Process oddities and endities

David Morgan

zombie processes

…the argument passed to exit by the child process...is stored in the kernel until the parent of the process retrieves the value with the wait system call. If the parent is not currently waiting, the exit value remains in the kernel until the parent calls wait to be notified of the termination of the child.... A process that has died but still has an uncollected exit value is called a zombie process. Many modern versions of ps list these processes as defunct.

Understanding Unix/Linux Programming, Bruce Molay
orphan/adopted processes

If a process exits before its children do, the children continue to run. They do not become orphans, instead, they are made children of the init process, sort of like wards of the state

Understanding Unix/Linux Programming, Bruce Molay
adopted

```c
#include <stdio.h>

int main(void)
{
  int result;
  int i,j,k;
  result = fork();
  if (result == 0) {
    printf("Child: my PID is %d\n", getpid());
    sleep(10);
    exit(0);
  }
  else
    printf("Parent: my PID is %d. I'm going to die in 10 sec, orphaning my child.\n", getpid());
  sleep(10); exit(0);
}
```

Child: my PID is 7192. I'm going to die in 10 sec, orphaning my child.