

1.

To understand what priority inversion is, you must first understand what priority-based scheduling is. The idea is that a task that is more important or has a higher priority task can be run instead of the lower task. An example of priority inversion is if you have process A and process C. Process A has a higher priority than process C. They both share a memory space. Process B has lower priority than process A, but it is running in shared memory so process A waits for process C to finish. Process B does not share the memory space with A and C. If B runs, then C will have to wait until B finishes. If C must wait, then A also must wait. This is priority inversion.

There are many ways to remedy this. One such way is priority inheritance. Whenever a high priority task is waiting for a low priority task that is using a shared resource, the low priority task is temporarily given the same priority level as the highest priority level task.

2.

The difference between a real time operating system (RTOS) and a general-purpose operating system (GPOS) is that RTOS's can guarantee that the task of highest priority will be run immediately. The Linux "real time" scheduler can make this happen most of the time, but not always. Also, the latency of the Linux scheduler is in the order of magnitude of 10's to 100's of microseconds. An RTOS generally has a latency of a few microseconds or less.