DAEMON

基本概念

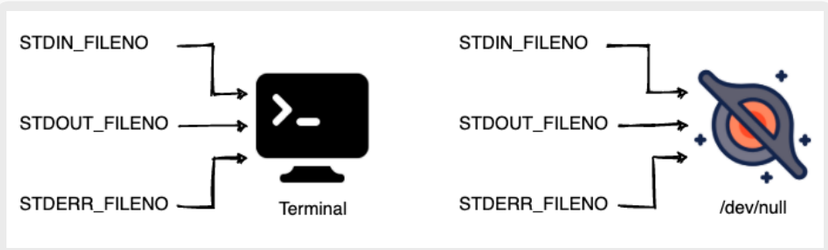
- daemon 进程又被称之为守护进程，其生命周期可以认为和 OS 运行的周期保持一致，即 OS 创建时即创建守护进程，OS 运行结束则守护进程随之结束
- 比较常见的守护进程包括 cron、sshd，以及用户可能会自定义安装的 redis-server、nginx、MySQL 等
- 特点
  - 生命周期很长。通常，一个 daemon 会在系统启动的时候被创建并一直运行直至系统被关闭
  - **它在后台运行并且不拥有控制终端。控制终端的缺失确保了内核永远不会为 daemon 自动生成任何任务控制信号以及终端相关的信号(如 SIGINT、SIGTSTP 和 SIGHUP)**
- **根据上面这一个特点，一些守护进程会将 SIGINT 和 SIGHUP 作为一种通知信号，若守护进程接收到了这两个信号，那么必然是用户或者是服务本身发出的。例如 nginx，当我们执行 nginx -s reload 热更新配置文件时，实际上是向 nginx 的 master 进程发送了一个 SIGHUP 信号**
- 此外，Linux 上也有一些特定的 daemon 会作为线程执行，比如 pdflush，则会定期将 buffer 中数据刷新至磁盘

创建一个 daemon

- 在 Linux 上创建一个守护进程基本上就是照着模板一路 copy 下来，其流程本身比较固定，几乎不可能删减其中的步骤
- 流程
  - ① **执行一个 fork()，创建出子进程之后父进程退出，子进程继续执行**
    - 如果守护进程是从 shell 中创建的，那么守护进程应该让出终端控制，不能占用
    - 子进程一定不会作为一个进程组的首进程，才有可能释放与当前终端的所有关联
  - ② 子进程调用 setsid() 开启一个新的会话，并释放与终端之间所有的关联
    - 一般情况下，终端下直接运行的进程当终端被关闭时，运行的进程也会退出执行
    - 调用成功后终端是否被关闭将不会影响到子进程的运行
  - ③ 清除进程的 umask ，即调用 umask(0)，以确保 daemon 有创建目录或文件的权限
  - ④ 关闭从父进程继承而来的已经打开的文件描述符，并将标准输入、标准输出以及标准错误重定向至 /dev/null 虚设备
    - 因为标准输入、标准输出以及标准错误和终端相关，而我们的守护进程和任何终端都不产生联系，留着这 3 个文件描述符完全没有必要
    - 但是我们又不能直接关闭掉这 3 个文件描述符，万一程序在某个地方执行了 printf()，那么就会出现错误
- 可以看到，守护进程其实就是一个孤儿进程，父进程在 fork() 之后就自行退出了。因此，守护进程一定会被 init 进程所接管。同时我们可以看到，孤儿进程本身其实是无害的

程序内的输入、输出重定向

- 在程序内，如果我们想要将 STDIN_FILENO、STDOUT_FILENO 以及 STDERR_FILENO 这 3 个描述符重定向到 /dev/null 的话，就需要借助 dup2() 这一系统调用
- 如果我们把文件描述符看作是指针的话，那么 dup2(A, B) 的作用就是把 A 指向的东西给到 B，让 B 的指向和 A 的指向相同
- int fd = open("/dev/null", O_RDWR);
dup2(fd, STDIN_FILENO);
dup2(fd, STDOUT_FILENO);
dup2(fd, STDERR_FILENO);



- **简单的来说，就是让 STDIN_FILENO 指向现在 fd 指向的那个设备文件**

分支主题 3