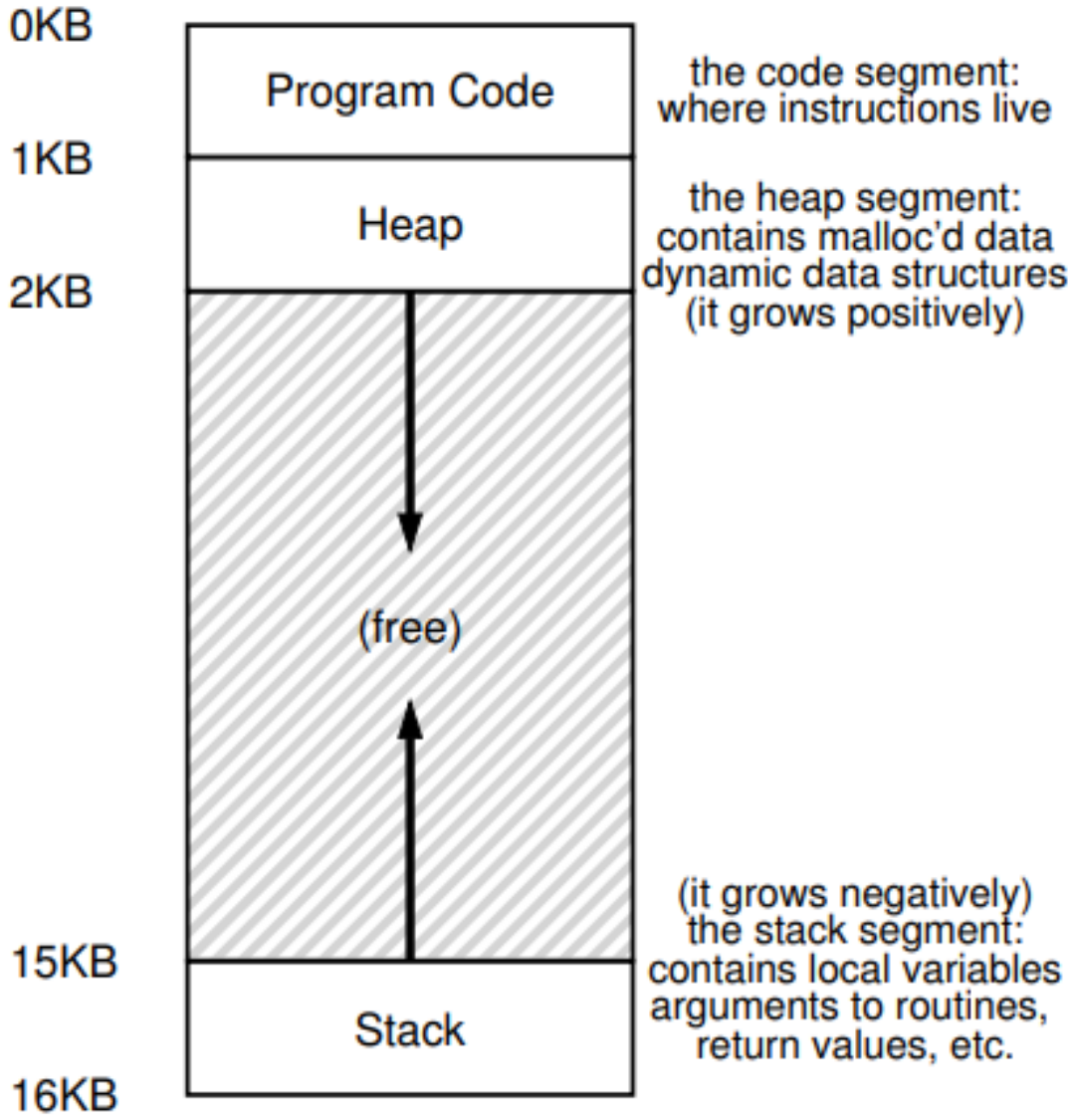

Figure 13.3: **An Example Address Space**

The Address Space

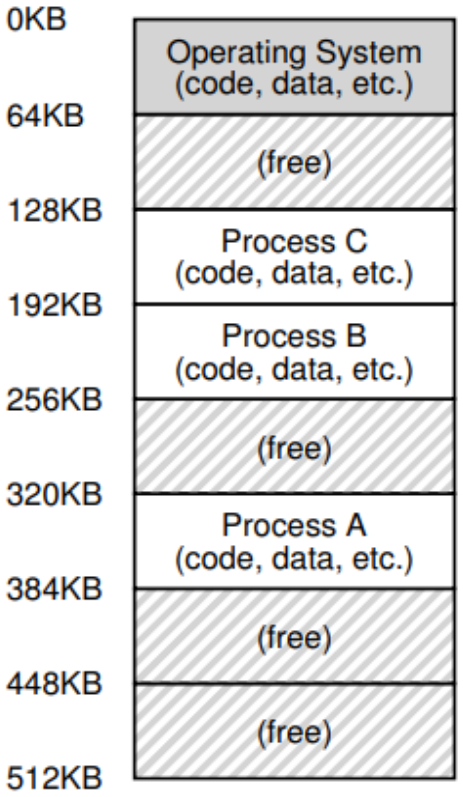

Figure 13.2: **Three Processes: Sharing Memory**

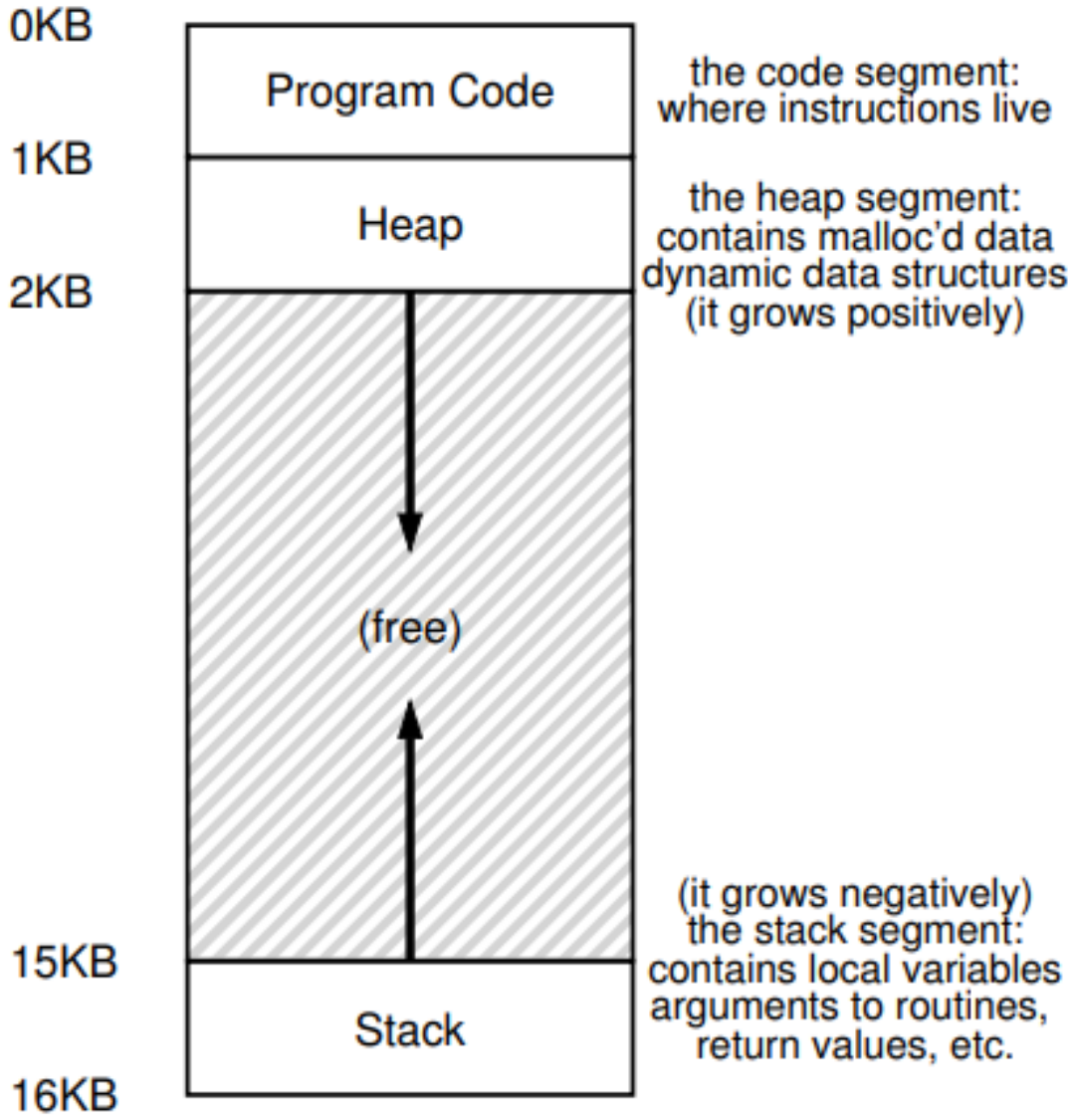

Figure 13.3: **An Example Address Space**

Virtual Memory Goals

- Transparency – Programs behave as if they own the physical memory.
- Efficiency - Time & Space
- Protection - Processes should not be able to affect the OS or other processes

THE CRUX: HOW TO VIRTUALIZE MEMORY

How can the OS build this abstraction of a private, potentially large address space for multiple running processes (all sharing memory) on top of a single, physical memory?

Next Time...

- Mechanisms – Hardware and OS support
- Policies – How to manage free space and paging