(1) How do threads reduce the high overhead of processes when processes try to share data in shared memory?

Threads reduce the overhead in performing IPCs by not using an operating system. Threads share by global variables in the same process, which allows threads in a process to share global variables, just as local variables in a process. Without using an operating system, each memory access will be faster. For example, processor mode change is not required. Context switching between a process that is performing an IPC and an operating system is not required either.

(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) What information must be managed in TCB (thread control block, also known as “private PCB”) and what information should be still in the PCB (also known as “global PCB”)? Note: This question does NOT ask example, but the type of the information in PCB and TCB as ideas.

   PCB includes the information that is common to all threads in a process, while TCB includes information specific to each thread in a process.

(5) Which of PCB (global PCB) or TCB (private PCB for each thread) does Program Counter (PC) register belong to? Briefly (but with a proper emphasis on the essential reason) justify your opinion.

   PC register should be in TCB. It is because each thread in a process usually performs a specific (different) function in parallel with other threads in the same process. This is impossible if each TCB does not have a copy of PC register.