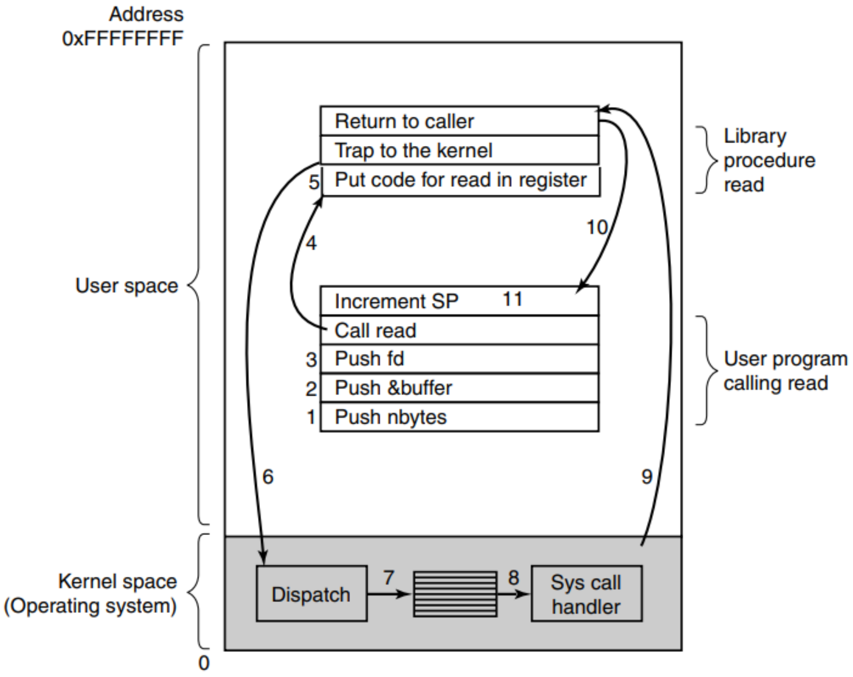
How to Trace Read call in Minix :-

Trace the read system call in Minix. You need to list all the functions and files involves when a program calls the read() system call step by step and explain what happens in each function

Ans:

read(fd, buffer, nbytes);

After a read system call is invoked the calling program pushes the parameters to the stack in reverse. The first parameter FD (file descriptor) and nbytes (no: of bytes to read) are called by value and buffer is passed by call by reference. Then the actual library call puts the code in a place possibly a register where the os expects it. Now a trap instruction is executed which changes the user mode to kernel mode so that kernel handles the system call. Before entering the kernel the process may have to go to the process manger. The dispatcher in kernel mode examines the call by its number and transmits to the correct system call handler. After sys call handler completes running then trap instruction is executed and mode changes back to user level. Trace is important for the monitoring to system calls. Tracing is the use monitoring and it will be beneficial for the see how system calls goes in user mode to kernel mode.



After successful execution the call returns number of bytes read.

write(fd, buffer, nbytes)

For write() system call the procedure is almost similar with the read call. The only difference is the data in buffer is written into the file whereas for read() system call the the data is read from the file to the buffer. After successful execution the write() system call returns number bytes written into the file.

In Minix there is some specific command to execute or find out trace is called ptrace is :-

Ptrace - process trace

Ptrace is use in minix to monitoring in the system