

## 参考

[https://shellscript.sunone.me/signal\\_and\\_trap.html](https://shellscript.sunone.me/signal_and_trap.html)

<https://genzouw.com/entry/2021/03/02/080000/2381/>

## 概要

trap コマンドで finally のような後処理を実行する。

trap コマンドは送出されたシグナルを捕捉し、あらかじめ指定されていた処理を実行するコマンドである

終了コードは trap で補足する前の終了コードになる。

```
trap finally EXIT
finally() {
  echo finally
  echo ここに後処理を書く
}
```

## サンプル

run\_finally.sh

```
#!/bin/sh

trap finally EXIT
# trap finally EXIT INT

RUN_COMMAND=${1}
FINALLY_COMMAND=${2}

finally() {
  # echo finally
  # echo "${FINALLY_COMMAND}"
  eval ${FINALLY_COMMAND}
}
# echo run
# echo "${RUN_COMMAND}"
eval ${RUN_COMMAND}
```

## 呼び出し

```
sh run_finally.sh "echo test" "echo 後処理 "
```

## 「;( セミコロソ)」との違い

コマンドの処理結果 (リターンコード) に関わらずコマンドを実行するだけなら

```
echo test; echo 後処理
```

のようにセミコロソで繋がればよいが、最後に実行したコマンドのリターンコードしか受け取れない

## 例

```
echo test; hoge; echo finally
echo $?
0
```

hoge コマンドがないのでエラーになりつつ、finally が表示されるがリターンコードは 0 になる

上記 shell の場合は、第一引数のリターンコードを取得できる

```
sh run_finally.sh "echo test; hoge" "echo finally"
echo $?
127
```

また、finally でエラーになってもリターンコードは書き換えられない

```
sh run_finally.sh "echo test" "echo finally;hoge"
echo $?
0
```