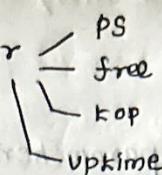


Bash (Scripting language)

1. Debug computer



Run a single command (do all this)

`#!/bin/bash`

`echo "Starting at: $(date)"`

`echo`

`echo "UPTIME"`

`uptime`

`Echo`

`echo "FREE"`

`free`

`echo`

`echo "WHO"`

`who`

`Echo`

`echo "Finishing at: $(date)"`

call command!

Just print.

`echo` → newline!

Same line → use semicolons

`echo "FREE"; free; echo`

Using variables & globs

1. loops

2. conditions etc...

Bash

1. Environment variables (\$)

2.

Local environ variables

>> `Var1=hello` → No spaces

>> `echo $Var1`

Global

export!

Globs: (characters in bash allows to create a list of files)

★, ? Common globs

* Use: Create sequences of file name - use as parameters! (in our scripts)

(with the file) ★ → all

→ echo *.py (all python files)

→ echo c* (start with file)

→ echo * (all)

? — only one character

echo ????.py (5 letters alone)

In python: glob module

conditional execution in bash:

* Python: if/else/elif

* bash: based on exit status of commands : do something!

/etc/hosts has 127.0.0.1 entry

0 — Success

if grep "127.0.0.1" /etc/hosts ; then

echo "Found"

else

echo "Not found"

fi

→ end block!

Task

* command evaluates the conditions received & exits with zero when they're true & with one when they're false

if test -n "\$PATH"; then echo "Empty Path"; fi

empty?

if [-n "\$PATH"]; then echo "Empty"; fi

Space

Anything inside []
evaluated

loops

while loop:

#!/bin/bash

n=1

while [\$n -le 5]; do

echo "Iteration number \$n"

((n+=1))

done

less than/equal to

bash script: Common (retries using loop: until a command succeeds)!
Network/resources locked!

\$1 → 1st command line argument
\$2 → 2nd command line argument

Python: sys.argv[1]

bash script

```
import sys
import random
value = random.randint(0,3)
print ('Returning: ' + str(value))
sys.exit(value)
```

0,1,2

until program succeeds (max 5 attempts)

#!/bin/bash

n=0

Command=\$1

while ! \$Command && [\$n -le 5]; do

sleep \$n

((n=n+1))

echo "Retry # \$n"

done;

./retry.sh randPy.py

0,1,2

* Why sleep? → wait for resources to be released!

Returning 1

Retry #1

Returning: 2

Retry #2

Returning: 0

→ Succeeded!

For loops

* for fruit in peach orange apple; do

 echo 'I like \$fruit';

done;

* all .HTML → .html

bash script

* basename: extract only name without extension

* #!/bin/bash

for file in *.HTML ; do

 name = \$(basename "\$file" .HTML)

 → double quotes: say file name with spaces

 mv file "\$file" "\$name.html"

done;

Rule 1: don't run directly - take test!

echo mv file "\$file" "\$name.html"



If fine! Then run!



challenge

$\rightarrow \text{tail } /var/log/syslog \longrightarrow \text{date/time comp-name PID event}$

$\text{tail } /var/log/syslog | \text{cut -d ', -f5-}$

Space as limiter → field number 5 and everything after it!
(Keep only process event message)

$\rightarrow \text{cut -d ', -f5- } /var/log/syslog | \text{sort} | \text{uniq -c} | \text{sort -nr} | \text{head}$

mask repetitive IPnes?

do this in every log file

Oct 23 08:01:55 hari-Pt syslog: [] syslog
 ↓ ↓ ↓
 1 2 3

-f3- (from time to end)

#! /bin/bash

```
for logfile in /var/log/*log; do
    echo "processing: $logfile"
    cut -d ',' -f5- $logfile | sort | uniq -c | sort -nr | head -5
done
```

when to use python & when bash

1. Bash - automate ~~to~~ work on all files!

lot of text/data processing commands

Combined with '|' pipeline - powerful!

1. Convert each word's first letter to uppercase

↳ Bash - complex (indexing)

↳ Python - easy

Readability

Bash: files & system Commands (Simple) → Linux (Windows - absent - power shell)

Python: hard to understand - go! → Python (OS!)

↳ Python std. lib / libraries.

which ever feel comfortable!

```
for file in $(ls ~/data | grep 'Jane')  
do  
    echo $file  
done
```

Process:

1. understand problem statement ↗ P
↘ P

2. Research (lib, modules) + doc ↗ P

3. Planning (datatypes, operation, all pieces - come together)

4. Writing (code + test)!

problem

* Servers (one) : Service: TICKY (used by teams) - Internal ticketing system
↓
syslog (success, error)

* Automation: use syslog → generate reports

* May 27 11:46:40 ubuntu.local TICKY= INFO: created ticket
[#1234] (username)

success

ticket

username

ticket number

1. Reports

- Rank errors (how many + freq)
- Statistics (users: info messages, error messages)

↓
Sort by username!

* generate webpages: Served by webserver!

* CSV - htm - html.py!

Planning & Research

* Regex (regex101.com)

* dictionaries — Sorted!

o/p: CSV file →

last step (bash: only calling commands moving files)!

May 27 11:45:40 ubuntu.local ticket: INFO: created ticket [#1234] (username)
Jun 1 11:06:48 ubuntu.local ticket: ERROR: connection to dB failed (username)

error →

- Timeout (retrieve info)
- ticket modified while updating
- dB Conn failed
- add info to a closed ticket
- permission denied while closing ticket
- ticket doesn't exist.

1. ranking of errors gen by sys (all errors: how many times - sort)!
2. No. of usage - statistics for service - users - how many info, error (username - sort).

error_message.csv

user_statistics.csv

nodes: and quads
Root: 4 nodes
edges: 6 edges

Troubleshoot & debugging techniques

1. Why not working? (crash, stuck, wrong, faster, less memory, send)

2. Even you can't those code!

Debugging:

1. get back to work as soon as possible (never repeat) - root cause

Troubleshooting: Identify, analyze, solve problems (fixing problems running in system)

Debug: Identify, analyze, removing bugs (actual code)

fixing system running the app

fixing code that system running.

Tools:

1. Wireshark] network connection

2. tcpdump

3 ps, top, free (resources)

4. Strace (system calls)

5. Ltrace (library calls of a software)

Debuggers:

1. line by line - inspect variables, interrupt (when specific condn met)

what could be failing - how to find it - how to solve - prevent it.

Integration tests - check integrity

Problem solving steps:

1. getting info (use, when, consequences - docs)

2. Reproduce case (how, when appears)

3. Root cause

4. perform remediation (long term - prevent in future).

5. document (what we do)! - invaluable.

Shutdown - overheat

Clean up fan : short

reduce dust : long
collection

Silently crashing application:

1. fails to open (say) → error? (details)

→ new version (not working)

reproduce,

strace (system calls!)

(System calls: calls (apps) make to running kernel!)

Strace -o file-name (store)
lsof file-name

doc: If directory doesn't exist, fails to start! (fix in next version)

It doesn't work when after

1. what is the issue (ticket / find ourselves)

Q: what you tried?
How (steps)?
expected?
Your o/p?

Consider simpler explanations / solutions
unless complex solution is the way!

when I tried to login, page keeps loading - never shows page I want!

1. reproduce! (same happens) - problem: our side!

2. specific site / entire? (specific)

3. what's going on that server (linux(ssh), top command,

overloaded!

A: System CPU wasted on waiting for OS (System calls),
backup system running!

Load average:

how much time processor is busy in a given minute!

(whole minute (can't be > amount of processors))

1 → whole minute (can't be > amount of processors)

A cores (40 - really really high)

5. Stop backup (kill/-stop)!

Immediate remediation: done (most of the time)

Long term:

1: Find actual issue → too much I/O happening in desc?

Processor

→ Backup System (reduce priority to access disc)

Too much network

(ifstop)

- concurrent traffic

* Problem: product category column doesn't exist (generating report)

* Reproduce:

app not working: user side problem - network routing, old config after clashing with new version, faulty hardware, permission (system)

1. Read logs /var/log/syslog (syslog)

• X Session-errors (users)

mac: /Library/logs

windows: Event viewer tool

2. unable to reach server/file format/permission
say: no error message / simple (no internal error)

3. other people - experience that error?

4. same error in same user logs in to different computer

5. config directory moved away? (clash b/w old config...)
(ask send that directory) - analyze crafting!

* Is config OK with previous version?

* Report bug, colleague, forums!

Root cause

1. overload server: backup server running (root cause: BW saturated, hard drive faulty, disc transfer low)

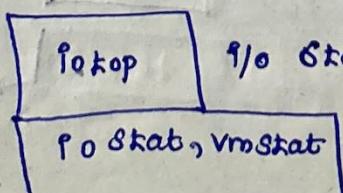
Actual root cause: (hypothesis → test → make sure)

1. Inspiration: gather info, look in to info!

2. check hypothesis in test env. (instead of production environment)

3. test server, test data... (extra safety)

* Problem with hardware: can't be reproduced (wait for server, use secondary server - migrate!)



`rsync -bwlimit`

`Trickle-limit BW`

→ Compress level reduction

- * Compression algo - too much computing power

→ nice (reduce priority)

Intermittent issues

Occasionally (random)

1. Fail to Suspend

2. webServer - unexpectedly stopped replying

hard to reproduce!

3. Corrupted contents

1. Get more involved (more info) - log
2. more logging info - identify! (when bug happens)

When issue triggers

1. Load
2. processes running at same time
3. usage of network.

Heisenberg - observer effect

(observing a phenomenon alters it)
observer goes away - annoying

memory management, network connection,
handle files.

- * Reboot → clean slate! (delete cache, temp, ...)

1. understand error messages

2. Add logging info

3. Generate new ideas for possible influences

↑

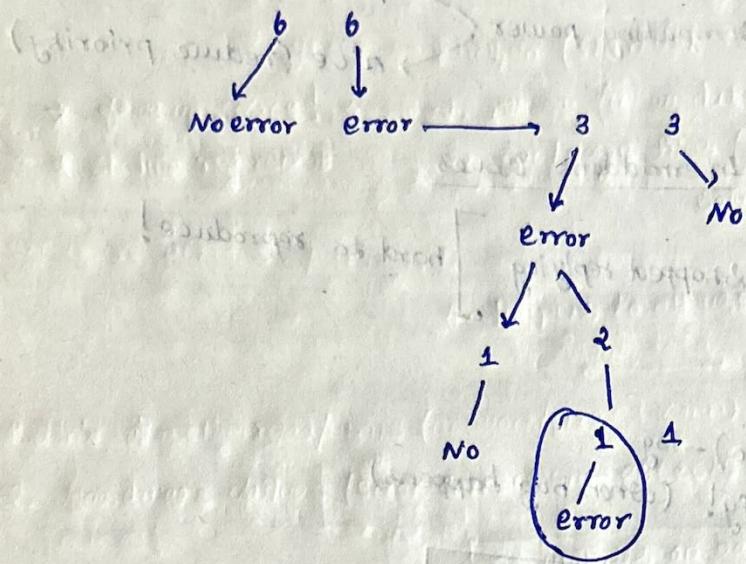
Intermittently failing script

1. dates - how formatted (possible error)

Root cause: date format wrong!

Binary Search

1. 10 config files - which causing bug



eg: which browser extension - causing crash (divide half) - - -

discover plugin, which change? (Keep cutting)! - bisect

bisect



* 2 points in time (git history)

* Repeatedly lets try the code at the middle point

* doesn't need to be git repo!

Find invalid data

head -15 file

tail -15 file

100 entries

head -50 | •/ex.py

head -25 | .ex.py

Slowness

* keep opening tabs: run out of memory (slow)

* close applications

why?

1. takes more CPU time (than getting?)

Identifying bottleneck - CPU time / reading / waiting / moving

resources exhausted? — CPU
memory
GPU

1. top (CPU time) / memory
2. netop — more network / I/O usage
3. iftop

macOS: Activity monitor

Windows: resource, perf. mon

How Computer uses?

Access data: Internal memory (fast), RAM (little less speed),
disc (slower), Network (worst)
Establish conn
low BW!

* make sure: store it on disc, read afterwards
process memory (too slow)
cache (faster!)

* Web proxy: form of cache.

Run out of RAM

Super slow (1. clear cache (not necessary) → uses swap from disc
2. swap data currently not in use (for efficiency))

lot of time - swapping

1. too many open app

2. Amount memory small (add RAM)...

3. memory leak! (close files properly).

Slowness

why slow?

* Elimination process → easy ones first

1. when (Startup) — too many startup apps.

* free up space! ← 2. reboot (when data ↑ without deleting)

* schedule reg reboot 3. too large files (really slow) — bug!

Solution: 1. reduce file size
2. keep size low.

4. Happens to all users? (Or for some?)

1. don't directly read/write (network/shared conn) each time
2. Hardware failure (error correction — affects performance) — keep an eye
3. Clean of malicious software!, extensions.

Slow web Server

1. Navigate & load page (slow for us)!

ab (apache benchmark tool)

ab -n 500 abc.example.com

500 requests

go to webserver instance

1) top → ffmpeg processes (high load)

(videotranscoding)

above no. of processors — not good!

1. change priorities

0 to 19 (lower-higher prio.)

nice
renice

for pid in \$(pidof ffmpeg); do renice 19 \$pid; done

slow (||) running — not good —

ffmpeg — CPU intensive

2) find what triggers?

ps aux → all running processes

* locate static /001.webm → return directory where located!

* cd /srv/deploy-videos

* many file (*.sh) → which triggers?

→ grep ffmpeg *

(couple of files): deploy.sh

vim deploy.sh

daemonize → parallelize! (daemon (run as))

only one at a time: delete daemonize!

(don't use -S)!

not done

Kill currently running files:

→ kill all -STOP (STOP doesn't kill completely)

→ To resume: CONT signal can be sent!

for pid in \$(ps aux | grep ffmpeg);

do

do while kpu -cont \$pid; do sleep 1; done

done

No command to wait until process ends

Solution: while (run as long as pid exists) / random sleep.

fix our code

efficient code:

* clearer - what it should? (try to make it faster: only when it's not fast enough)

* less bugs - first priority!

* optimize - only need? (less work - faster!)

Profiler

- * measures resources used by code!
- * gprof - c programs
- * c-profile - python

each function - how many times called? Space, time.

* expensive actions: parsing, iteration in a list, network conn.

*

Right data

1. List, Tuple, dictionary

↓
Iterate!
(ArrayList)

→ hash!
(HashMap)

- 1) Access all elements by position: use list (only once/two)
- 2) lookup based on key - use dictionary (bunch of operations)

Expensive loops

1. multiply by times executed! (unnecessary + cut!)
2. Discard: when not needed further!

local results

1. Local cache (parsing large file - readable)
2. Local copy of file from network!

How often update

once per day → Employees has in a dept

Units sold

* right now? (how up-to date) — short lived

monitor health
Stock level
username exists?

Validate

→ if needed recache!

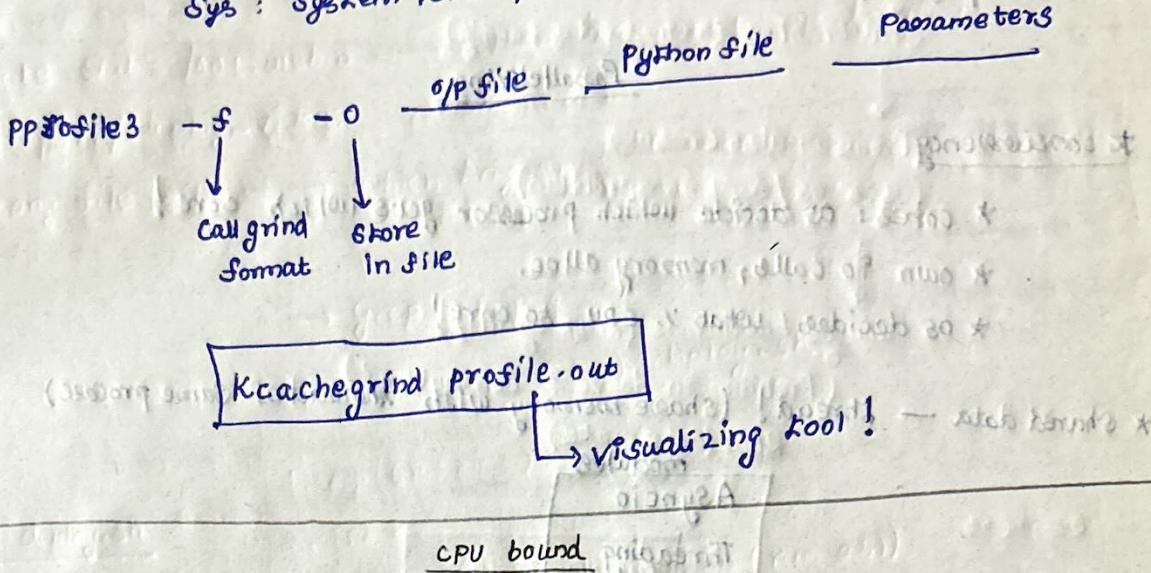
Say: if modification time changed,
recache

* 1st execution alone cache

→ take 1 min (cache 1min)

→ not worth!

Real : actual time to execute Command (wall clock time)
user: operations in user space
sys : system level operations



* Bottlenecked by CPU (I/O)

psutil

```
import psutil  
psutil.cpu_percent()  
psutil.disk_io_counters()
```

rsync

* rsync - remote sync - transfers & synchronizes files b/w comp & drives across network (delta transfer algorithm)

rsync [options] [source dir] [dest]

-v - verbose

-n

-a

-r

-b

-z

Copy / Sync locally (files)

rsync -zvh — —

directory locally rsync -zvh — —

// recursively ← rsync -zvh — —

multiprocessing: Python

```
#!/usr/bin/env python3  
import subprocess  
from multiprocessing import Pool  
  
def run(task):  
    print(task)  
    subprocess.run(["rsync", "-varv", task, dest])
```

src =

dest =

`dir_tasks = [os.path.join(src, x) for x in os.listdir(src)]`

`P = Pool(len(dir_tasks))`

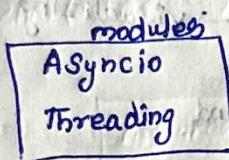
`P.map(run, dir_tasks)`

Parallelizing

* Concurrency

- * Cores: OS decides which processor gets which core!
- * own IO calls, memory alloc.
- * OS decides what Y, CPU to core!

* Shared data — Thread! (share memory with thread of same process)



want to use separate cores (processors): Use (split) code in to separate processors!

* Too much parallelism — waste in switching! (Right balance)

* Slowly growing in complexity

* Database (SQLite) Compared to CSV (large file)

↓ popular

database servers (separate)

↓ popular

added cache service

↓ load balancer!

↓ virtual machines

↓ user base growing

monitor

Complex slow systems

Backup

e-commerce

Billing System!

web

Fulfillment System!

DB Servers

Servers (users)

external users

1. waiting on network call (disk I/O) - Indexes

2. DB (Indexes)! - cache, distributing

3. CPU Saturation - Code improvements, dynamic (Cache), distribute (many requ.)

Using threads

Executor: In charge of distributing work among different workers

Futures modules (gives executors)

Threads

processes

executor = futures.ThreadPoolExecutor()

executor.submit (process - file, root, basename)

Note: loops done - when all tasks are scheduled

only completed (program) - when all tasks are completed!

Then call shutdown function on executor

processes: overhead gone more (using processes - uses more CPU)

Threads: waiting for their turn to write (System time) ↑

priorities unit ↑

Execution ↑

Crashing programs

* program terminates unexpectedly

* Device reboot - no apparent reason

* OS hangs (lose unsaved work)

* Starting threads - must close! (no resources)

No access to change source code

* Installation / configuration → on specific computer



All customers?

All invoice generation fails?

other invoices fine!



Specific Cust, Spec Computer (Suspicious)



Report generation (end of day) → crashed!



App seems to be crashing (when they use ^{specific} computer)

* move away local config - ran in default config.

* Reinstall! (crash on other apps?)

↓
Still crashing (reinst, def config)

↓
Browser also crashes
(user os / system - problem)

↓
change hardware (test again)!
(Problem on drive? Computer?)
RAM (degrade)

Overheat

Sensor data,
external devices!
(disconnect & work)!

memtest 86

check RAM health

* Check for bad sectors

* Smart tools - before fault (gives warning?)

OS issues:

* Time consuming
* Reinstall

Understand crashing app.

* System log, userlog

debug logging

Linux	macos
strace	dtks
windows	
process monitor	

↓
↳ different crash, different cause (different)

↓
↳ bad code → (not to be) returning error