Printers typically can print only a single document at a time and require seconds or minutes to do so. If more than one application requires printer access, it might result in delay and slow down the application for as long as the printer is busy in serving another application. '**Printer spooling**' has overcome this delay, as the Documents formatted for printing are stored into a buffer and retrieved and printed by a printer at its own rate. With spooling, multiple processes can write documents to a print queue without waiting. As soon as a process has written its document to the spool device, the process can perform other tasks, while a separate printing process operates the printer.

Implement printer spooler of a working environment equipped with **3 printers** using linked lists. Each node in the linked list **printerSpooler** depicts a **user** with a **user_id**, having identical values assigned to each **user_id** i.e. I,2,3….,. The **user** class might contain more than one document of a user in the buffer for printing. The functionalities are discussed below:

Insertion:

Every time when a user sends a print command a check is made if the user already exists in the **printerSpooler**. If the user's buffer already exists, the document node is inserted in the **user** linked list according to the **priority** of document.

If the user's buffer does not exist than a new node is inserted in the **printerSpooler** linked list.
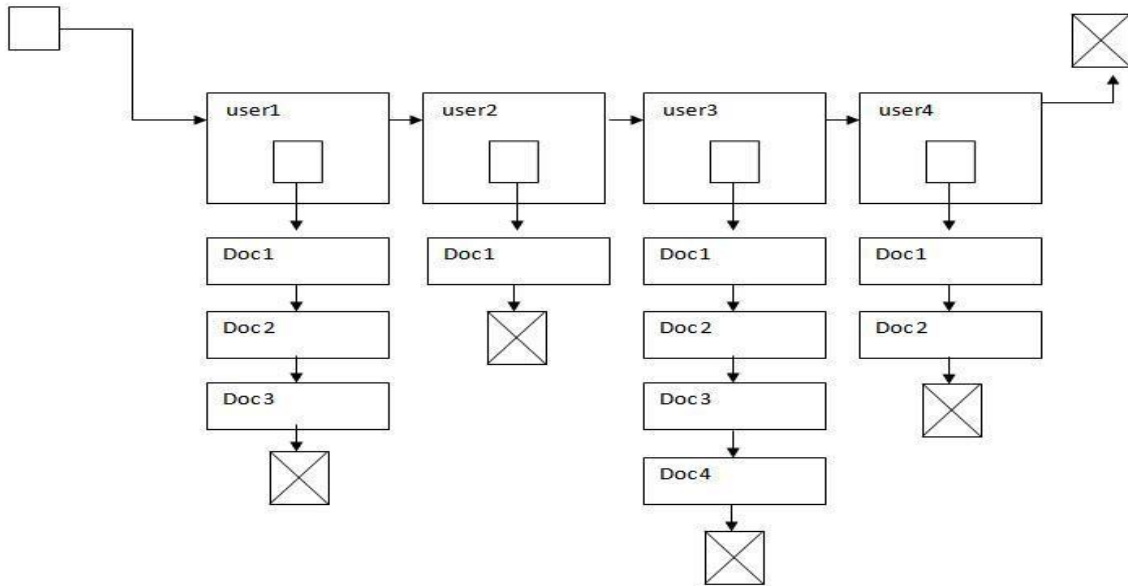
Deletion:

If all the documents in a user linked list gets printed and the user's **bufferIsEmpty**, User node gets deleted.

A user can also cancel a print command, if the **document_status**is 'spooling'. This results in the node deletion for that document from **user** linked list.

Advanced Features:

- A user can look for the status of the documents in his queue.
- Every document to be printed should have a priority from 1-5 i.e. 1(most prior) to 5(least prior)
- Insertion of document in **user** linked list should be on the basis of priority. High priority documents should be entertained first. In case of same priority documents First come first serve scheduling should be followed.
- When a command is sent by the user for printing, it should be queued in the printer having least burden.
- When a document gets printer, the user should be notified.

printerSpooler Printer1