- If bit=1 (page is modified), page is physically overwritten on the swap area.

vfork()

- fork() create child process by duplicating calling process.
  - Allocates separate memory space for the child process.
  - Create a new thread to execute the child process.
- If exec() is called, it release allocated segments and reallocate as per need of new program to be loaded.
- To avoid this, BSD UNIX developed vfork() system call to create new process. This should be used only while calling exec() in child.
- vfork() creates child process virtually. It doesn't duplicate parent process; rather suspends execution of parent and continue execution of child process until exec() or exit() is called (under parent thread of execution and in parent's address space itself).
- When exec() is called, then actual memory is allocated for the child process and new thread of execution is created for the child process. Hereafter child process executes independent of the parent process.
- terminal> man 2 vfork --> Historical description

Copy On Write

- Modern fork() syscall creates a logical copy of the calling process.
- Initially, child and parent both processes share the same pages in the memory.
- When one of the process try to modify contents of a page, the page is copied first; so that parent and child both will have separate physical copies of that page. This will avoid modification of a process by another process.
- This concept is known as "copy on write".
- The primary advantage of this mechanism is to speed up process creation (fork()).

Virtual memory

- Virtual memory is the memory that can be given to a process. Typically it includes RAM size (except kernel space) + Swap area.
- https://www.youtube.com/watch?v=qcBIvnQt0Bw&list=PLiwt1iVUib9s2Uo5BeYmwkDFUh70fJPxX

Assignments

1. Implement fast file copy program (assume max file size = 1 GB).

- step 1: open src file in rdonly mode.
- step 2: get size of src file (fstat() syscall)
- step 3: map src file contents to memory using mmap()
- step 4: create dest file in rdwr mode.
- step 5: make size of dest file, same as size of src file using ftruncate()
- step 6: map dest file contents to memory using mmap() -- MAP_SHARED.
- step 7: copy src file to dest file using memcpy()
- step 8: close src and dest files.