1. Consider a processor that supports virtual memory. It has a virtually indexed physically tagged cache, TLB, and page table in memory. Explain what happens in such a processor from the time the CPU generates a virtual address to the point where the referenced memory contents are available to the processor.

2. Distinguish between segmentation and paging.

3. Explain all the actions from the time a process incurs a page fault to the time it resumes execution. Assume that this is the only runnable process in the entire system.

4. Explain the following terms: working set of a process, thrashing, paging daemon, swapper, loader, and linker.

5. Explain page coloring and how it may be used in memory management by an operating system.

6. Explain clearly the costs associated with a process context switch.

7. Explain the functionality of the different layers found in the network protocol stack of an operating system such as Linux.

1. Physical memory would partition into equal sized page frames. Then we need to do is mapping. Mapping virtual addresses to physical addresses. CPU only knows about the virtual addresses and it will use the Page Table to do the translation. Then we use TLB to speedup this kind of address translation. And TLB is between the CPU and Page Tables. For those pages missed in the cache, what we need to do is to update the TLB. We even can use multiple-level paging.

2. Paging is that computer memory is divided into some really small partitions and it has a fixed size, which means it has the same amount of data and it is a physical unit. But the size of segmentation is changeable and logical. Normally pages all have contiguous memory address but segmentation is an independent space which include different address. For example, segmentation may has an address to store data and another address to store code. Also the procedures in page cannot be shared while procedures in segmentation can be shared between users. The protection of the program and data in segmentation can be separated while the protection in paging cannot.

3. If there is only one runnable process in the system. Once there is a page fault. The CPU will suspend the process. And the kernel will send to a page fault message to library OS. Once library OS get such information, it would to

for the page table to whether there is a such page. If there is no such page, there must be a fault. The process might exit. Otherwise, it will get this new page from the storage and put it in the cache and update the TLB. Once it updates the TLB, it will notify the kernel or directly to CPU. Then CPU can get that memory can resume the process.

4.Thrashing is a phenomenon in virtual memory schemes when it waste too many time in swapping pages. Normally it is caused by too many page fault. Working set just means the amount of memory which a process might require in period.
And a swapper is something that manipulates the entire process.
Loader is one part of the OS which is necessary to load the programs and libraries.
A program might turns to more than one objects after compiling. And linker is responsible for links them up, including links them to the library functions.

5. Page coloring is a process that try to allocate those free pages and try to make them as contiguous address. Normally it is used in mapping virtual address and physical memory by operating system.

6. The cost of context switch are most from the following parts. Switching form one process to another process might cause the change of the physical address, which is real expensive, and it also needs time to save the data and state of the old process. And some inter-process-call would happen in this kind of context switch. A process might send some request to a library OS, it might even cause a broad crossing. So all of these stuff could be the cost of context switch.

7. Application Layer -- System Call Interface -- Protocol Agnostic Interface -- Network Protocol -- Device Agnostic Interface -- Device Drivers -- Physical Hardware.
The app layer is the layer people can use the applications on. And all the apps access to a network starts from this layer.
And in System Call Interface calls are made form the application layer to the kernel.
The 3rd layer is where the socket is created.
The Network Protocol takes care of how to send and receive the data. That is, including some functions like routing.
The Device Agnostic Interface is used to link the Kernel and network device drivers.
The Device Drivers Layers is those drivers to drive the physical devices. Without drivers, devices could not work well.

The lowest level is physical level, including all the real devices.