

Process oddities and endities

David Morgan

© David Morgan 2003-05

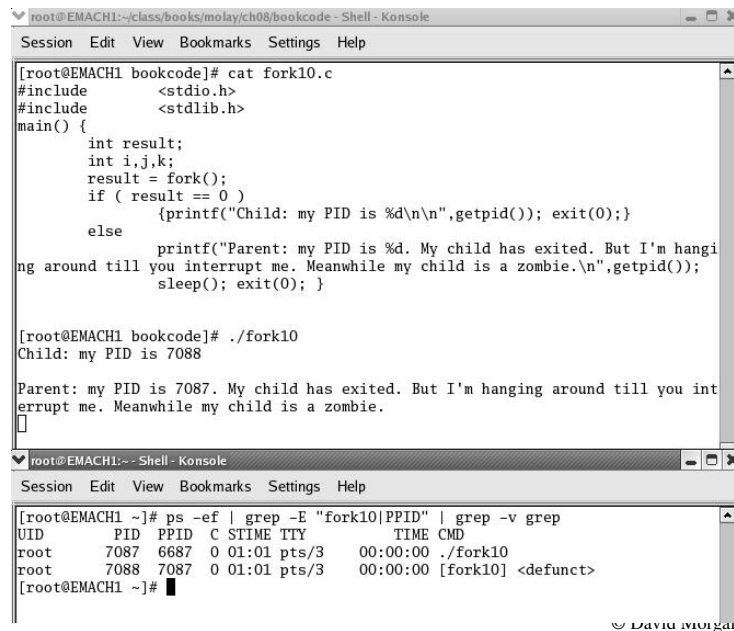
zombie processes

...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` `tytsem` 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

© David Morgan 2003-05

zombie



```
root@EMACH1:~/class/books/molay/ch08/bookcode - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@EMACH1 bookcode]# cat fork10.c
#include <stdio.h>
#include <stdlib.h>
main() {
    int result;
    int i,j,k;
    result = fork();
    if ( result == 0 )
        {printf("Child: my PID is %d\n\n",getpid()); exit(0);}
    else
        printf("Parent: my PID is %d. My child has exited. But I'm hanging around till you interrupt me. Meanwhile my child is a zombie.\n",getpid());
        sleep(); exit(0); }

[root@EMACH1 bookcode]# ./fork10
Child: my PID is 7088

Parent: my PID is 7087. My child has exited. But I'm hanging around till you interrupt me. Meanwhile my child is a zombie.
^

root@EMACH1:~/ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@EMACH1 ~]# ps -ef | grep -E "fork10|PPID" | grep -v grep
UID      PID  PPID  C  STIME TTY          TIME CMD
root     7087  6687  0  01:01 pts/3      00:00:00 ./fork10
root     7088  7087  0  01:01 pts/3      00:00:00 [fork10] <defunct>
[root@EMACH1 ~]#
```

© David Morgan 2003-05

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

© David Morgan 2003-05

adpoted

```
root@EMACH1:~/class/books/molay/ch08/bookcode - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@EMACH1 bookcode]# cat fork11.c
#include <stdio.h>
#include <stdlib.h>
main() {
    int result;
    int i,j,k;
    result = fork();
    if ( result == 0 )
        {printf("Child: my PID is %d\n\n",getpid()); sleep();}
    else
        printf("Parent: my PID is %d. I'm going to die in 10 sec, orphaning my child.\n",getpid());
        sleep(10); exit(0); }

[root@EMACH1 bookcode]# ./fork11
Child: my PID is 7193

Parent: my PID is 7192. I'm going to die in 10 sec, orphaning my child.
[root@EMACH1 bookcode]# █

root@EMACH1:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

[root@EMACH1 ~]# ps -ef | grep -E "fork11|PPID" | grep -v grep
UID      PID  PPID  C  STIME TTY          TIME CMD
root     7192  6687  0  01:05 pts/3      00:00:00 ./fork11
root     7193  7192  0  01:05 pts/3      00:00:00 ./fork11
[root@EMACH1 ~]#
[root@EMACH1 ~]# ps -ef | grep -E "fork11|PPID" | grep -v grep
UID      PID  PPID  C  STIME TTY          TIME CMD
root     7193    1  0  01:05 pts/3      00:00:00 ./fork11
[root@EMACH1 ~]# █
```

© David Morgan 2003-05