Swapping and charge for ECS

1) Users will be __charged__ for ECS space by $\int w\,dt$, where w is the number of words __attached__. A word is attached if it is in the map or if the user has specifically requested its attachment. It is possible to reference a word which is not attached with ECS read or write operations; if the word turns out not to be in ECS, the disk system will attach it, do the operation, and release it. Of course attached words are implemented by the disk system. A word is released either when it is removed from the map, or explicitly.

2) An allocation block is needed for attaching words, so that the number of words of ECS occupied by a process can be controlled.

3) Files which are to be shared by many users can be attached by the system; this results in their being treated as overhead. Alternatively there could be an EDITOR process which would attach the files and hence be charged for it. This is implemented by allowing a process to be __responsible__ for the files; in this case the process cannot release it until no-one else is using it.

4) When a process P is being considered for swapping, the extra ECS required to bring it in is computed. If the available space — this amount is not too small, the process can be brought in. The extent to which it is sharing files with others is not considered further. If it is necessary

to compute the load it places on the system at a later time (e.g. which asks for more space). This is considered to be the amount of ECT which would be freed if it were kicked out.

5) Processes are also charged for the attachment operation, at a rate calculated on the assumption that every attachment requires a disk transfer.