19CS2106S and 19CS2106A
Operating Systems Design
Skilling Assignment
Project list, guidelines, and submission

As you are aware that we insisted all the students to work on single xv6 code base for adding commands/system calls/different advanced features/functions. Such that, at the end of the semester we will have one customised/enhanced operating system as outcome. As a part of the skilling exercises, each student is supposed to implement and submit a working version of xv6 with an improved shell, missing user commands, or new/better functionalities within the kernel as project.

In this regard, we assigned Individual/standalone project to each student. The customization and enhancement tasks are taken from the course skilling exercises already done by the student. It is not team project. At maximum, per project, a student should add an Editor, shell, command/system call, and enhancement.

You project should customize the xv6 code base from $git clone git://github.com/mit-pdos/xv6-public.git 1900XXXXX-xv6

The corresponding directory 1900XXXXX-xv6 should be in your OSD login account.

Follow the given template/format while preparing the project report. Use visual paradigm or Star UML for drawing the diagrams.

Deadline for submission of PDF project report is 29th (cluster – II) and 30th October (cluster – I) respectively. Evaluation is in your skilling hours.

Project. No	**Task -1, 2 common to all projects - improved shell and Editor**		**Task -3 Adding a command - missing user commands**	**Task -4 Adding a system call or enhancement - new/better functionalities**	**Task – 5 Adding a system call or enhancement - new/better functionalities**
1.	shell	editor	square	date	lseek
2.	shell	editor	cp	Triply-Indirect-Block filesystem in xv6	dup2
3.	shell	editor	wc	Double-Indirect-Block filesystem in xv6	Completely-Fair-Scheduler (CFS) in xv6

4.	shell	editor	pwd	Priority scheduler in xv6	Implement lottery scheduling for XV6 operating system
5.	shell	editor	cd	chpr	Demonstrate Virtual Memory in xv6
6.	shell	editor	mv	back trace	Add improvement to the original implementation of xv6: Copy on Write optimization.
7.	shell	editor	rm	xv6 shared memory	Enhance xv6's file system with partitions and mounting
8.	shell	editor	ls	Porting xv6 with POSIX compliance + VFS + ulibc + ACPI	xv6 file system to add Mirrored RAID feature
9.	shell	editor	head	file system checker: fschk	Add a new system call for xv6 called getprocs(), which returns the number of processes that exist at the time of the call.
10.	shell	editor	touch	Demonstrate Virtual Memory in xv6	Add improvement to the original implementation of xv6: Kernel level threads.
11.	shell	editor	tail	xv6 operating system with stack and heap virtual memory relocated	Add a System Call in XV6 operating system which will return number of System call a process does.
12.	shell	editor	clear	top	implement copy-on-write fork in the xv6 kernel. Make a system call numfree() that returns number of free page frames in physical memory
13.	shell	editor	shutdown	getprocess	xv6 page fault handler
14.	shell	editor	getpinfo	Berkley Fast Filesystem (FFS) in xv6	Support kernel-level threading, so that concurrency within a single user-level process is possible. Provide a basic synchronization mechanism -- the mutex lock -- for said threads to use.
15.	shell	editor	ps	clone(), a new system call to create a new thread	Implementing a system call for acquiring fs-related statistics (superblock/bitmap)

16.	shell	editor	rename	join() to wait for a thread	Implementing "lazy allocation" within the sbrk() system call
17.	shell	editor	System call tracing	thread_create(void (*start_routine)(void*), void *arg), call malloc() to create a new user stack, use clone() to create the child thread and get it running.	dup
18.	shell	editor	halt	thread_join() should also be used, which calls the underlying join() system call, frees the user stack, and then returns.	xv6 with very simple file system
19.	shell	editor	nice	lock_init, lock_acquire(), and lock_release() to control the thread library hardware, using X86 atomic exchange to facilitate spin lock.	xv6 with signals
20.	shell	editor	malloc	alarm	Add support for symbolic links
21.	shell	editor	login	wait2	Increasing the maximum file length using a modified single indirection