computer science illuminated

Memory Management

Nell Dale & John Lewis

(adaptation by Michael Goldwasser)

Operating System

- One of the goals of an operating system is to allow many program to execute simultaneously.
- A given program, however, is generally written as if it is the only one!
- Machine code for a program relies on memory addresses for instruction branching and for data structures

Memory Management

- Operating systems must employ techniques to:
  - Track where and how a program resides in memory
  - Convert “logical” program addresses into actual memory addresses

Memory Management

- A **logical address** (sometimes called a virtual or relative address) is a value that specifies a generic location, relative to the program but not to the reality of main memory
- A **physical address** is an actual address in the main memory device

Memory Management

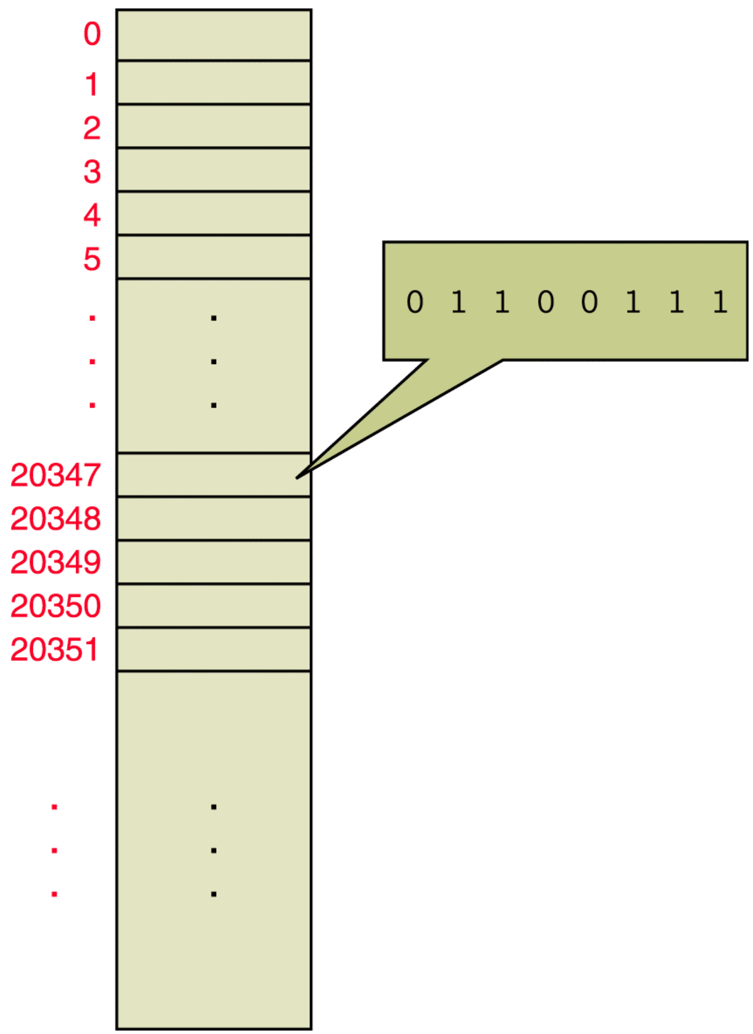


Figure 10.3
Memory is a continuous set of bits referenced by specific addresses

Single Contiguous Memory Management

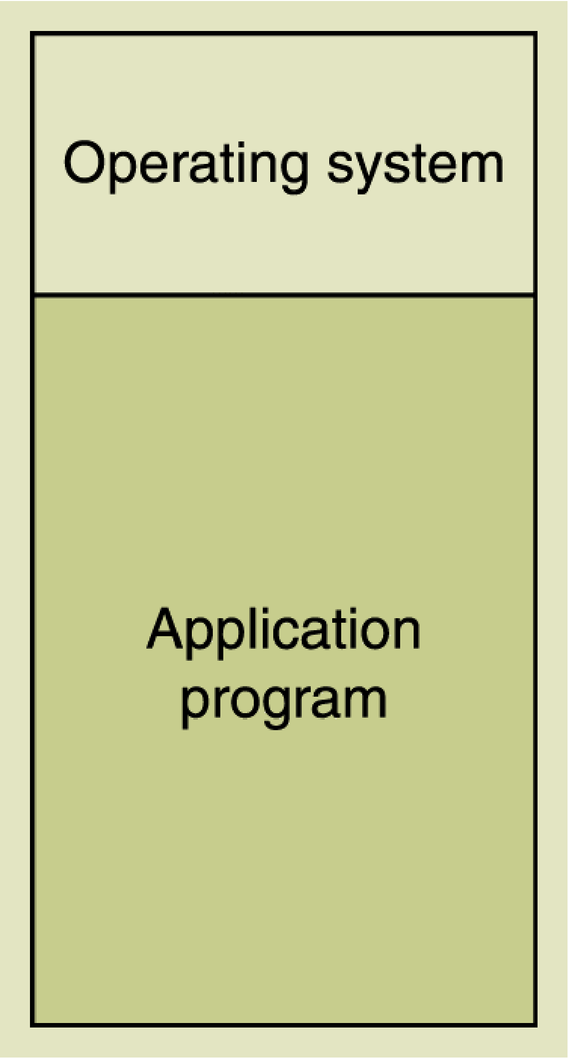


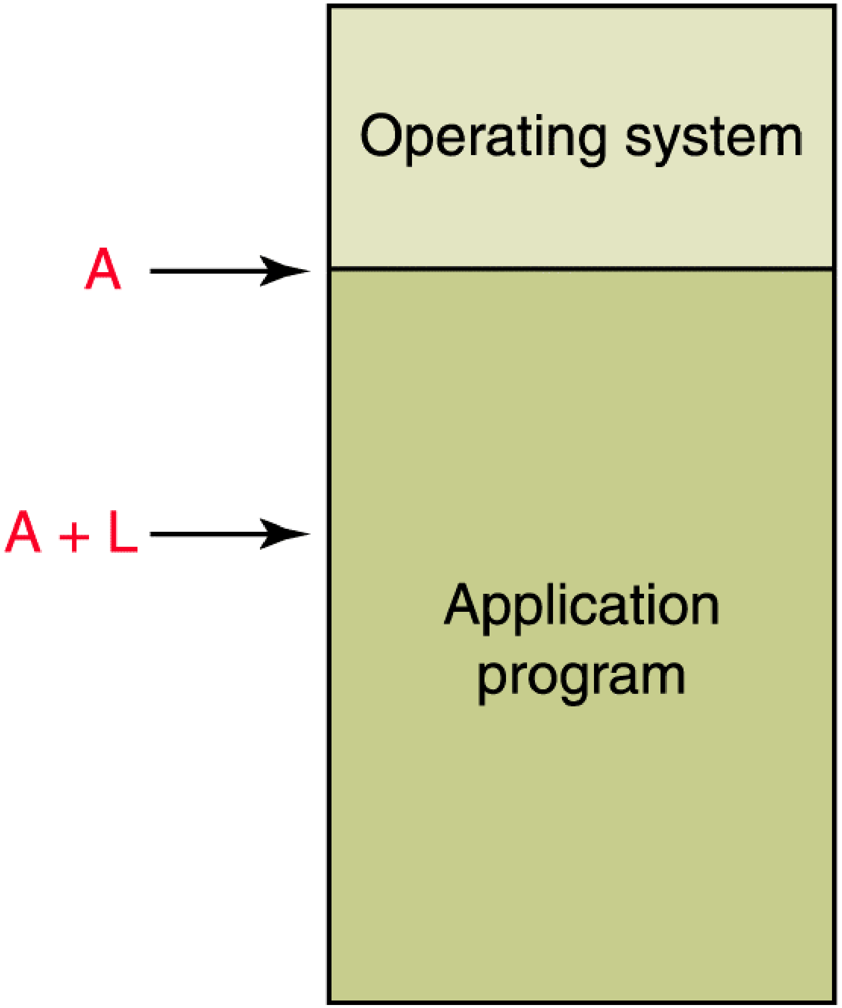
Figure 10.4
Main memory divided into two sections

- There are only two programs in memory
  - The operating system
  - The application program
- This approach is called **single contiguous memory management**

Single Contiguous Memory Management

- A logical address is simply an integer value relative to the starting point of the program
- To produce a physical address, we add a logical address to the starting address of the program in physical main memory

Single Contiguous Memory Management



Logical address L

translates to

Physical address A + L

Figure 10.5
binding a logical address to a physical one

Partition Memory Management

- Rather than OS and one other program, we consider OS and many programs.
- Each program can be given its own **partition** of the main memory.
- Two approaches taken by operating systems:
  - **fixed partitions**: memory is divided into many partitions which may be unequal, but which are fixed at the time to operating system boots.
  - **dynamic partitions:** the partitions are created to fit the specific need of each program.

Partition Memory Management

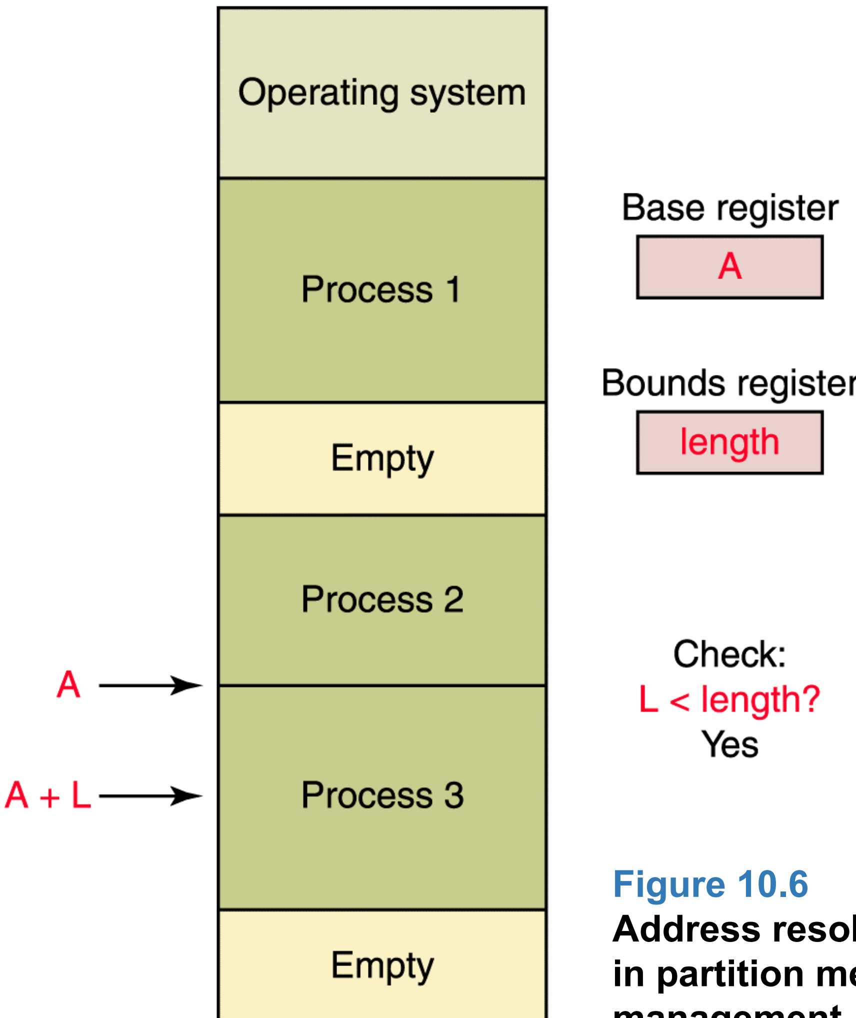


Figure 10.6
Address resolution in partition memory management

- At any point in time memory is divided into a set of partitions, some empty and some allocated to programs
- **Base register:** a register that holds the beginning address of the current partition
- **Bounds register:** a register that holds the length of the current partition

Partition Selection

Which partition should we allocate to a new program?

- *First fit*
  - the **first** partition big enough to hold it
- *Best fit*
  - the **smallest** partition big enough to hold it
- *Worst fit*
  - The **largest** partition big enough to hold it

Partition Memory Management

- **Advantage**:

  This scheme is fairly easy to implement.

- **Disadvantage**:

  Memory for an individual process must fit in a single partition to be contiguous. Might be that no single partition is large enough for a new program, even though enough free memory exists. (Dynamic partitioning partially avoids this)

Paged Memory Management

- **Paged memory technique:**

  Instead of dividing main memory into partitions, divide it into much smaller, fixed-size blocks of storage called **frames**.

- **Each process**:

  Memory is divided into **pages** (for the sake of our discussion, assume that pages are the same size as the frames)

Paged Memory Management

- The **Operating System** can assign memory in a way that the pages used by an individual process need not be contiguous!
- The operating system maintains a separate **page-map table** (PMT) for each process in memory

Paged Memory Management

P1 PMT

Page	Frame
0	5
1	12
2	15
3	7
4	22

P2 PMT

Page	Frame
0	10
1	18
2	1
3	11

Memory

Frame	Contents
0	
1	P2/Page2
2	
3	
4	
5	P1/Page0
6	
7	P1/Page3
8	
9	
10	P2/Page0
11	P2/Page3
12	P1/Page1
13	
14	
15	P1/Page2

.
.
.

Figure 10.7
A paged memory management approach

- To produce a physical address, you first look up the page in the PMT to find the frame number in which it is stored
- Then multiply the frame number by the frame size and add the offset to get the physical address

Demand paging

- An important extension is **demand paging**
  - not all parts of a program actually have to be in memory at the same time (the rest can be stored in secondary memory)
  - In demand paging, the pages are brought into memory on demand
  - The act of bringing in a page from secondary memory, which often causes another page (which one?) to be written back to secondary memory, is called a **page swap**

Virtual Memory

- The demand paging approach gives rise to the idea of **virtual memory,** the illusion that there are no restrictions on the size of a program
- Too much page swapping, however, is called **thrashing** and can seriously degrade system performance.