CIS 452 - Operating Systems Concepts

Nathan Bowman

Images taken from Silberschatz book

Introduction to Memory

OS is a resource manager

Probably the most important resource is the CPU

Previously studied how processes are managed and scheduled

Next resource we will investigate is **memory**

Terminology: for our purposes,

memory == main memory == RAM

Any code we wish to run or operand our program uses must be in memory

We have previously discussed importance of *multiprogrammming* -- keeping more than one process in memory

This is critical for improving system resource utilization and decreasing response time for users

Two major complications arise when dealing with memory:

- must protect memory from unauthorized processes
- want to make access simple for programmers

Techniques to acheive this will typically rely on hardware support

Fundamentally, memory is a very long array of bytes

Each byte has its own address

Both data and instructions are stored in this large array

Most of what a program does is load, modify, and store memory

```
a = a + 5;
```

We are not going to concern ourselves with caches

They are extremely important for performance, but they are mostly run by the hardware itself, not the OS

Two major complications arise when dealing with memory:

- **must protect memory from unauthorized processes**
- want to make access simple for programmers

OS does not have time to monitor every single memory access of every single process on the system

We will look at a very simple way this can be implemented with the help of hardware

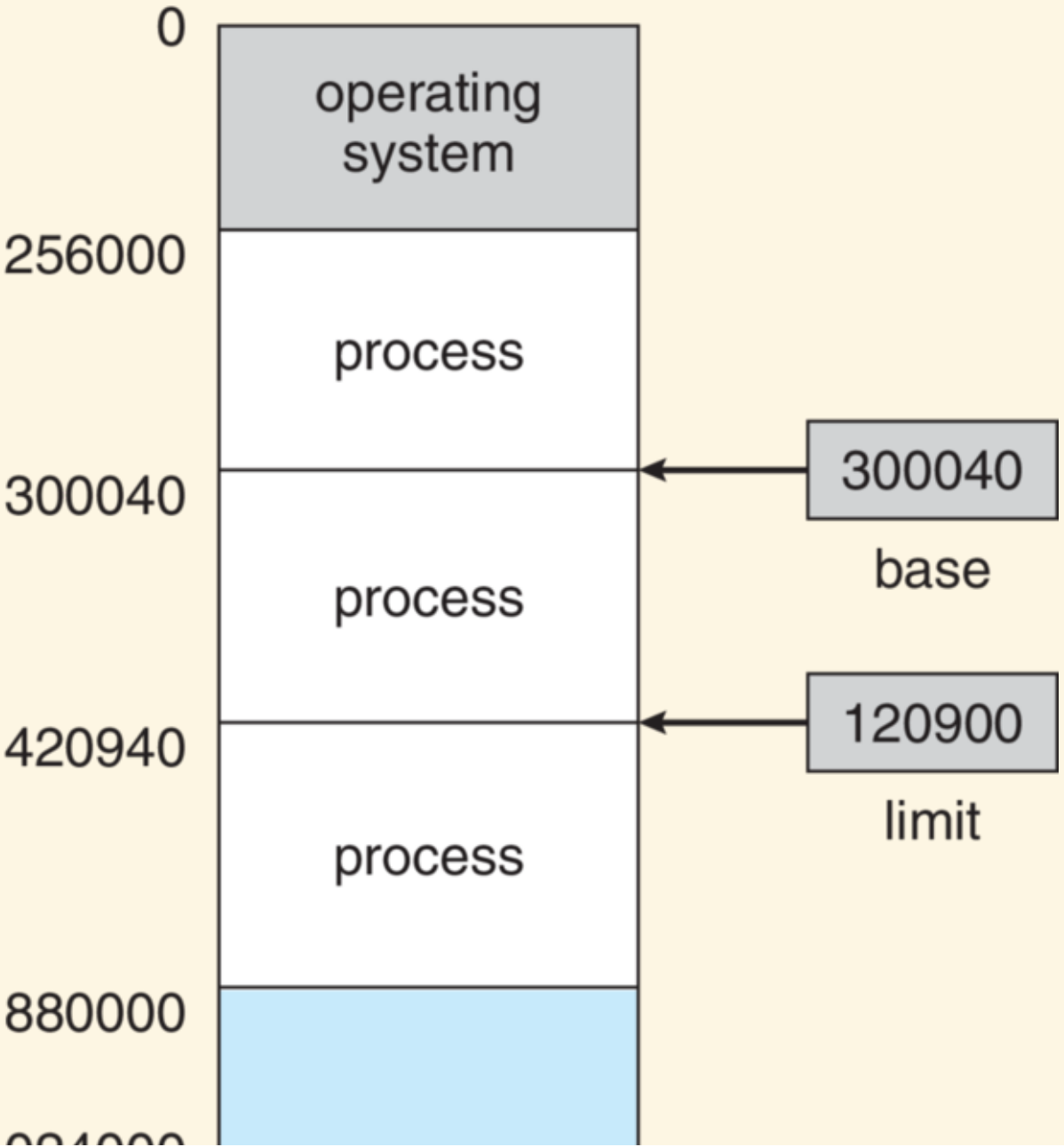
Each process is assigned a separate space in memory

Processes may access only their own memory space

Must determine *range of legal addresses* for a process to access

Use registers: **base register** and **limit register**

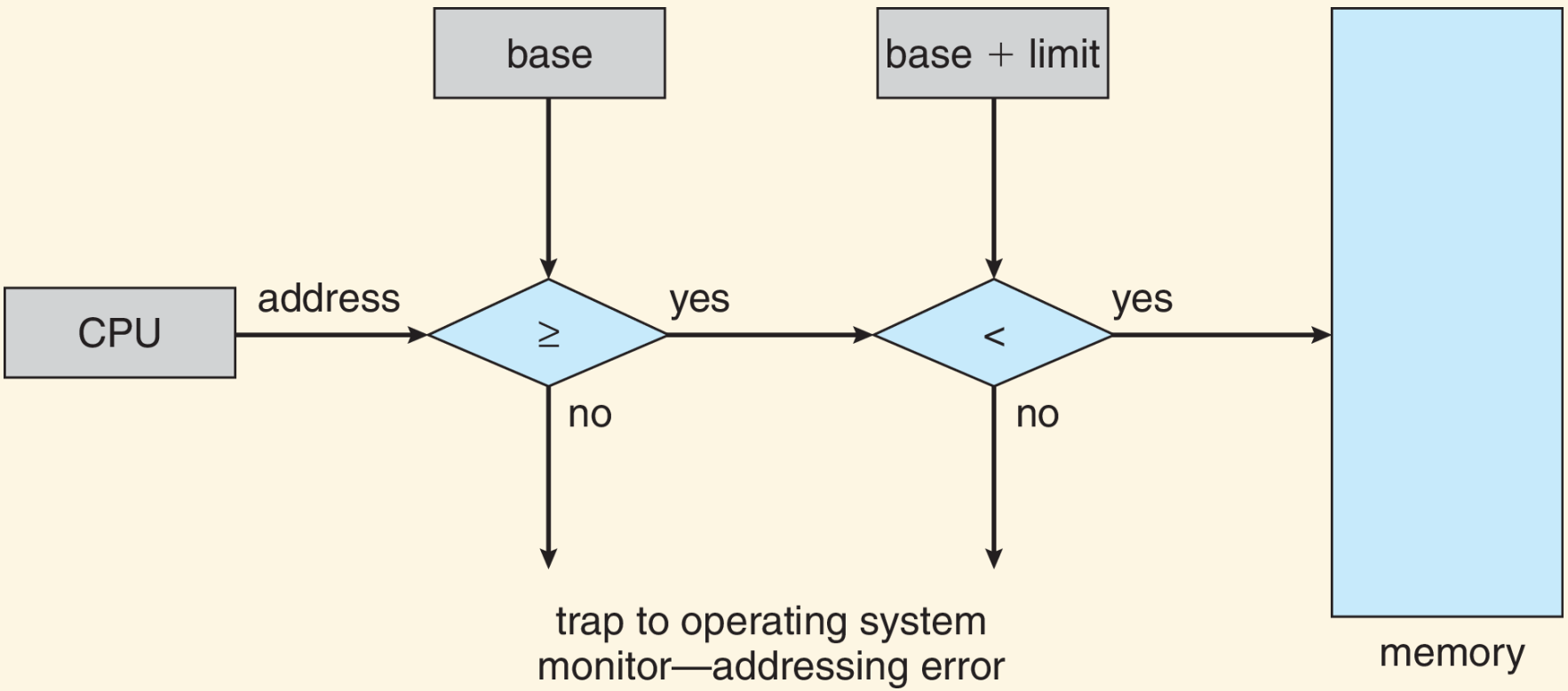
If b is value in base register and v is value in limit register, current process can access memory locations `[b, b + v]`


0
operating system
256000
process
300040
process
420940
process
880000
300040
base
120900
limit


1024000

Memory accesses checked only when in user mode -- OS can access whatever memory it needs

Changing `base` and `limit` registers must be a kernel-mode instruction


base
base + limit
CPU
address
≥
yes
<
yes
no
no
trap to operating system
monitor—addressing error
memory