

Acceptance criteria for the lib/daemon Subplot library

The Subplot project

2021-05-09 05:54

Contents

1 Introduction	1
2 Daemon is started and terminated	1
3 Daemon takes a while to open its port	2
4 Daemon never opens the intended port	2
5 Daemon stdout and stderr are retrievable	2

1 Introduction

The Subplot¹ library `daemon` for Python provides scenario steps and their implementations for running a background process and terminating at the end of the scenario.

This document explains the acceptance criteria for the library and how they're verified. It uses the steps and functions from the `lib/daemon` library. The scenarios all have the same structure: run a command, then examine the exit code, verify the process is running.

2 Daemon is started and terminated

This scenario starts a background process, verifies it's started, and verifies it's terminated after the scenario ends.

given there is no `"/bin/sleep 12765"` process
when I start `"/bin/sleep 12765"` as a background process as `sleepyhead`
then a process `"/bin/sleep 12765"` is running

¹<https://subplot.liw.fi/>

when I stop background process **sleepyhead**
then there is no **"/bin/sleep 12765"** process

3 Daemon takes a while to open its port

This scenario verifies that if the background process never starts listening on its port, the daemon library handles that correctly. We do this by using netcat² to start a dummy daemon, after a short delay. The lib/daemon code will wait for netcat to open its port, by connecting to the port. It then closes the port, which causes netcat to terminate.

given a daemon helper shell script **slow-start-daemon.sh**
given there is no **"slow-start-daemon.sh"** process
when I try to start **"/slow-start-daemon.sh"** as **slow-daemon**, on port **8888**
when I stop background process **slow-daemon**
then there is no **"slow-start-daemon.sh"** process

File: **slow-start-daemon.sh**

```
1 #!/bin/bash
2
3 set -euo pipefail
4
5 sleep 2
6 netcat -l 8888 > /dev/null
7 echo OK
```

4 Daemon never opens the intended port

This scenario verifies that if the background process never starts listening on its port, the daemon library handles that correctly.

given there is no **"/bin/sleep 12765"** process
when I try to start **"/bin/sleep 12765"** as **sleepyhead**, on port **8888**
then starting daemon fails with **"ConnectionRefusedError"**
then a process **"/bin/sleep 12765"** is running
when I stop background process **sleepyhead**
then there is no **"/bin/sleep 12765"** process

5 Daemon stdout and stderr are retrievable

Sometimes it's useful for the step functions to be able to retrieve the stdout or stderr of the daemon, after it's started, or even after it's terminated. This scenario verifies that lib/daemon can do that.

²<https://en.wikipedia.org/wiki/Netcat>

given a daemon helper shell script **chatty-daemon.sh**
given there is no "**chatty-daemon**" process
when I start **./chatty-daemon.sh** as a background process as **chatty-daemon**
when daemon **chatty-daemon** has produced output
when I stop background process **chatty-daemon**
then there is no "**chatty-daemon**" process
then daemon **chatty-daemon** stdout is "**hi there\n**"
then daemon **chatty-daemon** stderr is "**hola\n**"

We make for the daemon to exit, to work around a race condition: if the test program retrieves the daemon's output too fast, it may not have had time to produce it yet.

File: **chatty-daemon.sh**

```
1  #!/bin/bash
2
3  set -euo pipefail
4
5  trap 'exit 0' TERM
6
7  echo hola 1>&2
8  echo hi there
```