

ADVANCED BASH

Thursday, 27 May 2021

4:29 PM

CO-PROCESSES

Using Coprocesses

- A coprocess is a background process where your shell gets file descriptors for the process's stdin and stdout.

Implemented with pipe

Using Coprocesses

We need a script that is a filter.

```
#!/bin/bash
while
  read line
do
  echo $line | tr "ABC" "abc"
done
```


Using Coprocesses

```
coproc ./mycoproc.sh
```

- `echo BANANA >&"${COPROC[1]}"`
- `cat <&"${COPROC[0]}"`

NAMING A COPROC

```
coproc my { ./mycoproc.sh; }
```

- `echo BANANA >&"${my[1]}"`
- `cat <&"${my[0]}"`

DEBUGGING SCRIPTS:

Debugging Scripts

```
$ bash prog
```

Run prog; don't need execute permission

```
$ bash -x prog
```

Echo commands after processing; can also do `set -x` or

set +x inside of script

\$ bash -n prog

Do not execute commands, check for syntax errors only

Debugging Scripts

- `set -u`
Reports usage of an unset variable
- Lots of echo statements
- `tee`
`cmd | tee log.file | ...`

