Tuesday, May 6. 2014

Less known Solaris features: pwait

This is a nifty small tool that i'm using quite often in scripts that stop something and do some tasks afterwards and i don't want to hassle around with the contract file system. It's not a cool feature, but it's useful and relatively less known. An example: As i wrote long ago, you should never use kill -9 because often the normal kill is intercepted by the application and it starts to do some clean up tasks first before really stopping the process. So just because kill has returned, it doesn't imply that the process is away. How do you wait for process to disappear?

With pwait you have a small tool for this task. Let's assume you have tool running in the background

jmoekamp@kusanagi:~$ nohup sleep 500 &
[1] 7046

When you start pwait with the PID as the parameter, it just waits and doesn't return to the shell.

jmoekamp@kusanagi:~$ pwait 7046

Now we kill the process

jmoekamp@kusanagi:~$ kill 7046
jmoekamp@kusanagi:~$

As soon as the process is away, the pwait will return to shell.

jmoekamp@kusanagi:~$ pwait 7046
jmoekamp@kusanagi:~$

Posted by Joerg Moellenkamp in English, Solaris at 05:35

If you use a shell, what is the advantage of pwait over wait?

I think that pwait is only usefull if you execute it from a no-shell (it hasn't the built-in wait command).
Anonymous on May 6 2014, 10:03

Blindness by routine. You are obviously right ... i'm using pwait for quite a time .. so i never thought of the shell build-in
Anonymous on May 7 2014, 14:18

Thinking why pwait was created if wait existed, it must becase of this:

wait only can wait by child processes.

pwait can wait by other processes that don't need to be child process.

In the freebsd man page of pwait, it indicates a gotcha about pwait:
pwait is not a substitute for the wait(1) builtin as it will not clean up
any zombies or state in the parent process.
Anonymous on May 8 2014, 01:14