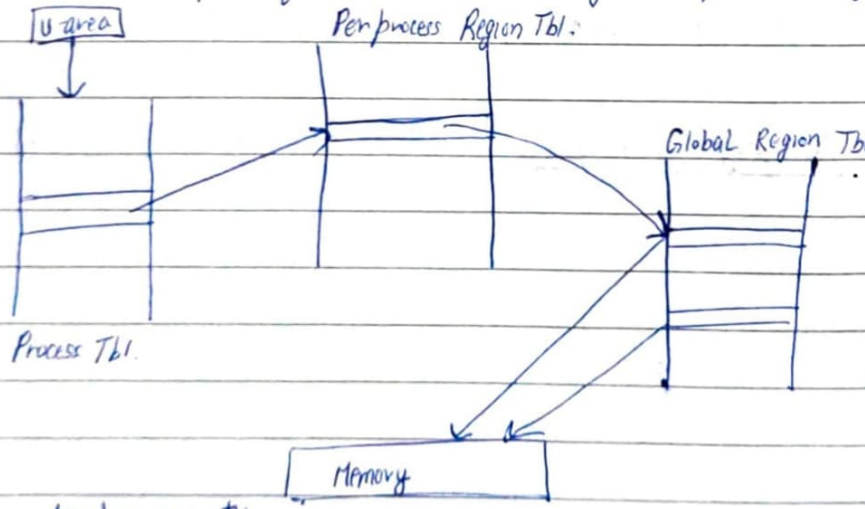
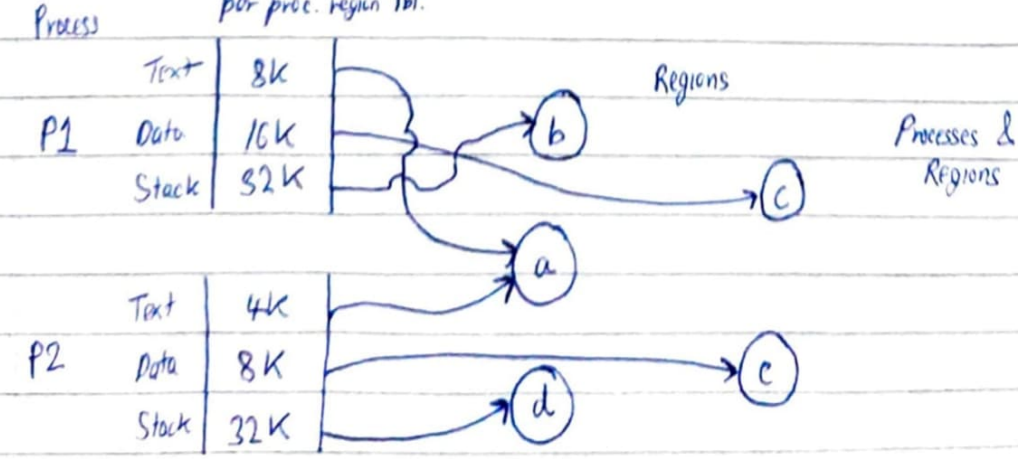
× Processes — Regions & Page Tables —

Regions :

× Contigious area of virtual address space of a process.
× Treatable as distinct shareable / protectable objects.
× Common regions : text/code, stack, data, bss, heap.
× Shareable : As copies (text regions)
: Via cooperation for common shared-memory regions.
× Private, per process region table : pregion & region table for maintenance.
: Exists in process table, u area are separate allocated area in memory.

× pregion : Points to a region table entry, containing start virtual address.
: Can be part of u-area or existing as separate entry/structure.

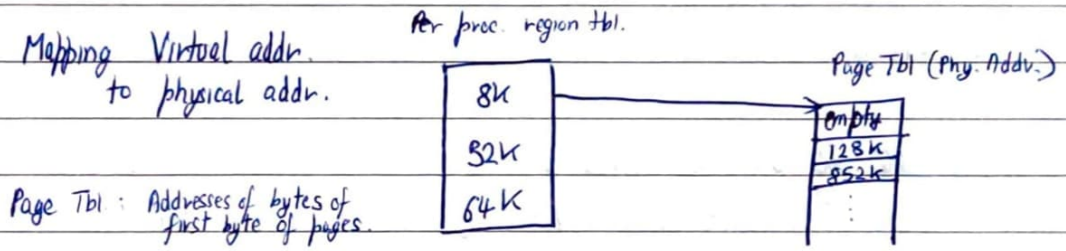




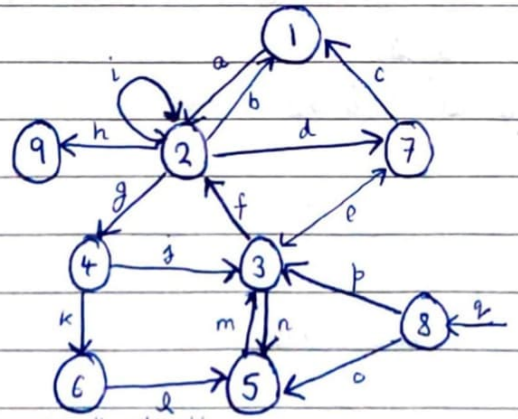
× Region concept is independant of OS's memory management policies (Implementable via segments (contiguous) or pages (discontiguous)).

Pages & Page Table –:

× Memory mgmt. divides physical memory into set of equal-sized blocks/pages.
× Page Size : 512B – 4KB
× Memory location address : <page #, byte offset in page>.
× Allows for greater flexibility in physical memory assignment. & reduces fragmentation.



Process Control –:



a : syscall, interrupt
b : return
c : return to user mode
d : preempted
e : preempted
f : reschedule process
g : sleep
h : exit
i : interrupt, interrupt return
j : wakeup
k : swap out
l : wakeup

1 : User Mode, Running
2 : Kernel Mode, Running
3 : Ready, In Memory
4 : Asleep, In Memory
5 : Ready, Swapped
6 : Asleep, Swapped
7 : Preempted
8 : Created
9 : Zombie

Process Table :

- Includes data on : Process state
  : Scheduling params (kernel or user mode)
  : Signal field for unhandled signals
  : Timers for process execution time & kernel resource utilization.
  : Event descriptor (for sleep state, etc)
  : Process identifiers ID & UIDs for privileges.
  : Fields to locate process & u-area in memory.

U-Area :

- Pointer to process table for identifying u-area entry.
- Real & Effective user IDs (creator & current executor, etc)
- Timer fields to record execution time in user & kernel mode
- Array for handlers for signals.
- Control field for identifying login terminal associated, if any.
- Error field for syscall errors.
- Return value field for syscall returns.
- I/O params for data transfer
- Current directory & current root inodes.
- User FD table
- Limit fields for size & write limits.
- Permissions.

Process Context —:

- User Address Space Content
- Content of H/W registers
- Kernel Data Structures that relate to the process
- Context is union of : User-Level Context
  Register Context
  System-Level Context
- User level context consists of :
  - Process' text, data, stack, shared memory occupying virtual addr. spc.
  - Parts of virtual addr spc. that do not periodically reside in main memory
- Register context :
  - Program Counter (PC) (Addr. is virtual addr. in user addr spc.)
  - Processor Status Register : H/W status of machine.
  - Stack Pointer : Current addr. of next entry in usr/kernel stack.
  - General-Purpose Registers (GPR) : Data generated by process during execution.
- System-Level Context :
  - Static Component :
    - Process table entry of process, defining state & control info.
    - U-Area, containing process control info to be accessed in process context only
    - Pregion entries, region tables & page tables.
  - Dynamic Component :
    - Kernel stack, containing stack frames of kernel procedures during process execution in kernel mode.
    - Composed of set of layers, visualized as UFO stack.