CS3500 Assignment 6 Lab Report CS19B012

1) In kernel/vm.c file, uvmcopy() is modified as follows, so as to map the parent's physical pages into the child, rather than allocating new pages:
   - To mark the shared page as non-writable & as a COW page, for the former part, *pte &= ~PTE_W is written. But for the latter part, there’s no flag bit designated, so a new flag bit PTE_COW has been defined in riscv.h, alongside PTE_R, PTE_W, etc
   - Now, *pte | = PTE_COW statement has been added to mark it as a COW page.

2) In this part, usertrap() in kernel/trap.c has to be modified, so as to accommodate write pagefaults caused by the afore defined COW pages. This is accomplished as follows:
   - If it is a COW page, data is copied from old page into a newly memory allocated page & mark the generated page as a non-COW & writable page.
   - Else if the error was generated due to other store page faults, user trap message is printed.

3) To make sure that each physical page is freed when the last PTE reference to it goes away, a new array reference_counts[PHYSTOP/PGSIZE] has been defined in kernel/kalloc.c .
   - A new function increase_reference_counts() has been defined so as to increase reference count when used in uvmpcopy() in kernel/vm.c .
   - To do the opposite (i.e) decrease the reference count, a new segment has been added to kfree(), which decreases the reference page count if the value in array is non-zero, so as to make sure it’s not freed, freeing is finally done when the value reaches 0, which is accomplished after a series of decrements

4) A new segment has been added to copyout() in kernel/vm.c, just as the one in (2) .