CO21BTECH11002
Aayush Kumar

How System Call works:

To create a system call we need to assign a call id to the system call. When a user enters the name of a system call, the console reads the name of the function and tries to find the corresponding function number. If the console is not able to find the function number then an error message is generated and execution of the call fails. If the function number is successfully found, the system call is executed.
System calls execute in kernel mode since the program needs to access low level resources, hence, system calls have more privileges. During the system call, the processor switches from user mode to kernel mode and when the call execution is finished, it switches back to user mode.

Observations and Reasons:

1. On declaring a large global array, the number of valid entries remains the same. This is because the global variables have different storage space and declaration of global variables does not affect the page table entries for the current process.

2. On declaring the large array as a local variable, the number of valid entries increases. This is because local variables take some stack space and affect the page table entries for the current process.

3. On repeating the process multiple times:
    a. The number of valid entries remains the same. This is because the amount of memory allocated to the process remains the same on each execution. The size of the page table is determined by the virtual memory allocated to the process which remains the same across multiple executions.

b. On each execution, the physical address changes. This is because of address space randomization mechanism which is a security feature implemented by many OS including xv6. The virtual address remains the same since there is no such randomization mechanism for virtual address.