Process creation on Unix using `fork`

```
int fork()
```

1. Creates and initializes a new PCB
2. Creates a new address space
3. Initializes the address space with a copy of the entire contents of the address space of the parent (with one exception)
4. Initializes the kernel resources to point to the resources used by parent (e.g., open files)
5. Create a kernel thread associated with this process and place that thread onto the ready queue

Why `fork` and `exec`?

`fork` is very useful when the child…

- is cooperating with the parent
- relies upon the parent's data to accomplish its task

➡ Simple interface