

Estática de processos em bash



7 DECEMBER

Turma P1

**Da autoria de: Tomás Candeias e
Afonso Boto**

Introdução

Com este trabalho pretendemos desenvolver um script em bash onde é apresentada as estatísticas da memória usada e quantidade de input/output que todos ou uma seleção de processos estão a realizar. Dividimos a informação por colunas, sendo a primeira referente ao nome do comando, a segunda referente ao utilizador do processo, a terceira referente ao ID do processo, a quarta referente à memória utilizada, a quinta referente ao RSS, a sexta referente à linha rchar do /proc/[pid]/io, a sétima referente à linha wchar do /proc/[pid]/io, a oitava e nona é o rate de leitura e escrita considerando as duas colunas anteriores e o período introduzido e por fim a última representa a data de início da execução do processo.

Variáveis auxiliares, declarações e expressões regulares

Declaramos arrays para guardar cada o valor de cada uma das colunas da tabela, já previamente mencionadas.

Declaramos, ainda, variáveis auxiliares uteis para o desenvolvimento do projeto.

```
#Declaração de arrays
declare -a pids
declare -a comms
declare -a user
declare -a mem
declare -a rss
declare -a date
declare -a rchar
declare -a wchar
declare -a rater
declare -a ratew

#Variaveis auxiliares
regex=".*"
user_aux=0
num_of_procs=-1
count=1
flag_mem=0
flag_rss=0
flag_rateR=0
flag_rateW=0
flag_reverse='-r'
flag_reverse2=
```

Configuramos a formatação do projeto pelo LC_NUMERIC, mudamos logo no início para a diretório “proc” e definimos uma expressão regular para os valores numéricos (num_regex) e para os valores do período (sleep_period). Criamos ainda um variável n que será uma variável auxiliar para guardar os pid’s.

```
export LC_NUMERIC="en_US.UTF-8"
cd /proc
num_regex='^[0-9]+$'
sleep_period=${@: -1}
n=0
```

Mensagem de erro e verificação do período

Em `error_prompt()` definimos a mensagem de erro que iremos utilizar sempre que considerarmos inválido algum argumento. `1>&2` significa o redirecionamento do `stdout` para o `stderr`.

No `if` case da verificação do período vamos verificar se o último argumento é um número válido para representar o período (único argumento obrigatório).

```
#Mensagem de erro
function error_prompt() {
    echo "Usage: $0 [-c <reg exp>] [-p <number of processes>] [-u <user>] [-m -t -d -w -r] " 1>&2 # redireciona stdout to stderr
    exit 1
}

#Verificar se o user_aux escreveu o periodo (ultimo argumento)
if ! [[ $sleep_period =~ $num_regex ]]; then
    error_prompt
fi
```

Verificação e gestão de opções

Usamos a comando built-in `getopts` para a definição de todos os argumentos possíveis, sendo que a opção for precedida de `:` obriga a inserção de um argumento posterior, como é o caso do `-c`, do `-p` e do `-u`.

De seguida usámos `switch case` para gerir cada opção, tendo usado a variavel auxiliar `'count'` para controlar o número de argumentos válidos pela razão anteriormente referida.

```
if [[ $count -ne $# ]]; then
    error_prompt
fi
```

Usámos as default cases para abranger opções inválidas gerando assim a mensagem de erro anteriormente declarada.

O código apresentado a seguir é autoexplicativo no uso das variáveis auxiliares.

```
#Definição dos argumentos e os respectivos erros
while getopts "c:p:u:mtdwr" flag; do
  case "$flag" in
    #-c restringe nome do comando
    c)
      regex=${OPTARG}
      if [[ $regex =~ num_regex ]]; then
        error_prompt
      fi
      count=$((count + 2))
      ;;
    #-p numero de processos
    p)
      num_of_procs=${OPTARG}

      if [[ $num_of_procs -le -1 ]]; then
        error_prompt
      fi
      count=$((count + 2))
      ;;
    #-u definir user(s)
    u)
      user_aux=${OPTARG}
      count=$((count + 2))
      ;;
    #-r reverter a tabela
    r)
      flag_reverse=
      flag_reverse2='-r'
      count=$((count + 1))
      ;;
  esac
done
```

```
    #-m ordenar por memoria
    m)
      flag_mem=1
      count=$((count + 1))
      ;;
    #-t ordenar por RSS
    t)
      flag_rss=1
      count=$((count + 1))
      ;;
    #-d ordenar por Rate Read
    d)
      flag_rateR=1
      count=$((count + 1))
      ;;
    #-w ordenar por Rate Write
    w)
      flag_rateW=1
      count=$((count + 1))
      ;;
    #Opções inválidas:
    \?)
      error_prompt
      ;;
    :)
      error_prompt
      ;;
  esac
done
```

Extração de informação para arrays

Iteramos todos os ficheiros presentes no diretório atual '/proc' que contém todos os ID's verificando assim, primeiramente, se é um valor numérico, para aceder exclusivamente a esses ID's, de seguida guardamos a informação que pretendemos nos arrays anteriormente declarados, recorrendo aos comandos cat, grep e awk.

Se por alguma razão não for possível aceder ao file /proc/\$pid/io definimos por default o valor do rchar[pid] e wchar[pid] como -1.

```
for pid in `ls /proc`; do
  if [[ $pid =~ $num_regex ]]; then

    #Extrair info para arrays
    pids[$n]=$pid
    comms[$pid]=$(cat /proc/$pid/comm)
    user[$pid]=$(ls -ld /proc/$pid | awk '{print $3}')
    mem[$pid]=$(cat /proc/$pid/status | grep "VmSize" | awk '{print $2}')
    rss[$pid]=$(cat /proc/$pid/status | grep "VmRSS" | awk '{print $2}')
    io=$( cat /proc/$pid/io 2>/dev/null )

    #Adicionar info aos arrays rchar wchar
    if [[ $? -ne 0 ]];then
      rchar[$pid]=-1
      wchar[$pid]=-1
    else
      rchar[$pid]=$(echo $io | grep "rchar" | awk '{print $2}')
      wchar[$pid]=$(echo $io | grep "wchar" | awk '{print $2}')
    fi

    #Adicionar date do processo
    date[$pid]=$(ls -ld /proc/$pid | awk '{print $6,$7,$8}')

    n=$((n + 1))
  fi
done
```


Preenchimento da tabela

Recorremos ao comando sleep para “adormecer” a execução durante o período introduzido pelo utilizador, que servirá para calcular as taxas de leitura e de escrita (rater e ratew).

Iteramos todos os ficheiros presentes no diretório atual '/proc' que contém todos os ID's verificando assim, primeiramente, se é um valor numérico, para aceder exclusivamente a esses ID's. De seguida, verificamos com o "if ! [[\$comms =~ (^|[:space:])\$pid(\$|[:space:])]]" ignoramos os processos que passaram a existir durante a execução do script.

Verificamos com o quarto if case, se o user[pid] corresponde ao user inserido e se o comms[pids] está de acordo com a regex introduzida pelo user.

Posteriormente, se for possível aceder ao file presente no /proc/pid/io calculamos os rates com uso do comando bc se não ficam por default como -1.

Para terminar, preenchemos a variável display_table com os valores válidos, ou seja, tirando os pid's com valores de memória ou rss ou outro char ou rate vazio.

```
sleep $sleep_period
display_table=""

for pid in `ls /proc`; do
    if [[ $pid =~ $num_regex ]]; then
        if ! [[ $comms =~ (^|[:space:])$pid($|[:space:]) ]]; then
            if [[ "${user[$pid]}" == $user_aux || $user_aux == 0 ]] && [[ "${comms[$pid]}" =~ $regex ]]; then
                io=$(cat /proc/$pid/io 2>/dev/null)
                if [[ $? -ne 0 ]]; then
                    rater[$pid]=-1
                    ratew[$pid]=-1
                else
                    oldRate=$(echo $io | grep "rchar" | awk '{print $2}' | sed 's/[, ]//')
                    oldRatew=$(echo $io | grep "wchar" | awk '{print $2}' | sed 's/[, ]//') #Calcular os rates
                    rater[$pid]=$(echo "scale=2;$oldRate/$sleep_period" | bc)
                    ratew[$pid]=$(echo "scale=2;$oldRatew/$sleep_period" | bc)
                fi

                if ! [[ "${rchar[$pid]}" == -1 ]]; then
                    if ! [[ "${mem[$pid]}" == "" ]] || ! [[ "${rss[$pid]}" == "" ]] || ! [[ "${ratew[$pid]}" -eq xyz ]]; then
                        comms[$pid]=$(echo ${comms[$pid]} | cut -d " " -f 1) #1ª coluna
                        printf -v linha "%-20s %-20s %6d %12d %12d %12d %12.2f %12.2f %15s\n" "${comms[$pid]}" "${user[$pid]}" "$pid" "${mem[$pid]}" "${rss[$pid]}" "${rchar[$pid]}" "${wchar[$pid]}" "${rater[$pid]}" "${ratew[$pid]}" "${date[$pid]}"
                        display_table="$display_table$linha" #adicionar o texto à variável
                    fi
                fi
            fi
        fi
    fi
done
```

Impressão final da tabela

Imprimimos o cabeçalho com a formatação correta para cada coluna.

Usamos uma estrutura de if/else cases para definir todas as possibilidades diferentes para apresentar a tabela.

Quando temos a opção -m ou -t ou -d ou -w, usamos o comando sort com a opção -k para definirmos a coluna, -g porque se trata de valores numéricos e a \$flag_reverse pois se o utilizador tiver colocado a opção -r o valor da flag será '-r', caso contrário, será vazio.

Quando temos a opção -p a definir o número de processos a serem impressos, usamos o comando head com a opção -n que define o número de linhas a apresentar.

```
#Impressão da tabela
printf "%-20s %-20s %6s %12s %12s %12s %12s %12s %15s\n" "COMM" "USER" "PID" "MEM" "RSS" "READB" "WRITEB" "RATER" "RATEW" "DATE"

if [[ $num_of_procs == -1 ]]; then
  if [[ $flag_mem == 1 ]]; then
    printf "%b" "$display_table" | sort -k 4 -g $flag_reverse
  elif [[ $flag_rss == 1 ]]; then
    printf "%b" "$display_table" | sort -k 5 -g $flag_reverse
  elif [[ $flag_rateR == 1 ]]; then
    printf "%b" "$display_table" | sort -k 8 -g $flag_reverse
  elif [[ $flag_rateW == 1 ]]; then
    printf "%b" "$display_table" | sort -k 9 -g $flag_reverse
  else
    printf "%b" "$display_table" | sort -k 1 -g $flag_reverse2
  fi
else
  if [[ $flag_mem == 1 ]]; then
    printf "%b" "$display_table" | sort -k 4 -g $flag_reverse | head -n $num_of_procs
  elif [[ $flag_rss == 1 ]]; then
    printf "%b" "$display_table" | sort -k 5 -g $flag_reverse | head -n $num_of_procs
  elif [[ $flag_rateR == 1 ]]; then
    printf "%b" "$display_table" | sort -k 8 -g $flag_reverse | head -n $num_of_procs
  elif [[ $flag_rateW == 1 ]]; then
    printf "%b" "$display_table" | sort -k 9 -g $flag_reverse | head -n $num_of_procs
  else
    printf "%b" "$display_table" | sort -k 1 -g $flag_reverse2 | head -n $num_of_procs
  fi
fi
```


Testes e resultados

```
tomascandelas@Ubuntu: ~/Desktop/SO/Praticas$ ./procstat.sh 2
```

COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
at-spi2-registr	tomascandelas	1971	162828	7676	31783	31783	0.00	0.00	dez 7 23:35
at-spi-bus-laun	tomascandelas	1889	309884	9728	37555	37555	0.00	0.00	dez 7 23:35
bash	tomascandelas	5493	10884	5216	371676	371676	0.00	0.00	dez 7 23:45
cat	tomascandelas	6025	8224	520	4292	4292	0.00	0.00	dez 7 23:55
cat	tomascandelas	6026	8224	584	4424	4424	0.00	0.00	dez 7 23:55
chrome	tomascandelas	6020	963192	330216	202291324	202291324	737.50	737.50	dez 7 23:55
chrome	tomascandelas	6029	274548	64552	179137	179137	0.00	0.00	dez 7 23:55
chrome	tomascandelas	6030	274580	64692	179151	179151	0.00	0.00	dez 7 23:55
chrome	tomascandelas	6034	274580	15912	155409	155409	0.00	0.00	dez 7 23:55
chrome	tomascandelas	6058	570132	125992	272170	272170	2.50	2.50	dez 7 23:55
chrome	tomascandelas	6060	347228	86448	9093638	9093638	0.00	0.00	dez 7 23:55
chrome	tomascandelas	6061	316968	34760	251194	251194	0.00	0.00	dez 7 23:55
chrome	tomascandelas	6128	15141112	114320	119724	119724	0.00	0.00	dez 7 23:55
chrome	tomascandelas	6140	4723376	168000	1574354	1574354	0.00	0.00	dez 7 23:55
chrome	tomascandelas	6157	4712948	182092	3208370	3208370	2137.50	2137.50	dez 7 23:55
chrome	tomascandelas	6277	9082340	240836	7293737	7293737	714.50	714.50	dez 7 23:55
chrome	tomascandelas	6297	4586540	53412	17075	17075	0.00	0.00	dez 7 23:55
chrome	tomascandelas	6387	557244	58696	120884	120884	0.00	0.00	dez 7 23:56
dbus-daemon	tomascandelas	1684	8452	5496	324227	324227	0.00	0.00	dez 7 23:35
dbus-daemon	tomascandelas	1894	7372	4360	48442	48442	0.00	0.00	dez 7 23:35
dconf-service	tomascandelas	1997	156488	5872	40225	40225	0.00	0.00	dez 7 23:35
Discord	tomascandelas	4432	5305516	158820	90960294	90960294	85907.00	85907.00	dez 7 23:38
Discord	tomascandelas	4435	202324	46016	95981	95981	0.00	0.00	dez 7 23:38
Discord	tomascandelas	4436	202324	46364	95982	95982	0.00	0.00	dez 7 23:38
Discord	tomascandelas	4438	202324	7664	1	1	0.00	0.00	dez 7 23:38
Discord	tomascandelas	4465	500684	126544	295747	295747	0.00	0.00	dez 7 23:38
Discord	tomascandelas	4472	276828	68712	39820610	39820610	69.00	69.00	dez 7 23:38
Discord	tomascandelas	4539	30966584	715800	11171074	11171074	17461.50	17461.50	dez 7 23:38
Discord	tomascandelas	4566	527748	52756	193374	193374	0.00	0.00	dez 7 23:38
evolution-adre	tomascandelas	2014	673488	29720	137393	137393	0.00	0.00	dez 7 23:35
evolution-alarm	tomascandelas	2111	637080	58848	393920	393920	0.00	0.00	dez 7 23:35
evolution-calen	tomascandelas	1998	838856	30256	160960	160960	0.00	0.00	dez 7 23:35
evolution-sourc	tomascandelas	1986	390500	25696	150802	150802	0.00	0.00	dez 7 23:35
gdm-x-session	tomascandelas	1772	164016	6656	56271	56271	0.00	0.00	dez 7 23:35
gjs	tomascandelas	2028	2930508	26804	70992	70992	0.00	0.00	dez 7 23:35
gnome-session-b	tomascandelas	1798	190972	15520	5561618	5561618	0.00	0.00	dez 7 23:35
gnome-session-b	tomascandelas	1909	691712	17720	690310	690310	0.00	0.00	dez 7 23:35
gnome-session-c	tomascandelas	1902	90060	4320	24460	24460	0.00	0.00	dez 7 23:35
gnome-shell-cal	tomascandelas	1977	582068	21596	137793	137793	0.00	0.00	dez 7 23:35
gnome-shell	tomascandelas	1923	4862900	321596	697469636	697469636	1440187.00	1440187.00	dez 7 23:35
gnome-terminal	tomascandelas	5485	813068	50328	789302	789302	1137.00	1137.00	dez 7 23:45
goa-daemon	tomascandelas	1735	546312	36480	131449	131449	0.00	0.00	dez 7 23:35
goa-identity-se	tomascandelas	1746	318600	11320	337725	337725	240.00	240.00	dez 7 23:35
gsd-a11y-settin	tomascandelas	2053	313960	8876	39058	39058	0.00	0.00	dez 7 23:35
gsd-color	tomascandelas	2054	439320	28652	323850	323850	0.00	0.00	dez 7 23:35
gsd-datetime	tomascandelas	2055	374412	17276	96386	96386	0.00	0.00	dez 7 23:35
gsd-disk-utilit	tomascandelas	2097	231792	5988	25103	25103	0.00	0.00	dez 7 23:35
gsd-housekeepin	tomascandelas	2058	314196	9088	461805	461805	0.00	0.00	dez 7 23:35
gsd-keyboard	tomascandelas	2061	343964	26308	301516	301516	0.00	0.00	dez 7 23:35
gsd-media-keys	tomascandelas	2062	1022164	29004	356522	356522	0.00	0.00	dez 7 23:35

```
tomascandelas@Ubuntu: ~/Desktop/SO/Praticas$ ./procstat.sh -m -p 10 2
```

COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
Discord	tomascandelas	4539	30960384	707088	15803045	15803045	20155.50	20155.50	dez 7 23:38
code	tomascandelas	45567	19466328	223688	34217219	34217219	2844.50	2844.50	dez 8 00:02
chrome	tomascandelas	6128	15141112	114732	210976	210976	0.00	0.00	dez 7 23:55
chrome	tomascandelas	6277	9079908	233980	12983165	12983165	2143.50	2143.50	dez 7 23:55
Discord	tomascandelas	4432	5305516	157776	110200700	110200700	85489.00	85489.00	dez 7 23:38
gnome-shell	tomascandelas	1923	4897800	349096	950763484	950763484	1149036.00	1149036.00	dez 7 23:35
chrome	tomascandelas	6140	4723376	168000	1594310	1594310	0.00	0.00	dez 7 23:55
code	tomascandelas	45508	4711552	141728	5099313	5099313	0.50	0.50	dez 8 00:02
chrome	tomascandelas	6157	4710100	181500	11064094	11064094	712.50	712.50	dez 7 23:55
code	tomascandelas	45629	4617124	110616	6361160	6361160	0.00	0.00	dez 8 00:02

```
tomascandelas@Ubuntu: ~/Desktop/SO/Praticas$ ./procstat.sh -m -r -p 10 2
```

COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
dbus-daemon	tomascandelas	1894	7372	4360	51193	51193	0.00	0.00	dez 7 23:35
cat	tomascandelas	6025	8224	520	4292	4292	0.00	0.00	dez 7 23:55
cat	tomascandelas	6026	8224	584	4424	4424	0.00	0.00	dez 7 23:55
dbus-daemon	tomascandelas	1684	8452	5556	522318	522318	0.00	0.00	dez 7 23:35
procstat.sh	tomascandelas	190695	9628	3456	9177682	9177682	13222855.00	13222855.00	dez 8 00:22
nacl_helper	tomascandelas	6031	10748	4236	10204	10204	0.00	0.00	dez 7 23:55
bash	tomascandelas	5493	10884	5220	928103355	928103355	0.00	0.00	dez 7 23:45
systemd	tomascandelas	1664	19368	10568	900390312	900390312	0.00	0.00	dez 7 23:35
gnome-session-c	tomascandelas	1902	90060	4320	24460	24460	0.00	0.00	dez 7 23:35
dconf-service	tomascandelas	1997	156488	5872	40321	40321	0.00	0.00	dez 7 23:35

```
tomascandelas@Ubuntu:~/Desktop/S0/Praticas$ ./procstat.sh -m -r -p 5 -c "c.*" 2
```

COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
cat	tomascandelas	6025	8224	520	4292	4292	0.00	0.00	dez 7 23:55
cat	tomascandelas	6026	8224	584	4424	4424	0.00	0.00	dez 7 23:55
code	tomascandelas	45510	186264	41300	81837	81837	0.00	0.00	dez 8 00:02
code	tomascandelas	45511	186264	42160	81837	81837	0.00	0.00	dez 8 00:02
code	tomascandelas	45555	256956	61168	230264	230264	0.00	0.00	dez 8 00:02

```
tomascandelas@Ubuntu:~/Desktop/S0/Praticas$ ./procstat.sh -r 2
```

COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
Xorg	tomascandelas	1774	1680340	195760	11221306	11221306	8419.50	8419.50	dez 7 23:35
xdg-permission-	tomascandelas	1975	235592	6108	20652	20652	0.00	0.00	dez 7 23:35
xdg-document-po	tomascandelas	2223	605116	6556	70140	70140	0.00	0.00	dez 7 23:35
update-notifier	tomascandelas	3567	419196	29644	124495321	124495321	0.00	0.00	dez 7 23:36
tracker-miner-f	tomascandelas	1680	724792	25048	49693430	49693430	0.00	0.00	dez 7 23:35
systemd	tomascandelas	1664	19368	10568	957148600	957148600	0.00	0.00	dez 7 23:35
soffice.bin	tomascandelas	6497	1756724	311544	32870735	32870735	32.00	32.00	dez 7 23:56
snap-store	tomascandelas	2187	1575392	389556	34665570	34665570	0.00	0.00	dez 7 23:35
pulseaudio	tomascandelas	1678	3129836	21168	22246803	22246803	17497.50	17497.50	dez 7 23:35
procstat.sh	tomascandelas	211715	9628	3728	16133227	16133227	11201611.00	11201611.00	dez 8 00:31
oosplash	tomascandelas	6462	161620	4812	222928	222928	0.00	0.00	dez 7 23:56
nautilus	tomascandelas	4025	1205376	78616	15807984	15807984	256.00	256.00	dez 7 23:37
nacl_helper	tomascandelas	6031	10748	4236	10204	10204	0.00	0.00	dez 7 23:55
indicator-multi	tomascandelas	2103	421384	31080	78021256	78021256	58263.00	58263.00	dez 7 23:35
ibus-x11	tomascandelas	1957	196024	25984	292740	292740	0.00	0.00	dez 7 23:35
ibus-portal	tomascandelas	1960	240408	9196	70913	70913	0.00	0.00	dez 7 23:35
ibus-memconf	tomascandelas	1954	166608	8972	37353	37353	0.00	0.00	dez 7 23:35
ibus-extension-	tomascandelas	1955	275172	30512	878052	878052	0.00	0.00	dez 7 23:35
ibus-engine-sim	tomascandelas	2140	166604	9068	42509	42509	0.00	0.00	dez 7 23:35
ibus-daemon	tomascandelas	1950	314656	10196	74892	74892	0.00	0.00	dez 7 23:35
gvfs-udisks2-vo	tomascandelas	1714	317604	11540	2531772	2531772	0.00	0.00	dez 7 23:35
gvfs-mtp-volume	tomascandelas	1751	235700	5616	46024	46024	0.00	0.00	dez 7 23:35
gvfs-gphoto2-vo	tomascandelas	1726	238108	6332	50400	50400	0.00	0.00	dez 7 23:35
gvfs-goa-volume	tomascandelas	1730	235872	6240	27196	27196	0.00	0.00	dez 7 23:35
gvfsd-trash	tomascandelas	2042	317668	10636	119485	119485	0.00	0.00	dez 7 23:35

```
tomascandelas@Ubuntu:~/Desktop/S0/Praticas$ sudo ./procstat.sh -u root 2
```

```
[sudo] password for tomascandelas:
```

COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
accounts-daemon	root	862	241980	9396	642773	642773	44.00	44.00	dez 7 23:34
acpid	root	863	2540	784	213839	213839	0.00	0.00	dez 7 23:34
bluetoothd	root	867	9056	5648	1820218	1820218	1566.00	1566.00	dez 7 23:34
cron	root	868	9416	3056	1112324	1112324	0.00	0.00	dez 7 23:34
cups-browsed	root	14664	180440	12616	102759	102759	0.00	0.00	dez 8 00:00
cupsd	root	14663	28444	8964	500937	500937	0.00	0.00	dez 8 00:00
fwupd	root	2336	377784	25788	838455	838455	0.00	0.00	dez 7 23:35
gdm3	root	1001	239612	8568	292137	292137	0.00	0.00	dez 7 23:34
gdm-session-wor	root	1639	167196	9736	258418	258418	0.00	0.00	dez 7 23:35
irqbalance	root	883	81900	3688	2783118	2783118	0.00	0.00	dez 7 23:34
ModemManager	root	958	313752	10732	260491	260491	0.00	0.00	dez 7 23:34
networkd-dispat	root	887	39308	20108	1298875	1298875	0.00	0.00	dez 7 23:34
NetworkManager	root	875	339700	23624	427944	427944	0.00	0.00	dez 7 23:34
polkitd	root	891	245176	14956	44239015	44239015	0.00	0.00	dez 7 23:34
procstat.sh	root	220007	9624	3600	18894116	18894116	12068994.00	12068994.00	dez 8 00:32
snappd	root	901	1669188	32412	5885870	5885870	0.00	0.00	dez 7 23:34
sudo	root	220001	12100	4700	187764	187764	0.00	0.00	dez 8 00:32
switcheroo-cont	root	905	235604	5908	209487	209487	0.00	0.00	dez 7 23:34
systemd-journal	root	310	83532	46328	3946527	3946527	1847.00	1847.00	dez 7 23:34
systemd-logind	root	908	16996	8276	10756212	10756212	0.00	0.00	dez 7 23:34
systemd	root	1	167944	11868	1737836228	1737836228	0.00	0.00	dez 7 23:34
systemd-udev	root	337	23860	7568	68322042	68322042	0.00	0.00	dez 7 23:34
thermald	root	910	125880	10052	290730	290730	14.50	14.50	dez 7 23:34
udisksd	root	911	395860	15480	1216120	1216120	0.00	0.00	dez 7 23:34
unattended-upgr	root	1022	117836	22500	1228173	1228173	0.00	0.00	dez 7 23:34
upowerd	root	1164	252364	10028	625254	625254	0.00	0.00	dez 7 23:34
wpa_supplicant	root	913	14196	9276	50778	50778	0.00	0.00	dez 7 23:34