

## CS3510 : ASSIGNMENT 1

ABHAY SHANKAR K : CS21BTECH11001

- (1) (a) Describe a mechanism for enforcing memory protection in order to prevent a program from modifying the memory associated with other programs.
- Dynamic tainting: During dynamic memory allocation, both the memory and the corresponding pointer are 'tainted' using the same 'taint mark'. Taint marks are then suitably propagated while the program executes and are checked every time a memory address  $m$  is accessed through a pointer  $p$ ; if the taint marks associated with  $m$  and  $p$  differ, the execution is stopped and the illegal access is reported.
  - Paged virtual memory: Given an application, unallocated pages and pages allocated to any other application, do not have any addresses from the application point of view. Every memory address either points to a page allocated to that application, or generates an interrupt called a page fault.
- (b) While memory protection is necessary, there are several examples where programs require memory to be shared between them. Please give two such examples.
- Shared memory facilitates inter-process communication (IPC), i.e. a way of exchanging data between programs running at the same time. One process will create an area in RAM which other processes can access.
  - Shared memory can be used to conserve memory space by directing accesses to copies of a piece of data to a single instance instead.

Source : Wikipedia

- (2) Please study the PCB (Process Control Block) structure of a Linux system from any reliable Internet source. Describe any 10+ fields in the PCB of a process in the latest Linux Operating System. Please cite your source as well like in question 2.

unsigned int	__state;	::The process state::
int	prio;	::Process/thread priority::
struct sched_statistics	stats;	::Scheduling statistics::
int	pdeath_signal;	::The signal sent when the parent dies::
pid_t	pid;	::Thread ID from kernel's POV::
pid_t	tgid;	::Thread group ID equivalent to POSIX process-ID::
struct task_struct __rcu	*real_parent;	::Real parent process::
struct list_head	children;	::List of child processes::
struct list_head	sibling;	::List of sibling processes::
struct task_struct	*group_leader;	::Leader of thread group::
struct list_head	ptraced;	::list of tasks this task is using ptrace() on::
struct list_head	thread_group;	::list of threads of this process::
unsigned long	nvcs;	::#Voluntary Context switch counts::
unsigned long	nivcs;	::#Involuntary Context switch counts::
u64	start_time;	::Monotonic time in ns::
u64	start_boottime;	::Boot based time in ns::
char	comm[TASK_COMM_LEN];	::executable name, excluding path, like ls::
struct fs_struct	*fs;	::Filesystem information::
struct files_struct	*files;	::Open file information::

Sources :

- GitHub
- Stack Overflow