Provide succinct answers to the following problems:

- **Problem 1:**
  What is an I/O controller? Explain its role. How is it different from I/O devices?

- **Problem 2:**
  Why do we use DMA-enabled devices? Explain the working principles of DMA.

- **Problem 3:**
  What is memory–mapped I/O? How is it different from CPU-supervised I/O?

- **Problem 4:**
  What is a *bootstrap loader*? Is it different from the bootstrap program and if so, how?

- **Problem 5:**
  An application address space consists of text, data, stack, heap and shared memory. Which of those process elements can either grow or shrink and which cannot? Justify.

- **Problem 6:**
  What is a process state and what are such possible states that a process might find itself in? Designate which states are bound with the kernel–mode and which with the user–mode of execution.

- **Problem 7:**
  What does the *fork()* Unix system return? What its effect? In the same spirit, what does the *exec()* call return and how does it complement the work of *fork()*?

- **Problem 8:**
  Write a program that uses both *fork()* and *exec()* calls and returns the top-10 jobs executing in your system at the time of execution (excluding of course the program in question). Provide code and execution script.

Homework answers are expected to be typed using a text-formatter (for example *LibreOffice, OpenOffice* or *LATEX*). Handwritten answers are also welcome provided that they are legible.

---

1. portions of memory *shared* with other process for collaboration purposes