

## # 要求

---

实现自己的 `popen(const char *cmd, const char *mode)` 和 `pclose(FILE *fp)`

`popen`有以下几个细节

1. `pipe`打开管道
2. `fork`创建子进程
3. `exec`在子进程中实现 `/bin/sh -c "cmd"`
4. `dup()`复制标准IO流到管道

`pclose`有以下细节

1. 使用`wait`等待子进程关闭
2. 子进程与`popen`的返回值绑定，或者叫有关

## # 实现

---

```
→ popen ./popen
>>> ===== TEST popen =====
# /*****
#   > FILE NAME: TEST.PY
#   > AUTHOR: RACLE
#   > MAIL: RACLERAY@QQ.COM
#   > CREATED TIME:
# *****/
#CODING = UTF - 8

IMPORT COLLECTIONS

DEF TEST(ARG):
    """TEST.

    :PARAM ARG:
    """
    QUEUE = COLLECTIONS.DEQUE()
    QUEUE.APPEND(1)
    QUEUE.APPEND(ARG)
    PRINT("HELLO")
    RETURN

>>> ===== TEST myopen =====
# /*****
#   > FILE NAME: TEST.PY
#   > AUTHOR: RACLE
#   > MAIL: RACLERAY@QQ.COM
#   > CREATED TIME:
# *****/
#CODING = UTF - 8

IMPORT COLLECTIONS

DEF TEST(ARG):
    """TEST.
```

```
    QUEUE.APPEND(ARG)
    PRINT("HELLO")
    RETURN
>>> ===== TEST myopen =====
# /*****
#   > FILE NAME: TEST.PY
#   > AUTHOR: RACLE
#   > MAIL: RACLERAY@QQ.COM
#   > CREATED TIME:
# *****/
#CODING = UTF - 8

IMPORT COLLECTIONS

DEF TEST(ARG):
    """TEST.

    :PARAM ARG:
    """
    QUEUE = COLLECTIONS.DEQUE()
    QUEUE.APPEND(1)
    QUEUE.APPEND(ARG)
    PRINT("HELLO")
    RETURN
→ popen
```

childpid 可以使用一个数组储存，linux库实现就是数组。

popen的 r 和 w 的动作接受者是 cmd 的进程，r 表示 cmd 从管道读，所以管道保持写端。w 表示 cmd 往管道中写，管道保持读端。

