➢ Task Control Block (TCB) or Process Control Block (PCB)

When a task is created, it is assigned a task Control Block (TCB). TCB is a data structure that is used by OS to maintain the state of a task when it is preempted. When the task regains control of the CPU, the TCB allows the task to resume execution exactly where it left off. The TCB is initialized when a task is created. An example TCB structure used in μC/OS-II (a Real-Time Operating System) is shown below

```
struct os_tcb {

   OS_STK         * OSTCBStkPtr ; // a pointer to the current top-of-stack for the task
   struct os_tcb * OSTCBNext;  // for doubly linked list of TCBs
   struct os_tcb * OSTCBPrev;  // for doubly linked list of TCBs
   INT16U        OSTCBDly;     // a task needs to be delayed for this amount of clock ticks
   INT8U         OSTCBStat;    // task state: Dormant, Ready, Running, Waiting
   INT8U         OSTCBPrio;    // task priority

} OS_TCB
```

In this class project, things are much simpler because we have only 3 tasks, all tasks are always ready to run, and the priorities of the tasks are the same. Thus, it would be enough to save the following registers to TCB when context-switched.

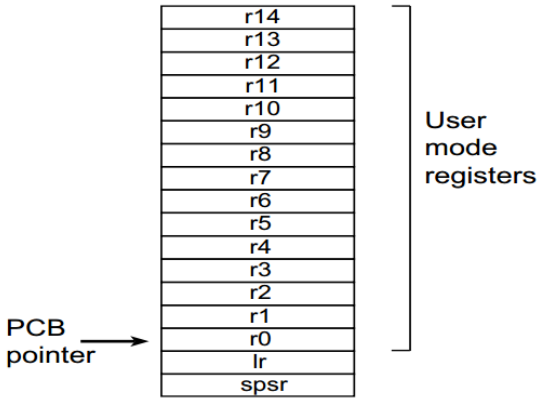

Figure 6-2 PCB layout

➢ One thing you have to be careful: The fact that User mode registers (because the tasks are running in User mode) have to be stored in PCB means that the ISR (or scheduler) in Interrupt (IRQ) mode should have access to User mode registers. To make your life easier, ARM provides optional suffix (^) for LDM and STM. With that suffix, the User mode registers are accessible in other (IRQ, FIQ…) mode instead of the current (IRQ, FIQ…) mode registers.

For example, assume that the following `STM` instruction is executed in ISR (Interrupt Service Routine), which is running in IRQ mode.

`STM sp, {r0-lr}^` // store r0 ~ r14 **User-mode registers** to stack.