(1) How do threads reduce the high context switching overhead in processes?

Threads reduce the overhead in performing context switching by not “switching” the entire PCBs. Threads perform context switching by switching only TCBs, which are smaller than PCBs. Switching smaller amount of data reduces the time needed for each context switching.

(2) Show the typical internal structure of a process that consists of multiple threads.
(3) What information do multiple threads in a process can share?

   Global variables

(4) Can threads in a process share global variables in other processes? If yes, briefly describe how. If not, briefly explain why not.

   No. Global variables in a process is still protected from other processes.

(5) Most of the operating systems today includes Stack Pointer (SP) register in their private PCB (i.e., “TCB”). What are the advantages in this design? Mention at least two different advantages.

   By including SP register in TCB, it is possible for threads to make their own function calls (i.e., “subroutine calls”). Without including SP register in a TCB, threads will not be able to make function calls.