

runit

这是一道Shell Good Coder考试题

`runit` 是一个应用 (application) 启动管理工具。通过 `Procfile` 文件启动相应的进程。

1 试题描述

1.1 Procfile

`Procfile` 包含进程名字和启动进程的命令，用 `:` 分隔。如：

```
web: python -m SimpleHTTPServer $PORT
date: date $DATE_FORMAT
web_2: while true ; do nc -l $PORT < index.html
```

- 进程名字可以包含：字母, 数字, 下划线
- `Procfile` 中不可以写后台命令
- `runit` 将这些命令运行在后台
- `runit` 默认使用当前路径下的 `Procfile` 文件
- 如果多次使用 `$PORT` 变量，则值递增。如第一个 `$PORT` 的值是 `8080`，则第二个 `$PORT` 的值为 `8081`，如果不在 `.env` 中设置 `$PORT` 变量的值，则自动设置默认值为 `8080`

1.2 环境变量

如果当前目录下存在 `.env` 文件，则从其中读取环境变量。这个文件由 键/值对 构成。如：

```
PORT=8080
DATE_FORMAT=' +%Y-%m-%d | %H:%M:%S '
```

1.3 程序执行

- `runit` 启动 `Procfile` 中的所有进程
- `runit -f procfile -e env_file`
- `runit -c` 检查 `Procfile`, `env_file` 文件格式的正确性
- `runit -h` 打印帮助

1.4 其他要求

- `Procfile` 和 `.env` 文件中可存在 `#` 注释
- `usage` 内容第一行如下，其余内容自由发挥

```
Usage: runit [-c] [-f procfile|Procfile] [-e envfile|.env]
```

- 日志打印到屏幕，格式如下，不同的进程的日志输出不同的颜色(web, date, web_2分别是不同的颜色)

```
11:39:45 web      | python -m SimpleHTTPServer 8080 started with pid 781
11:39:45 date      | date +%Y-%m-%d|%H:%M:%S started with pid 790
11:39:45 web_2     | nc -l 8081 < index.html started with pid 801
```

- `runit` 按照 `Procfile` 的描述启动进程，例如web就是启动一个前台进程（非后台进程或daemon）
- `runit` 可以接收SIGINT和SIGTERM信号，杀掉已启动的进程。确保 `runit` 在子进程运行结束后才退出。（如果进程可瞬间完成或是后台进程，则这个功能不起作用）
- 除 `grep` 外不允许使用其他外部命令，如：`sed`, `awk`, `ps`, `bc`
- 遵循[shell编程规范](#)

2 参考和帮助(hints)

- 如果仍不明白题目，请看[这个帮助视频](#)
- ["Colorizing" Scripts](#)
- [Parameter Substitution](#)
- [String Operations](#)
- [wait](#)
- [getopts](#)
- [trap](#)
- [POSIX Character Classes](#)
- [printf](#)
- 列出所有builtin: `help`。验证一个命令是否是builtin: `type -a <command>`
- 将一个string作为代码执行: `bash -c "${command}"`

3 如何做题

```
## 1. 获取代码
git clone http://git.n.xiaomi.com/liuzhuo1/runit.git

## 2. 编写代码
vim runit

## 3. 运行测试，确保所有测试用例通过
make test
```



```
## 4. 运行一个典型的Procfile
make run
```



```
## 5. 提交代码评审

## 仅提交runit即可
```

4. 结果

1. 跑过测试用例（必要条件）

```
make test
```

```
sylzd@ubuntu:~/myworkspace/mi_homework/runit.git$ sudo make test
bash bin/roundup test/test_args.sh test/test_envfile.sh test/test_procfile.sh
Runit: command line arguments
  it_displays_usage_with_error_args: [PASS]
  it_displays_usage_with_hyphen_and_h: [PASS]
  it_verifies_invalid_procfile_and_exit_with_err: [PASS]
  it_verifies_invalid_envfile_and_exit_with_err: [PASS]
Runit: envfile
  it_ignores_comments_in_env_file: [PASS]
  it_assigns_a_default_port_number: [PASS]
  it_allows_overriding_the_port_number: [PASS]
  it_can_increase_port_number: [PASS]
Runit: procfile
  it_runs_simple_processes: [PASS]
  it_passes_environment_variables_to_processes: [PASS]
  it_ignores_comments_in_proc_file: [PASS]
=====
Tests: 11 | Passed: 11 | Failed: 0
sylzd@ubuntu:~/myworkspace/mi_homework/runit.git$ date
Wed Jan 24 20:33:48 PST 2018
```

1. 确认程序启动后的行为

```
make run
## 程序屏幕输出及颜色
## Ctrl + C 之后进程是否被干净的杀掉
```

```
sylzd@ubuntu:~/myworkspace/mi_homework/runit.git$ make run
rm -rf dist
bash bin/roundup test/test_args.sh test/test_envfile.sh test/test_procfile.sh
Runit: command line arguments
  it_displays_usage_with_error_args: [PASS]
  it_displays_usage_with_hyphen_and_h: [PASS]
  it_verifies_invalid_procfile_and_exit_with_err: [PASS]
  it_verifies_invalid_envfile_and_exit_with_err: [PASS]
Runit: envfile
  it_ignores_comments_in_env_file: [PASS]
  it_assigns_a_default_port_number: [PASS]
  it_allows_overriding_the_port_number: [PASS]
  it_can_increase_port_number: [PASS]
Runit: procfile
  it_runs_simple_processes: [PASS]
  it_passes_environment_variables_to_processes: [PASS]
  it_ignores_comments_in_proc_file: [PASS]
=====
Tests: 11 | Passed: 11 | Failed: 0
mkdir dist
cp -r prod/* dist/
cp -r prod/.env dist/
cp runit dist/
cd dist ; ./runit
20:39:01 web | python -m SimpleHTTPServer 8080 started with pid 87818
20:39:01 date | date $DATE_FORMAT started with pid 87826
Wed Jan 24 20:39:01 PST 2018
Serving HTTP on 0.0.0.0 port 8080 ...
20:39:01 web_2 | nc -l 8081 < index.html started with pid 87834
```

```
20:33:01 web_2 | ... ^Cmake: *** wait: No child processes. Stop.  
make: *** Waiting for unfinished jobs....  
make: *** wait: No child processes. Stop.
```

1. 阅读代码

```
## 代码风格  
## 命名准确性  
## 改进性建议（更好的写法，潜在的bug和风险）
```

• 代码风格规范

shellcheck检查结果

```
sylzd@ubuntu:~/myworkspace/mi_homework/runit.git$ shellcheck runit  
sylzd@ubuntu:~/myworkspace/mi_homework/runit.git$  
sylzd@ubuntu:~/myworkspace/mi_homework/runit.git$
```

• 命名

下划线命名，全局为大写，局部为小写，没有无意义及单字符命名

• 改进性建议

1. 重复代码

```
function verify() {  
    verify_file ${ENV_FILE}  
    if [[ $? -gt 0 ]]; then  
        echo -e "\033[31m File ${ENV_FILE} is not exist \033[0m"  
        exit 1  
    fi  
    verify_file ${PROC_FILE}  
    if [[ $? -gt 0 ]]; then  
        echo -e "\033[31m File ${PROC_FILE} is not exist \033[0m"  
        exit 1  
    fi  
}
```

此处有重复代码，但是对((verify_file \${ENV_FILE} + verify_file \${PROC_FILE} > 1))判断，测试会出错。

2. 冗余逻辑

感觉main里面逻辑有点冗余，可以进一步简化

3. 变量定义

```
declare proc_name  
declare proc_cmd  
proc_name=$(echo "${line}" | cut -d : -f 1)  
proc_cmd=$(echo "${line}" | cut -d : -f 2-)
```

这里变量的声明定义，为了过掉shellcheck中的 SC2155: Declare and assign separately to avoid masking return values. 错误而声明，感觉多写了两行没甚作用。