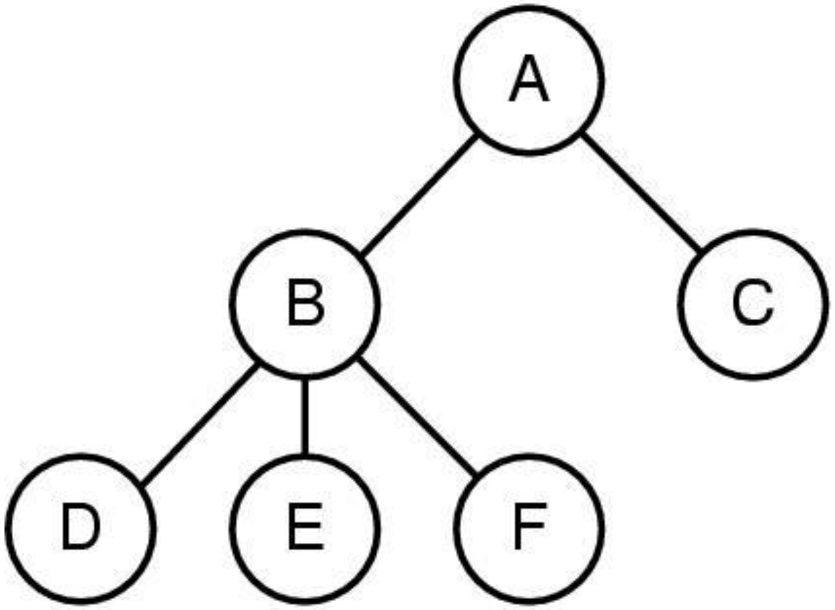
Some System Calls For Process Management

Process management

Call	Description
pid = fork( )	Create a child process identical to the parent
pid = waitpid(pid, &statloc, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process' core image
exit(status)	Terminate process execution and return status

process란?

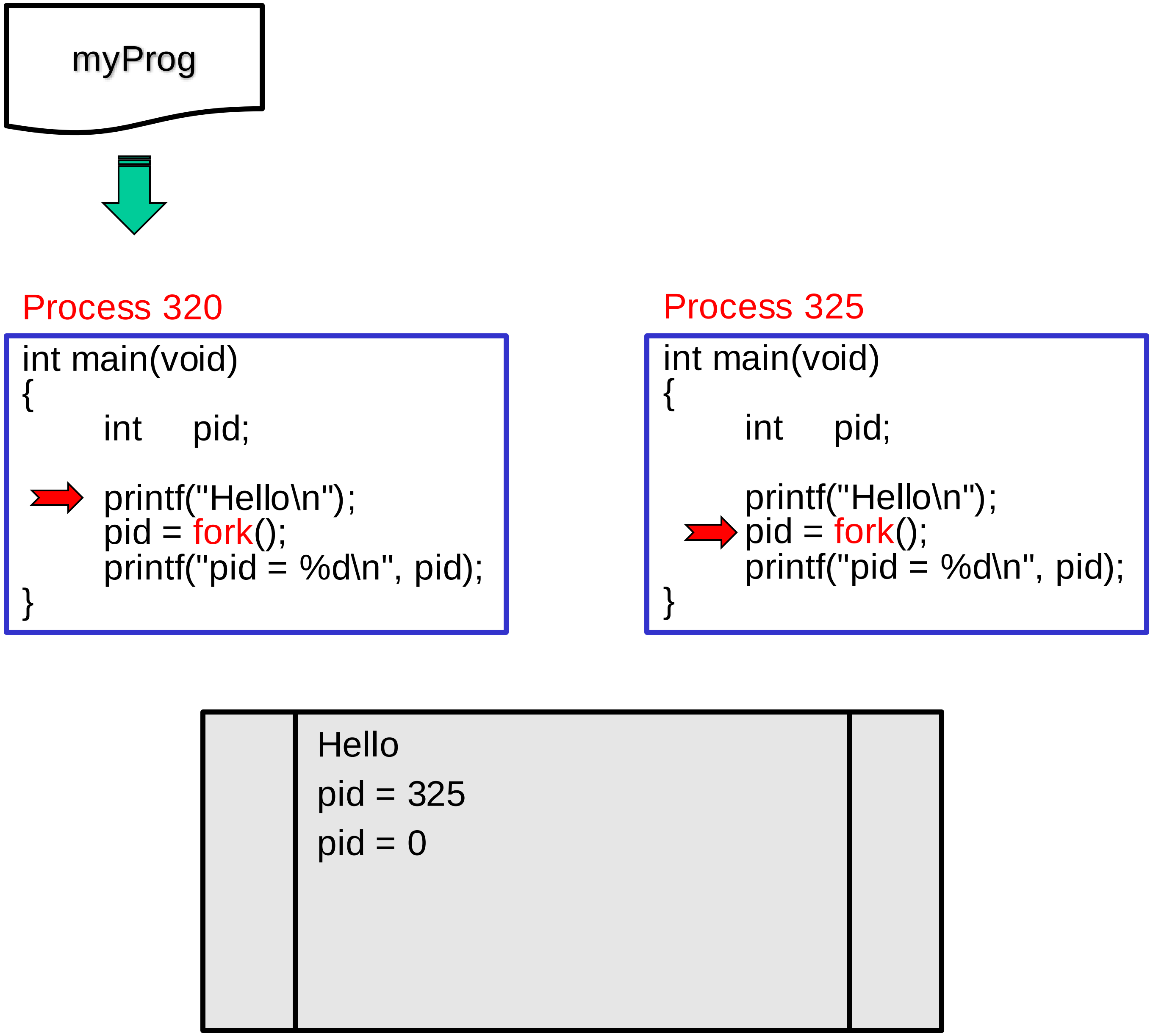


System Calls (1)

- **간단한 toy shell:**

```
while (TRUE) {                             /* repeat forever */
   type_prompt( );                         /* display prompt */
   read_command (command, parameters)      /* input from terminal */

   if (fork() != 0) {                      /* fork off child process */
      /* Parent code */
      waitpid( -1, &status, 0);            /* wait for child to exit */
   } else {
      /* Child code */
      execve (command, parameters, 0);     /* execute command */
   }
}
```



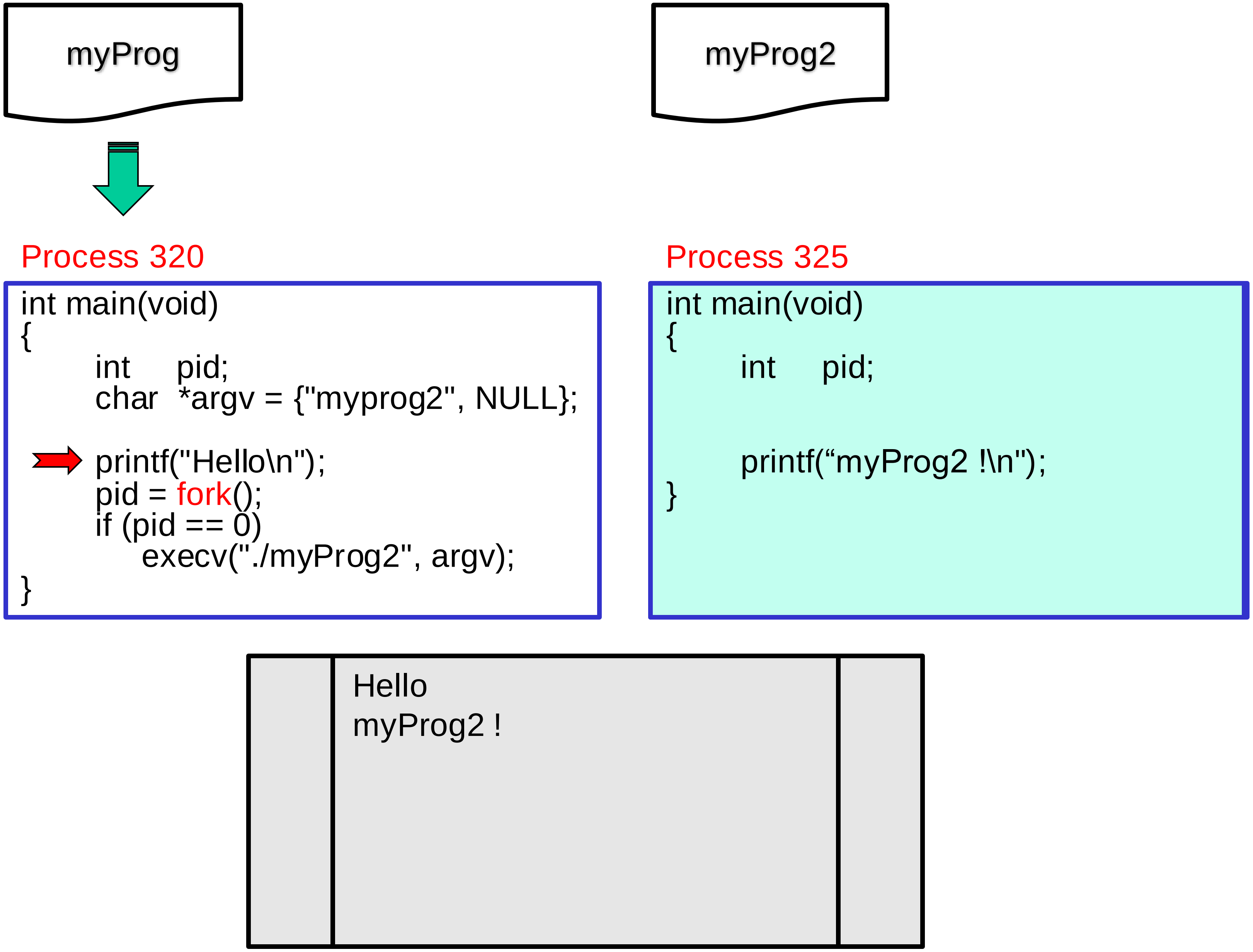
```
int main(void)
{
        int     pid;

        printf("Hello\n");
        pid = fork();
        printf("pid = %d\n", pid);
}
```

```
int main(void)
{
        int     pid;

        printf("Hello\n");
        pid = fork();
        printf("pid = %d\n", pid);
}
```

```
Hello
pid = 325
pid = 0
```



```
int main(void)
{
        int     pid;
        char  *argv = {"myprog2", NULL};

        printf("Hello\n");
        pid = fork();
        if (pid == 0)
            execv("./myProg2", argv);
}
```

```
int main(void)
{
        int     pid;


        printf("myProg2 !\n");
}
```

```
Hello
myProg2 !
```