Problem Set 2
Due Sunday January 29, 11:59pm

1.Add mmap, unmap to xv6. UNIX mmap(addr, length, rw, fd, offset) is a system call to map the contents of the open file fd, at a virtual address, addr, with read-only or read-write permission. A user program can then use normal instructions, e.g., *addr = 5, to read/write data to the file. Unmap(addr, length) removes the file mapped to addr from the virtual address space and copies any modifications back to the file. Process exit implies unmap.

You may assume/require that: the address to put the file is always beyond the in-use portion of the address space; it does not overlap any other mmap region; only one process at a time maps a given file; the process never forks; the entire file is loaded into memory at the time mmap is called, etc. Hint: what happens if a user process passes a pointer to an mmap region as an argument to a system call?

2. Add demand paging to mmap. Modify your solution in part 1 to allow the program to restart without the contents of the file being into memory. Instead, as the program references pages in the file, the pages of the file are brought in as needed.

Hint: as a first step, see what happens when you run a test program that references a missing mmap location.

Hint: what happens when a process makes a system call with a pointer to a page that is not resident in memory?