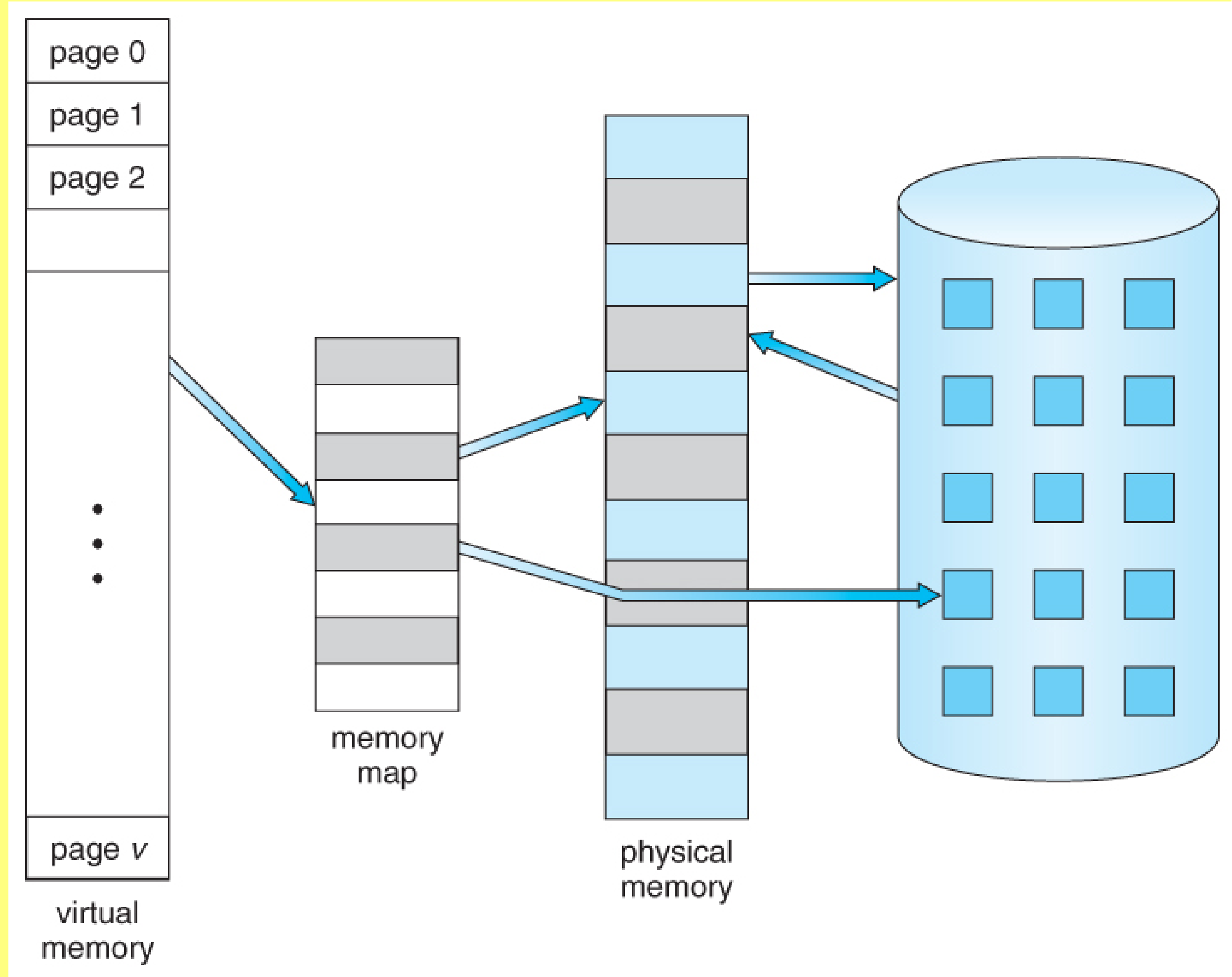
If the file is too big, how can you use the Operating System's Virtual Memory to solve disk access problems?

Be sure to comment on at least another students post.

My Explanation:

When we have a file that is considered big for an operating system's virtual memory to handle, we can solve the disk access problems by breaking the process memory requirements down into smaller bites or pages in order to store the pages in memory non-contagiously.

For example, we can practice this method for Virtual Memory by loading only the portions of processes that were actually needed in the program. The advantages of having these loading the portions of the processes given to us is that programs can be written for a much larger address space, improve CPU utilization and system from left over memory to utilize it, and less I/O needed to swap processes in and out of the RAM in order to speed processes up accordingly.



<u>Other Explanation:</u>

When a file is loaded by a program, it attempts to load it entirely into RAM. This could be costly for large files, like images and videos. When reducing the quality of the file is impossible, and it's too big for the RAM, it is loaded partially into RAM and the rest is loaded onto the hard disk as "dynamic memory", or Virtual Memory. The operating system attempts to load the least-used parts of the file in RAM, but it cannot always easily determine which parts those are, in which case it loads it randomly.

The Virtual Memory in Windows is set automatically in most cases, to be 3 or 4 times the amount of RAM installed (modern versions have a minimum and maximum, older versions use a raw multiple). This is done automatically by the operating system, and (in c++ and other native languages) no additional programming is needed. Some Linux distributions resize it automatically, and it can be changed manually by the user in all OS's. Unless you have an unusually small/large amount of RAM, or run lots of massive programs regularly, it's ill advised to change it from the default.

When speed is needed, virtual memory must be avoided, as hard drives can have 10x-500x the access time of RAM. However, the virtual memory is occasionally unavoidable; usually when other programs are taking up available memory (which can be quite often).

If you're expecting to read files larger than the RAM you can allocate, then it's best to devise a special reader that doesn't load the whole file into RAM. For example, a spell checker is going to check 20 letter words almost never, so the 20 letter words don't need to be loaded into RAM. When playing a video, it only needs the next X seconds decoded to run smoothly, you don't need the entire decoded file in RAM at once. When displaying lots of large images, you don't need to decode the ones not on screen. Special programs which need that speed have high requirements to run (Photoshop and AutoCAD both need 2-8gb of RAM, most games need at least 4GB to run at low resolutions).

If your program isn't expecting to take up gobs of RAM, then the Virtual Memory protocol built into the OS will handle any overages quite satisfactorily. Otherwise, utilize special read protocols.