

Assignment Project Exam Help

Add WeChat powcoder

CS:3620 Operating Systems

Assignment Project Exam Help

<https://powcoder.com>

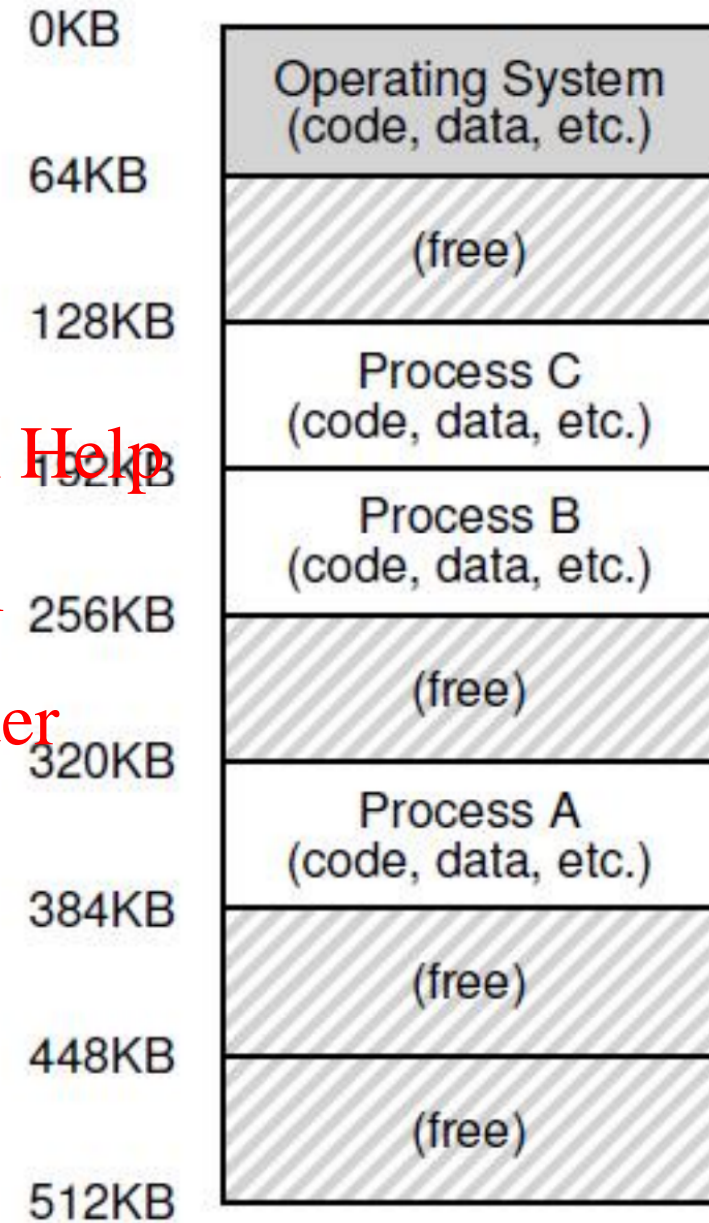
Add WeChat powcoder

Introduction to Virtual Memory

Assignment Project Exam Help

Why virtualize memory?

- Because real view of memory is messy!
- Earlier, memory had only code of one running process (and OS code)
- Now, multiple active processes timeshare CPU
 - Memory of many processes must be in memory
 - Non-contiguous too
- Need to hide this complexity from user



Assignment Project Exam Help

Abstraction: (Virtual) Address Space

- Virtual address space: every process assumes it has access to a large space of memory from address 0 to a MAX
- Contains program code (and static data), heap (dynamic allocations), and stack (used during function calls)
 - Stack and heap grow during runtime
- CPU issues loads and stores to virtual addresses

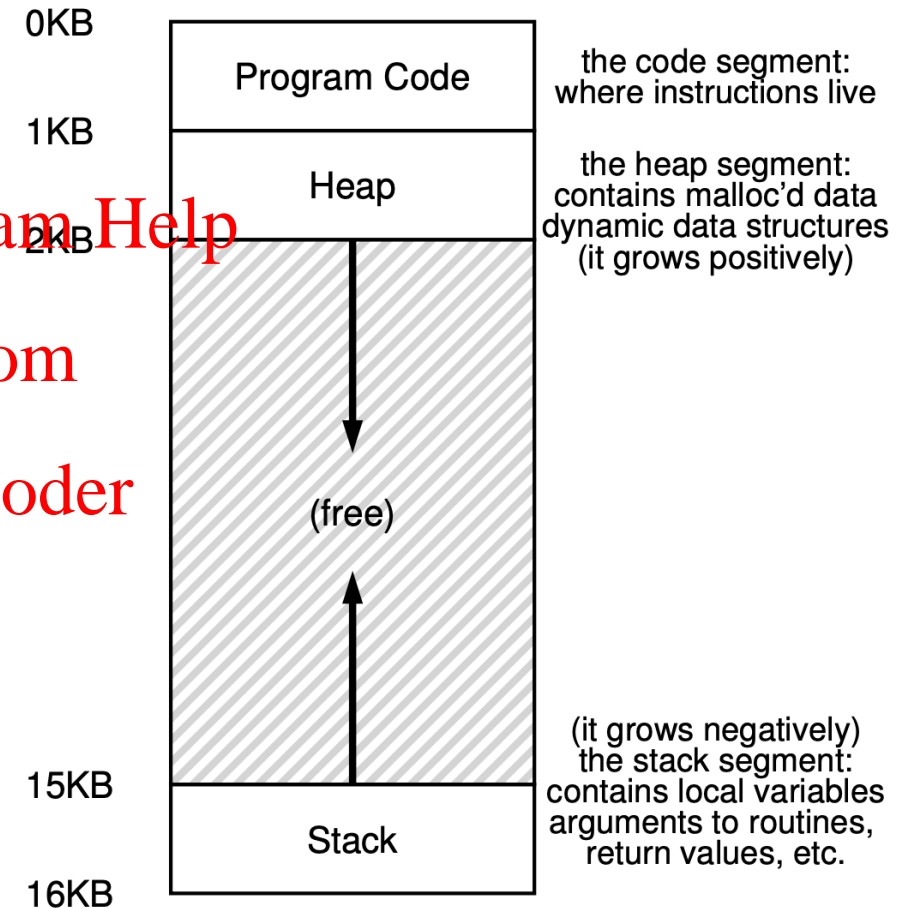


Figure 13.3: An Example Address Space

Assignment Project Exam Help

How is actual memory reached?

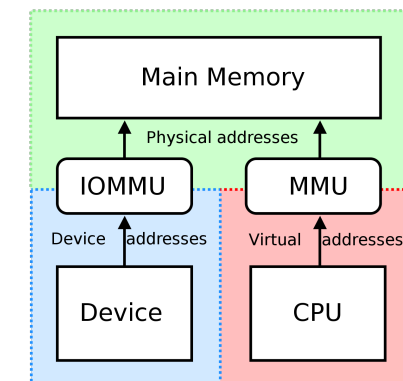
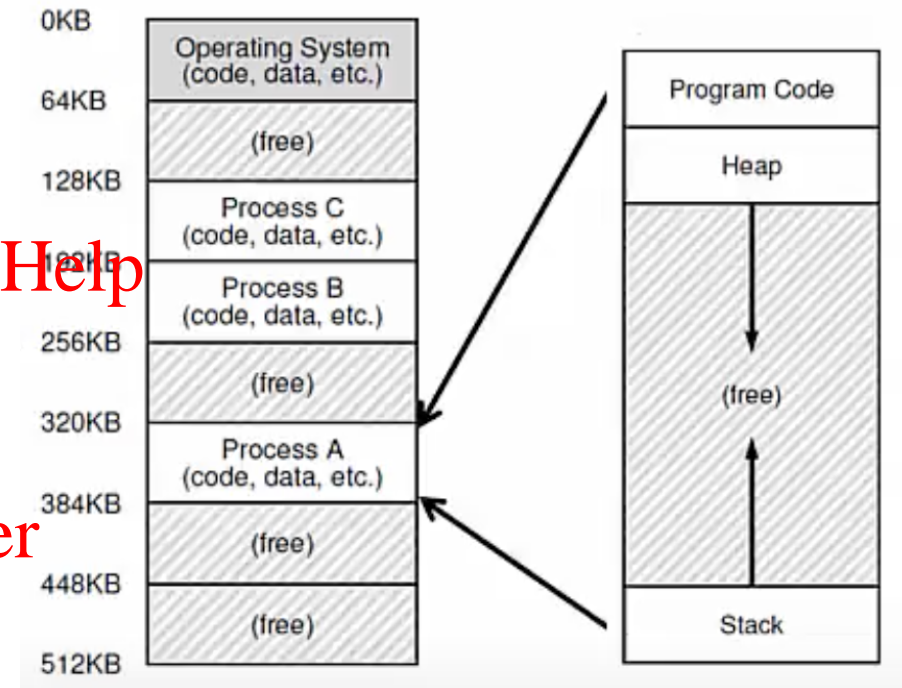
- Address translation from virtual addresses (VA) to physical addresses (PA)

- CPU issues loads/stores to VA but memory hardware accesses PA

- OS allocates memory and tracks location of processes

- Translation done by memory hardware called Memory Management Unit (MMU)

- OS makes the necessary information available

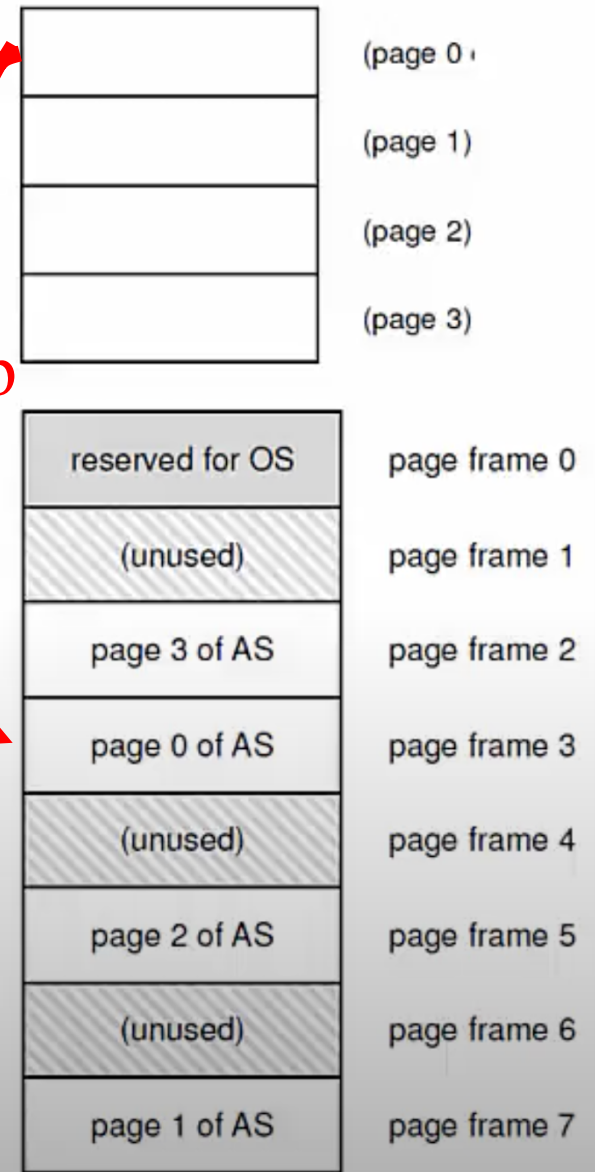


Assignment Project Exam Help

Example: Paging

- OS divides virtual address space into fixed size pages, physical memory into frames
- To allocate memory, a page is mapped to a free physical frame
- Page table stores mappings from virtual page number to physical frame number for a process (e.g, page 0 to frame 3)
- MMU has access to page tables, and uses it to translate VA to PA

Virtual memory space of a process

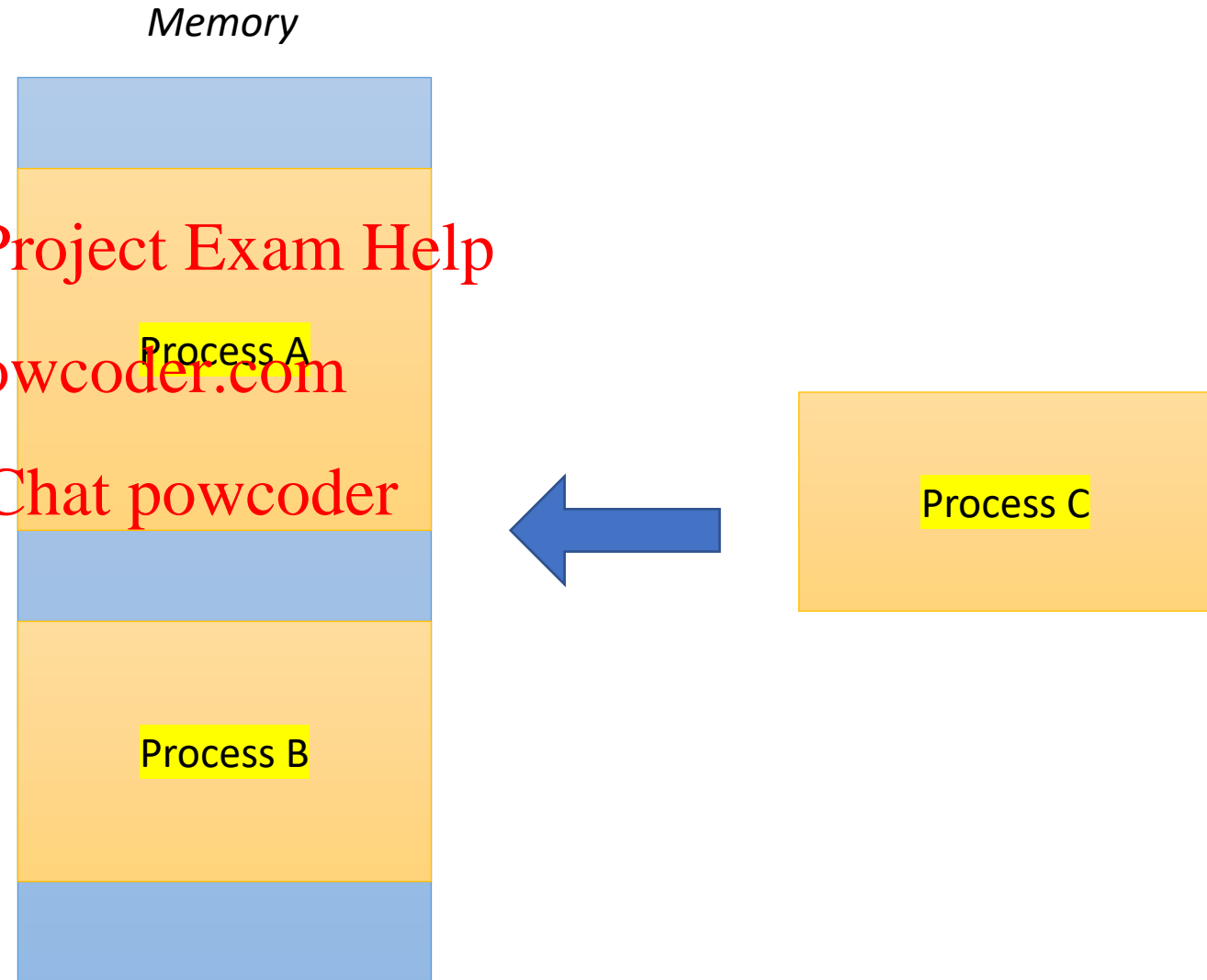


Physical memory

Assignment Project Exam Help

Why paging?

- Why not just place the entire memory image of a process in one chunk in the physical memory ?



Goals of memory virtualization

Add WeChat powcoder

- Transparency: user programs should not be aware of the messy details

Assignment Project Exam Help

<https://powcoder.com>

- Efficiency: minimize overhead and wastage in terms of memory space and access time

Add WeChat powcoder

- Isolation and protection: a user process should not be able to access anything outside its address space

Assignment Project Exam Help

How can a user allocate memory?

- OS allocates a set of pages to the memory image of the process

- Within this image

- Static/global variables are allocated in the executable
- Local variables of a function on stack
- Dynamic allocation with malloc on the heap

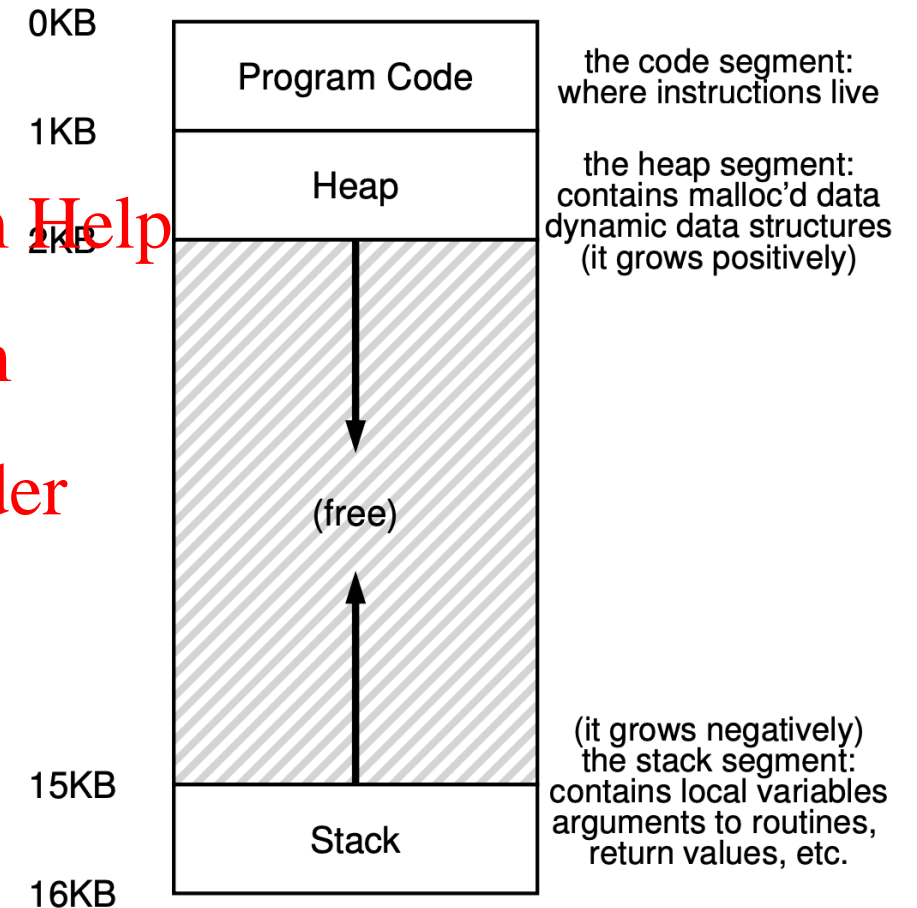


Figure 13.3: An Example Address Space

Assignment Project Exam Help

Memory allocation system calls

- malloc implemented by C library
 - Algorithms for efficient memory allocation and free space management
- To grow heap, libc uses the brk/sbrk system call
- A program can also allocate a page sized memory using the mmap() system call
 - Gets “anonymous” page from OS

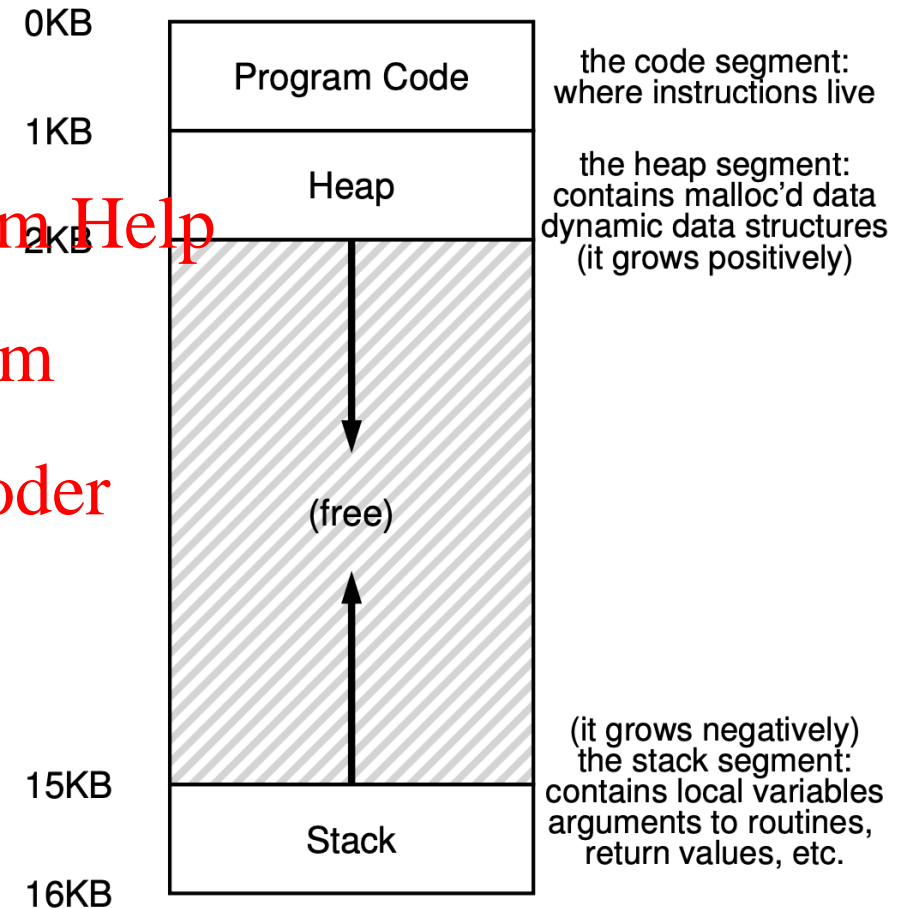


Figure 13.3: An Example Address Space

A subtle point: what is the address space of the OS?

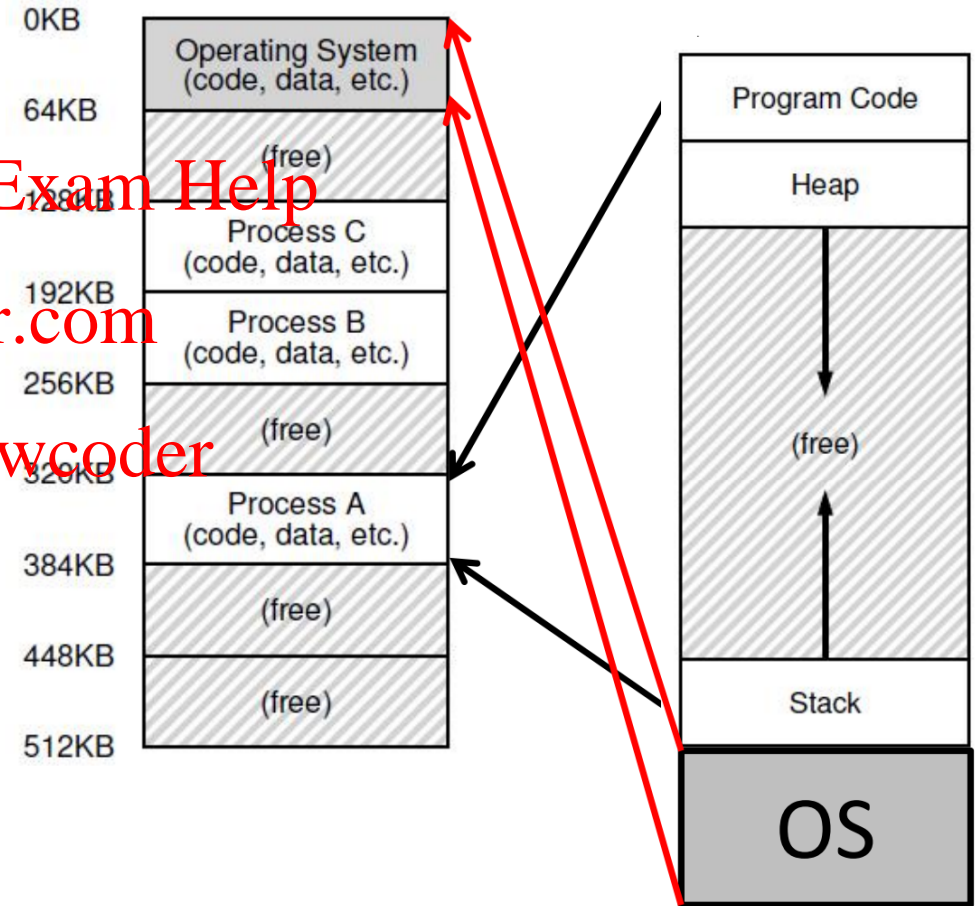
Add WeChat powcoder

- OS is not a separate process with its own address space
- Instead, OS code is part of the address space of every process
- A process sees OS as part of its code (e.g., library)
- Page tables map the OS addresses to OS code

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Assignment Project Exam Help

How does the OS allocate memory?

- OS needs memory for its data structures
- For large allocations, OS allocates a page
- For smaller allocations, OS uses various memory allocation algorithms (more later)
 - Cannot use libc and malloc in kernel!

Assignment Project Exam Help

Disclaimer

- *These lecture slides are based on a slide set by Youjip Won (Hanyang University) and Mythili Vutukuru (IIT Bombay)*

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder