Problem 1.

(A) 3-9: *Context switch* happened when we need to switch our original process to another new process. When an interrupt or system call appeared and the system is about to change its routine to a new process, the state of the original process will be saved into the process control block(PCB\_original), then reloads the saved state for the new process and runs it. When another interrupt is triggered, calling the system to change its routine back to the original process, the state of new process will be saved into the(PCB\_new), then system reloaded the old process’s saved state, executing it again. The whole process mentioned above is named *Context switch.*

3-11: When the parent process terminates before its child processes, all its children will become orphan and immediately be “adopted” by init process, whose PID is 1, to let the orphans keep running. Another situation is that when a child process terminates before its parent process and the parent forgets to retrieve it by calling functions like wait(), then the child will become a zombie process and occupied a minimal space of the memory. If this happened, init will also adopt the zombies either when the zombie’s parent process is killed by system call or exits itself automatically. After zombies being adopted, it will be removed by init process since init periodically also runs wait() function.

(B) 1: typeof() is a keyword in C language, we can either write with expression or write with a type to indicate the argument of typeof(). In general, it takes one argument and returns its type.

2: offsetof is macro. It has two parameters, MEMBER and TYPE, returning the address of a member we said it MEMBER of a structure of a type TYPE stored in the memory from address o.

3: According to LKD, macro container\_of() let us find the parent structure containing any given member variable. With it, we can define a simple function to return the parent structure which contains any list\_head.

4: typeof is the extension instead of macro while offsetof is indeed macro. Macro is a UI or program that it expands itself when being used. Marco statement contains parameter information and the macro definition name. It is helpful when a series of instructions is being used by different programmers working simultaneously.

5: It is not really necessary, after we tried it ourselves, we found it possible to write the macro without it. The compiler doesn’t really care for the value, it just adds offset to the address of the structure and returns the new address.