

# Assignment 1 - Question 3

Author - Ashutosh Sahu - 11840260

**Note** - In case the code/text formatting looks odd in this PDF, you can view the original Dropbox Paper [here](#).

## A. Which process adopts the child process when its parent is killed?

It is the **systemd** (one of the daemon processes) process that adopts a process after its parent process gets killed. There are multiple **systemd** processes running in the system. One of those **systemd** processes is the default init process.

According to [Wikipedia](#),

“Like the init **daemon**, **systemd** is a **daemon** that manages other **daemons**, which, including **systemd** itself, are background processes. **systemd** is the first **daemon** to start during booting and the last **daemon** to terminate during shutdown.”

According to [linode](#),

“**systemd** is the default init system for the major Linux distributions”

However, it is not the init process that adopts the a process of which that parent gets killed.

It is always the nearest ancestor **systemd** process in the `pstree` which adopts the process.

You can verify all the facts written by first running multiple `bash` in the same terminal and then running the following command:

```
pstree -p | grep "systemd\|bash"
```

This should give you the following output:

```
systemd(1)---ModemManager(976)---{ModemManager}(994)
      |---systemd(1170)---(sd-pam)(1173)
      |---systemd(1568)---(sd-pam)(1569)
```

```

| -gnome-terminal-(3650)---bash(491
5)---bash(4981)---bash(5004)---bash(5025)---bash(5046)---bash
(5067)

| | -bash(1164
5)---watch(11679)

| | -bash(1557
9)---grep(5708)

|-systemd-journal(351)
|-systemd-logind(1001)
|-systemd-resolve(924)
|-systemd-timesyn(926)---{systemd-timesyn}(970)
|-systemd-udev(374)

```

Here you can see that 4th line is the thread of all the `bashes` that we ran in the same terminal.

Here, the init process (parent of all the processes) is a **systemd** process whose PID is 1.

Also, in the above observation, the nearest ancestor **systemd** process to the `bash` es is process no. 1568. Now try killing one of the `bash` whose PID is between 4981 to 5046 (for ex. 5004). You will observe from the `watch` command that process 1568 (nearest ancestor **systemd**) has adopted the child process! (Refer to Fig.)

```

Every 0.1s: ps -l -t 2 -f --forest ashutosh-OMEN: Sun Mar 15 13:42:46 2020
F S UID      PID  PPID  C  PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
0 S ashutosh  4915  3650  0   80   0 - 2719 do_wai 12:41 pts/2    00:00:00 bash
0 S ashutosh  4981  4915  0   80   0 - 2719 poll_s 12:41 pts/2    00:00:00 \_ bash
0 S ashutosh  5025  1568  0   80   0 - 2719 do_wai 12:41 pts/2    00:00:00 bash
0 S ashutosh  5046  5025  0   80   0 - 2719 do_wai 12:41 pts/2    00:00:00 \_ bash
0 S ashutosh  5067  5046  0   80   0 - 2719 poll_s 12:41 pts/2    00:00:00 \_ bash

```

*Fig: Terminal after you have killed process 5004*

Another thing to observe in the WCHAN column in figure is that both processes with ID 4981 and 5067 now have access to the Standard Input (as both of them have `poll_s` entry). The reason is that 5067 already had access to the Standard Input before we killed 5004. After killing, process 4981 (parent of 5004) also requests to access the Standard Input as it was waiting for it's child to exit.

## B. Is this a special process?

**systemd** is a special process. According to [linode](#),

“**systemd** is a Linux initialization system and service manager that includes features like on-demand starting of daemons, mount and automount point maintenance, snapshot support, and processes tracking using Linux control groups. **systemd** provides a logging daemon and other tools and utilities to help with common system administration tasks.”

Hence, this process is capable of creating other daemon processes and adopting processes when their parent gets killed.

## C. Is it always the same process?

As said in part A, it is the nearest **systemd** process in the `pstree` that is responsible for adopting processes whose parent gets killed. However, when the system boots, each of the **systemd** processes has different types of child processes to handle.

Hence, every time a `bash` is spawned in a terminal, it is always the same **systemd** that has an ancestral relationship to it. Hence when its (`bash` 's) parent terminates it is the same **systemd** that adopts the process every time. If we are somehow able to change this ancestral relationship, we might be able to change the adopting process.

Hence, the simple answer to this question is,

“until the ancestral relationship between `bash` and **systemd** in the `pstree` remains same, the adopting process will remain the same”.

## D. Explain your observation

All the observations are explained in part A.