Muhammad Shabir Khan

Module: R0: The Missing Semester

Section: Command Line Environment **Task:** 02

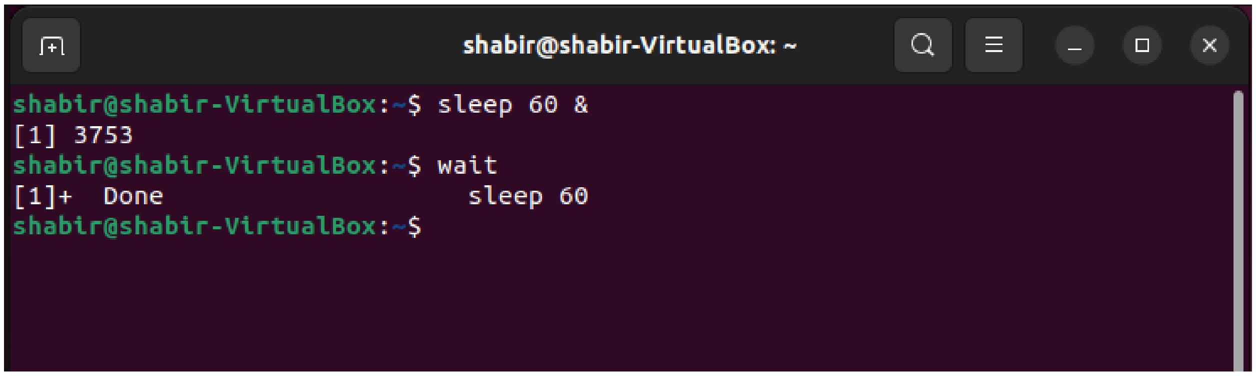
Task:

Say you don't want to start a process until another completes. How would you go about it? In this exercise, our limiting process will always be sleep 60 &. One way to achieve this is to use the wait command. Try launching the sleep command and having an ls wait until the background process finishes.

However, this strategy will fail if we start in a different bash session since wait only works for child processes. One feature we did not discuss in the notes is that the kill command's exit status will be zero on success and nonzero otherwise. kill -0 does not send a signal but will give a nonzero exit status if the process does not exist. Write a bash function called pidwait that takes a pid and waits until the given process completes. You should use sleep to avoid wasting CPU unnecessarily.

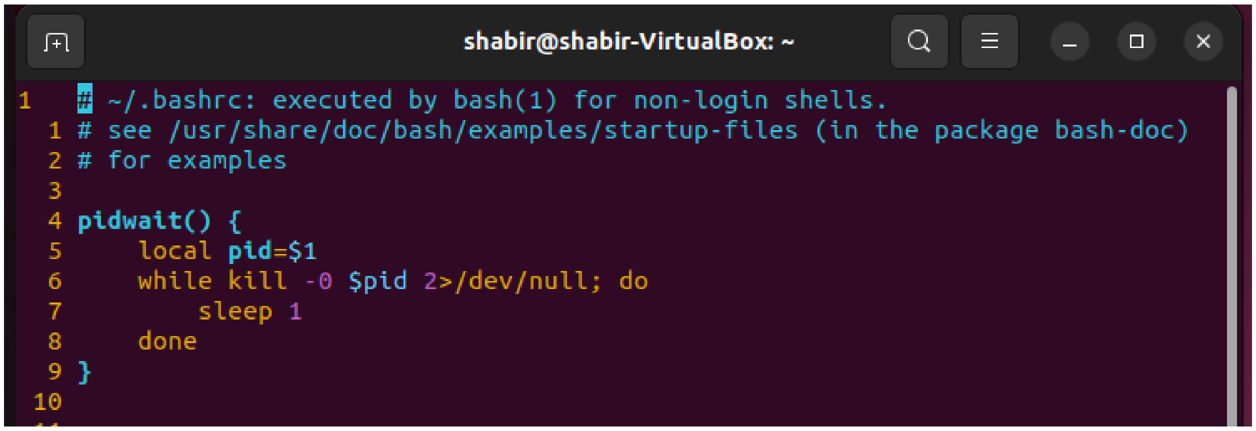
Explanation:

Run the ***'sleep 60 &'*** command to run this command in the background and then ***'wait'*** command to waiting for the command to complete. The ***wait*** command will wait for all background processes to complete. In this case, it will wait for the **sleep 60** command process to finish. The output of the terminal is shown below,



```
shabir@shabir-VirtualBox:~$ sleep 60 &
[1] 3753
shabir@shabir-VirtualBox:~$ wait
[1]+  Done                    sleep 60
shabir@shabir-VirtualBox:~$
```

As mentioned, the wait command only works for child processes within the same session. For this, I create a function called **pidwait** that takes a PID as an argument and waits until the given process completes. The pidwait function was created in the ~/.bashrc file. The pidwait function script is shown below,

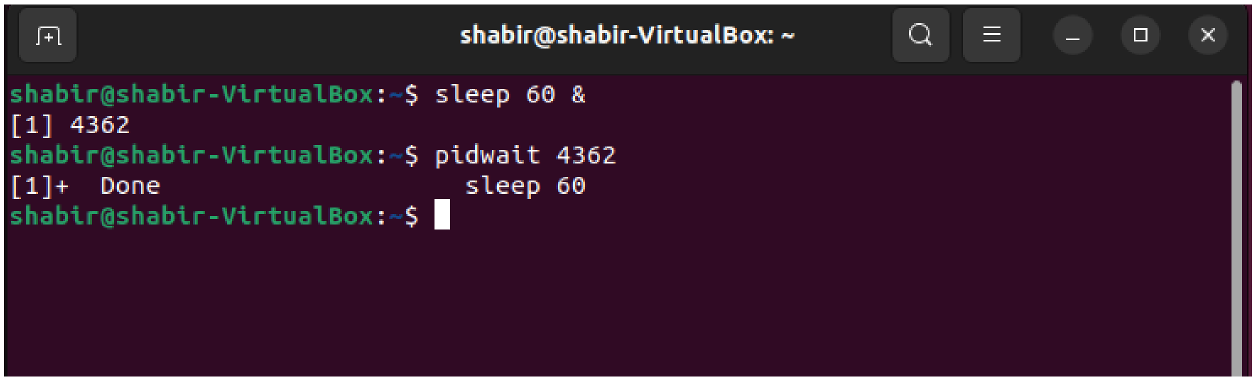


```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

pidwait() {
    local pid=$1
    while kill -0 $pid 2>/dev/null; do
        sleep 1
    done
}

```

The functionality of the pidwait function is shown below in the terminal output,



```
shabir@shabir-VirtualBox:~$ sleep 60 &
[1] 4362
shabir@shabir-VirtualBox:~$ pidwait 4362
[1]+  Done                    sleep 60
shabir@shabir-VirtualBox:~$ 
```