**13.** Schedule the following jobs (given as Name/Duration/Deadline) so that they all meet their deadlines: A/5/9, B/7/13, C/1/10.

**SOLUTION**

The initial time is t=0

t=0 + 5 => A finished at time t=5

t=5 + 1 => C finishes at time t=6

t=6 + 7 => B finishes at time t=13

possible order: ACB

OR

t=0 + 1 => C finishes at time t=1

t=1 + 5 => A finishes at time t=6

t=6 + 7 => B finishes at time t=13

possible order: CAB

**14.** Give one advantage and one disadvantage of the segmented allocation method over the paged allocation method.

**SOLUTION**

A disadvantage of the segmented allocation method over the paged allocation method is that the segmented allocation causes memory fragmentation.

An advantage of the segmented allocation method over the paged allocation method is that it is faster to compute the exact address.

**15.** When would you load into memory the pages of a program that is being started?

**SOLUTION**

Here is applied the Locality principle. A process is likely to need soon the pages next to the page that was just loaded, so there will take place the prefetch of the neighboring pages.

**16.** When does a process change state from RUN to READY?

**SOLUTION**

**17.** Given a UNIX file system configured with a block size of B bytes that contain A addresses, and i-nodes having S direct link, one simple indirection link, one double indirection link, and one triple indirection link, give the formula for the maximum file size possible.

**SOLUTION**

**18.** What happens with the data when you delete a file that has a hard link pointing to it?

**SOLUTION**

The data will remain on the hard-disk, because by deleting the file we only delete the pointer to the data from that file, but the pointer to the data from the hard link will remain, and also the data.

**19.** Give a method for prevendint deadlocks.

**SOLUTION**

Pick an order for the resources and lock and unlock them in that order everytime.

**20.** What is a binary semaphore, and what is the effect of its P method, when called by multiple concurrent processes/threads?

**SOLUTION**