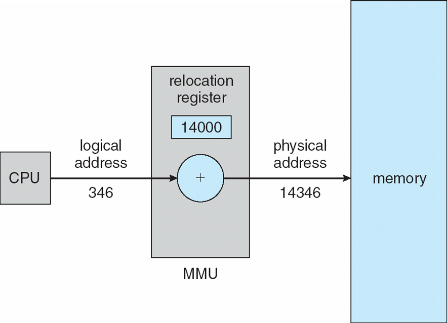
1. What are the types of storages that CPU can access?
2. Why don’t we load the process always into address 0000?
3. Why do need base and limit registers?

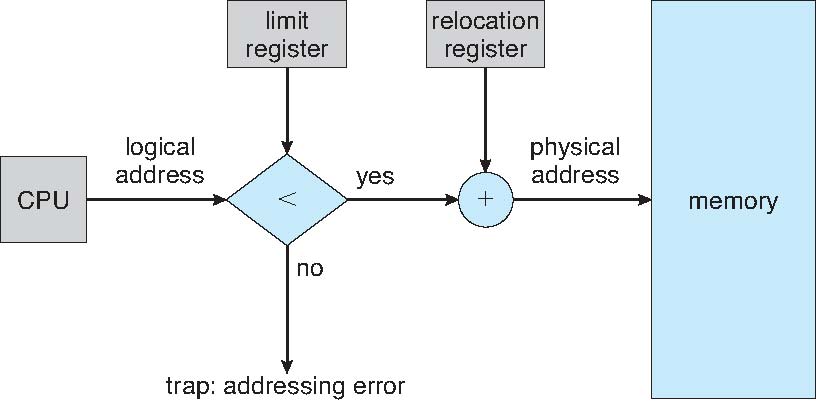
Diagram

Description automatically generated

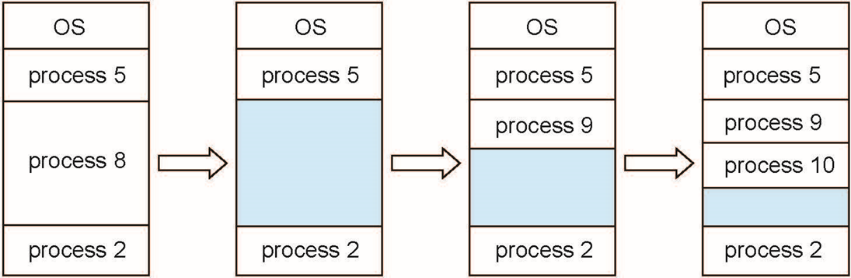
1. What are the stages of address binding (binding instructions and data to memory)?
2. Where are the logical addresses generated from?
3. Where are the physical addresses generated from?
4. At what stages they are the same and different?
5. What is MMU?
6. Dynamic loading of routines.



1. Static linking vs. dynamic linking
2. Explain the swapping. Where are the processes swapped to?
3. What is roll out and roll in?
4. Name some of the swapping versions in UNIX, Linux and Windows.
5. Does the swapped out process need to swap back in to same physical addresses?
6. A process may be waiting for an I/O operation when we want to swap that process to free up memory. What are the solutions?
7. Is swapping supported in Mobile OSs? Why? What do they do instead?
8. Contiguous allocation? Each process contained in single contiguous section of memory.



1. Fixed-sized partitions vs. variable-partitions



1. How to satisfy a request of size ***n*** from a list of free holes?
2. External fragmentation vs. internal fragmentation.
3. What is compaction? When is it possible? Problems?
4. What other two many solutions to fragmentation?
5. Segmentation example:

