Processes (1)

Dr. Jun Zheng

CSE325 Principles of Operating Systems

8/30/2019

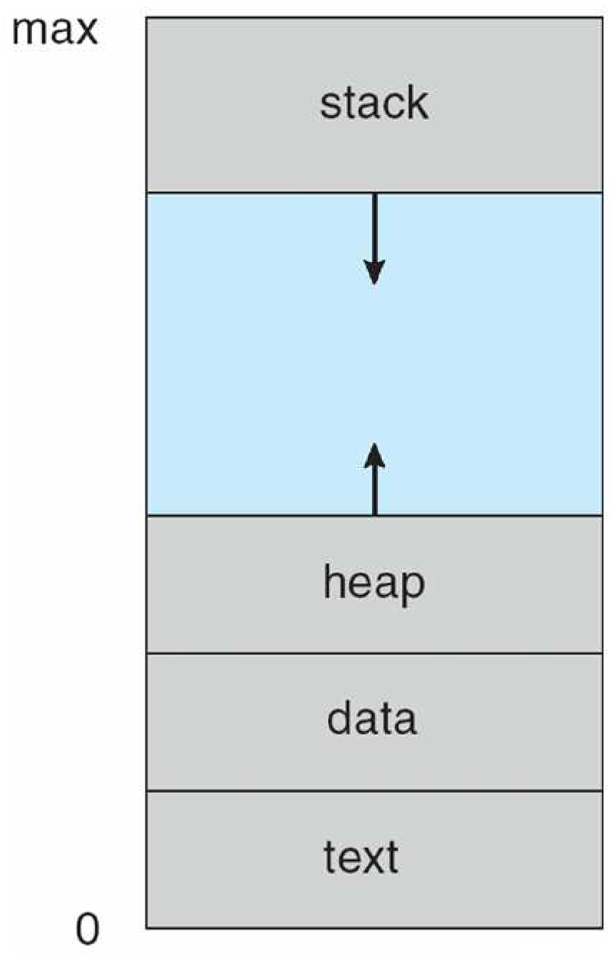
NEW MEXICO TECH

SCIENCE · ENGINEERING · RESEARCH · UNIVERSITY

Process Concept

- An operating system executes a variety of programs:
  - Batch system – **jobs**
  - Time-shared systems – **user programs** or **tasks**
- Textbook uses the terms ***job*** and ***process*** almost interchangeably
- **Process** – a program in execution; process execution must progress in sequential fashion
- Multiple parts
  - The program code, also called **text section**
  - Current activity including **program counter**, processor registers
  - **Stack** containing temporary data
    - Function parameters, return addresses, local variables
  - **Data section** containing global variables
  - **Heap** containing memory dynamically allocated during run time

Process's Address Space (idealized)

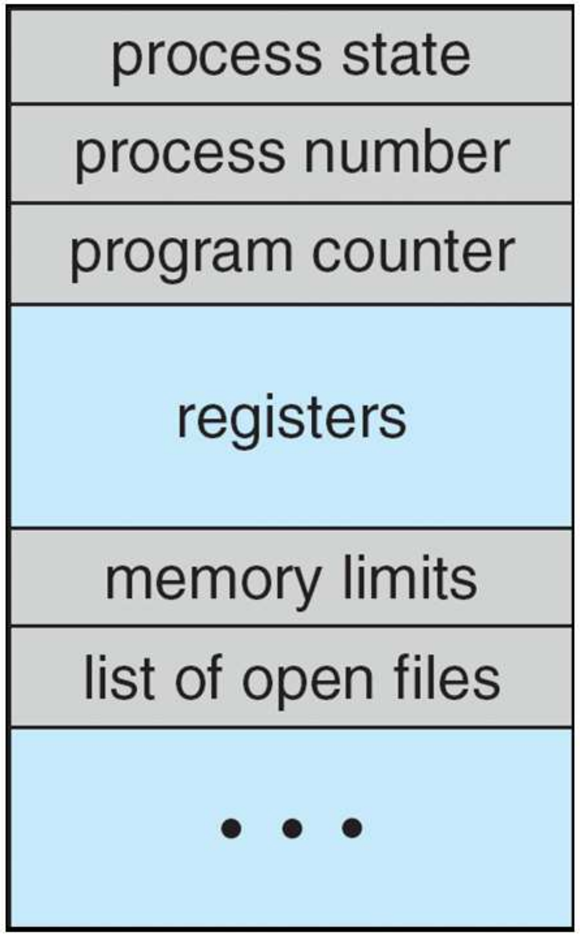


Process Concept (Cont.)

- Program is ***passive*** entity stored on disk (**executable file**), process is ***active***
  - Program becomes process when executable file loaded into memory
- Execution of program started via GUI mouse clicks, command line entry of its name, etc
- One program can be several processes
  - Consider multiple users executing the same program

Process Control Block (PCB)

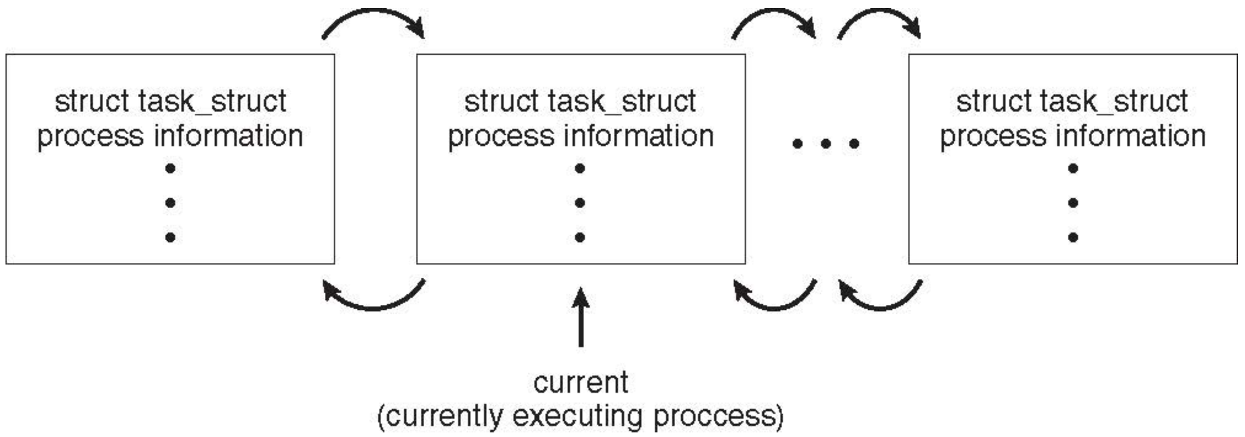
- Information associated with each process (also called **task control block**)
- Process state – running, waiting, etc
- Program counter – location of instruction to next execute
- CPU registers – contents of all process-centric registers
- CPU scheduling information- priorities, scheduling queue pointers
- Memory-management information – memory allocated to the process
- Accounting information – CPU used, clock time elapsed since start, time limits
- I/O status information – I/O devices allocated to process, list of open files



Process Representation in Linux

- Represented by the C structure `task_struct`

-
  ```
  pid t_pid; /* process identifier */
  long state; /* state of the process */
  unsigned int time_slice /* scheduling information */
  struct task_struct *parent; /* this process's parent */
  struct list_head children; /* this process's children */
  struct files_struct *files; /* list of open files */
  struct mm_struct *mm; /* address space of this process */
  ```


struct task_struct
process information
struct task_struct
process information
struct task_struct
process information
current
(currently executing proccess)