Exercises on Processes and Signals.

1. Parent and Child Processes

1. Run:

```
sleep 300 &
echo $!
```

   - What does **`$!`** print? How can you use it with **`ps`** to find that process's parent PID (`PPID`)?

2. There is a script in your login dir: `spawn.sh`:

```
#!/bin/bash
echo "Parent PID: $$"
sleep 10000 &
```

   - Run it like this: **`./spawn.sh&`**
   - The & will send the process to the background
   - Question: What is the PPID of the `sleep` process?

2. Background Processes

Goal: Practice starting and managing background jobs.

1. Start a command:

```
sleep 20000 &
```

   - Use **`jobs`** to check the background job status.
   - Bring it to the foreground and stop it with **`Ctrl+Z`**, then send it to background again with `bg`.

- **`fg sleep`**
- **`CTLR+Z`**
- **`bg`**

2. Try `ps -fu $USER` — can you find this process?

4. Sending Signals (9, 15, 1)

1. Start `sleep 9999 &` and note its PID.
2. Send signal **15 (TERM)**:

```
kill -15 <PID>
```

   - What happens? Check with `ps`.

3. Start the process again, then send signal **9 (KILL)**:

```
kill -9 <PID>
```

   - What's the difference compared to signal 15?

4. Start another process and send signal **1 (HUP)**:

```
kill -1 <PID>
```

   - Which kinds of programs respond to a HUP signal, and how?