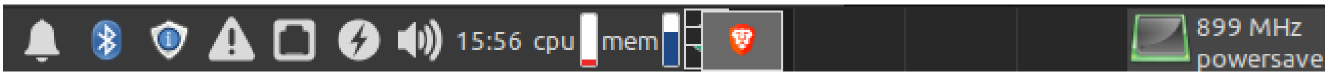
# Savage leaky programs

It’s come to my attention recently that despite a fresh install of Linux Mint, certain programs seem to leak like a basket and hang around after they’re closed too.

I’d noticed my machine freezing intermittently and adding the memory monitor panel item revealed that the system memory was filling up.

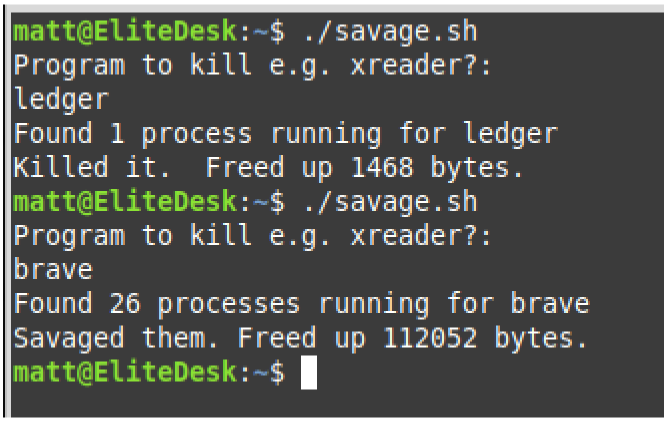


The blue mem bar fills up over time when Brave is left open. Disappointing for such an otherwise excellent Web Browser.

*xreader* and *brave* seemed to be the main culprits but since rebuilding my desktop machine, I’ve not been using many other programs apart from *ledger live* to track the value of my crypto currency portfolio while the fed prints money *ad infinitum* during the coronavirus pandemic. I digress.

Killing processes gets old really quick, so I wrote a quick’n’dirty little shell script to do it for me. Rather than killing individual processes, it savages all processes by the same name.

I shall call it *savage.sh* and share it with the world, right here. Not on github.



```
matt@EliteDesk:~$ ./savage.sh
Program to kill e.g. xreader?:
ledger
Found 1 process running for ledger
Killed it.  Freed up 1468 bytes.
matt@EliteDesk:~$ ./savage.sh
Program to kill e.g. xreader?:
brave
Found 26 processes running for brave
Savaged them. Freed up 112052 bytes.
matt@EliteDesk:~$
```

Killing all running processes for ledger and brave using savage.sh

```
#!/bin/bash
# savage.sh finds all process ID's for the specified program running under your own user account and kills them
# in order to free up system resources.  Some programs have severe memory leaks and consume vast amount of RAM and
# swap if left running over time.
#
# Usage: savage.sh
#
# Written by M. D. Bradley during Coronavirus pandemic, March 2020

#Variables
user=`whoami`
memfree=`free | grep Mem | awk {'print $4'}`
#Code
echo "Program to kill e.g. xreader?: "
read program
pidcount=`ps -fu $user | grep $program | awk {'print$program'} | wc -l`
ps -fu $user | grep $program | awk {'print$2'} | while read eachpid; do
        kill $eachpid >/dev/null 2>&1
done
memfree2=`free | grep Mem | awk {'print $4'}`
freedmem=$(( memfree2 - memfree ))
if [ $pidcount -eq 1 ]
then
        echo "Found $pidcount process running for $program"
        echo "Killed it.  Freed up $freedmem bytes."
fi
if [ $pidcount -gt 1 ]
then
        echo "Found $pidcount processes running for $program"
        echo "Savaged them. Freed up $freedmem bytes."
fi
```