



Kernel and its “Controlling Role”

- “The one program **running at all times** on the computer” is the **kernel***. Everything else is either a system program or an application program ?
- Even though without a kernel an application **can not** achieve results

the kernel is **not always running**

Note: On a **single processor system**, **either** the kernel **or** an application program **or** a non-kernel system program is running.

* Silberschatz et al.: “OS Concepts”, 7th Edition



Insecure System Call

- Consider a hypothetical system call `zeroFill()`, which fills a user buffer with zeroes
`zeroFill(char* buffer, int bufferSize)`
- The following kernel implementation of `zeroFill` contains a security flaw. *What is the vulnerability, and how would you fix it?*

```
void zeroFill(char* buffer, int bufferSize){  
    for (int i=0; i < bufferSize; i++){  
        buffer[i] = 0;  
    }  
}
```



Solution + Follow-up Question

- The user buffer pointer `buffer` is **untrusted**, and could point anywhere.
- In particular, it could point into the kernel address space. This could lead to a system crash.
- Fix: **verify** whether the pointer is a **valid user address**
- Is it a security risk to execute **the original zeroFill** function in user mode?
- No. User mode code cannot access the kernel's address space. If it tries, HW raises an exception.