

## Sistemas Operativos

### 2020/2021

# Trabalho 1 - Estatísticas de processos em bash

fortnyce@fortnyce:~/Desktop/SO/avaliacoes\$ ./procstat.sh 10									
COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
agent	fortnyce	2315	236892	4760	22078	192	0	0	Dec 7 20:39
applet.py	fortnyce	3397	59480	34520	3308125	981120	.80	2.40	Dec 7 20:40
at-spi2-registr	fortnyce	2997	162908	6500	39064	253470	0	0	Dec 7 20:39
at-spi-bus-laun	fortnyce	2089	305564	6844	22299	1027	0	0	Dec 7 20:39
awk	fortnyce	293216	13772	1428	10116	0	0	0	Dec 7 23:32
awk	fortnyce	300689	1064	4	1692	0	0	0	Dec 7 23:42
bamfdaemon	fortnyce	2441	270932	28204	1607006	540650	0	0	Dec 7 20:39
basename	fortnyce	294364	2376	588	4292	0	0	0	Dec 7 23:35
basename	fortnyce	295800	376	4	1692	0	0	0	Dec 7 23:36
basename	fortnyce	299995	376	4	1692	0	0	0	Dec 7 23:41
basename	fortnyce	300882	2520	356	4292	0	0	0	Dec 7 23:42
basename	fortnyce	303159	2520	352	4292	0	0	0	Dec 7 23:49
bash	fortnyce	2418	11604	5328	367407	11860	0	0	Dec 7 20:39
bash	fortnyce	4475	11608	5340	1058519958	2716726	0	0	Dec 7 20:42
bash	fortnyce	4582	10876	5096	247731	2022	0	0	Dec 7 20:42
blueberry-obex-	fortnyce	2324	278892	33456	1513010	5177	0	0	Dec 7 20:39
cat	fortnyce	4812	8224	520	4292	0	0	0	Dec 7 20:43
cat	fortnyce	4813	8224	520	4436	144	0	0	Dec 7 20:43
chrome	fortnyce	278517	4718052	146728	1194759	1719796	145.20	0	Dec 7 22:42
chrome	fortnyce	304957	4720512	170464	2039732	2499858	290.40	0	Dec 7 23:52
chrome	fortnyce	308307	4579812	53136	34720	12	0	0	Dec 8 00:09
chrome	fortnyce	308397	4755268	280156	2417452	2493346	435.60	0	Dec 8 00:09
chrome	fortnyce	308616	4674868	141712	34265	912	0	0	Dec 8 00:11
chrome	fortnyce	308643	4528748	57912	213	209	0	0	Dec 8 00:11
chrome	fortnyce	4805	1416216	301480	1308048730	537007641	302.40	879.40	Dec 7 20:43
chrome	fortnyce	4816	273732	64116	184324	21	0	0	Dec 7 20:43
chrome	fortnyce	4817	273732	64692	184325	56	0	0	Dec 7 20:43
chrome	fortnyce	4821	273732	16848	86967890	46075021	0	0	Dec 7 20:43
chrome	fortnyce	4843	742408	198760	1483044	1266693	.50	0	Dec 7 20:43
chrome	fortnyce	4845	382776	188988	209895526	229672097	13375.50	5793.60	Dec 7 20:43
chrome	fortnyce	4858	364266	39560	112272525	116716312	0	0	Dec 7 20:43
chrome	fortnyce	4871	345220	32740	0	0	0	0	Dec 7 20:43
chrome	fortnyce	4952	4597484	81920	797198	98	0	0	Dec 7 20:43
chrome	fortnyce	4966	4716984	176168	17318852	401480	0	0	Dec 7 20:43
chrome	fortnyce	5293	5450952	817320	439415784	3783414	67324.60	.40	Dec 7 20:43
chrome	fortnyce	5219	4645588	167484	2781847	23996	0	0	Dec 7 20:43
chrome	fortnyce	5271	556520	58160	112039	406	0	0	Dec 7 20:43
cinnamon	fortnyce	2285	3894252	352244	82112982	71631080	3551.40	380.00	Dec 7 20:39
cinnamon	fortnyce	2352	276544	31336	966160	5265	0	0	Dec 7 20:39
cinnamon-launch	fortnyce	2283	54772	30508	961126	3968	0	0	Dec 7 20:39
cinnamon-screen	fortnyce	3278	468508	75904	2330930	231901	0	0	Dec 7 20:39
cinnamon-sess10	fortnyce	2094	375604	25948	9436688	296030	0	0	Dec 7 20:39
code	fortnyce	4379	4769048	173028	8942899	53074222	25.60	2537.10	Dec 7 20:41
code	fortnyce	4383	186260	41076	81845	20	0	0	Dec 7 20:41
code	fortnyce	4384	186260	41632	81845	4	0	0	Dec 7 20:41
code	fortnyce	4412	544132	180928	1364912	4769552	0	0	Dec 7 20:42
code	fortnyce	4423	261048	63012	7997439	4393760	14295.90	513.10	Dec 7 20:42
code	fortnyce	4444	15324196	29692	112786523	32020117	1547.80	13.90	Dec 7 20:42
code	fortnyce	4509	4576120	174412	126173731	29883841	2.60	2.60	Dec 7 20:42

**José Trigo** Nº 98597  
**Pedro Monteiro** Nº 97484

Iniciamos o script dizendo à shell qual o interpretador a utilizar com “#!/bin/bash” e é executado o comando “cd” para mudar a diretoria para /proc.

```
#!/bin/bash
cd /proc
```

De seguida, um ciclo for percorre cada sub diretoria existente na diretoria /proc, é guardado o seu caminho relativo na variável entry\_basename e utiliza-se uma expressão regex para verificar se o nome da pasta apenas inclui caracteres numéricos. Em caso afirmativo, trata-se de uma sub diretoria de /proc relativa a um processo. Por fim, o caminho relativo (nome da pasta) é guardado num array que contém todas as sub diretórias válidas (que tenham um nome numérico e por isso correspondam a uma pasta de um pid).

```
# ciclo for para cada ficheiro ou diretoria contido em /proc/
for entry in /proc/*; do
# obter apenas o basename (caminho relativo da pasta) (o PID)
entry_basename=$(basename $entry)
# Obter apenas folders ou files com nomes apenas numéricos
if [[ $entry_basename =~ ^[0-9]+$ ]]; then
# Obter apenas folders com permissões de Leitura
if [ -r "$entry/io" ] && [ -r "$entry/comm" ] &&
[ -r "$entry/status" ] && [ -r "/proc/$entry_basename" ]; then
# Lista com os PIDs todos (que permitem Leitura) que mais
tarde vão ser utilizados nas funções
pid_list+=($entry_basename)
fi
fi
done
```

Na função “read\_io()”, são inicializados dois arrays que irão conter os valores numéricos de “rchar” e “wchar”.

É percorrido cada pid guardado no array analisado anteriormente e verifica-se se a pasta do processo ainda existe, dada a natureza dinâmica da criação e terminação de processos, e só depois é que se procede ao armazenamento do valor numérico de “rchar” e “wchar” nos respetivos arrays.

Caso o processo tenha sido terminado, é guardado o valor 0 em cada array para preservar a estrutura destes e evitar incompatibilidades com a função “process\_list()”, que recorrerá a estes mais tarde.

```
read_io () {
    rchar_array=() # Inicializar arrays
    wchar_array=()
    for pid in ${pid_list[@]}; do
        # verificar se a diretoria ainda existe
        if [ -e /proc/$pid ]; then
            # Obter todas as Linhas com rchar
            rchar=$(grep 'rchar' $pid/io)
            # Obter apenas o valor numérico
            rchar_value=$(echo $rchar | grep -o -E '[0-9]+')
            # Obter todas as Linhas com wchar
            wchar=$(grep 'wchar' $pid/io)
            # Obter apenas o valor numérico
            wchar_value=$(echo $wchar | grep -o -E '[0-9]+')
            # Guardar valores de rchar e wchar em arrays
            rchar_array+=($rchar_value)
            wchar_array+=($wchar_value)
        else
            # Se o processo entretanto terminar, preencher o array com 0 à mesma
            # para não haver incompatibilidade de indexes na função process_list
            # que utiliza os valores destes arrays
            rchar_array+=(0)
            wchar_array+=(0)
        fi
    done
}
```

O cálculo de “ratew” e “rater” dá se da seguinte maneira:

```
read_rate_array=() #Inicializar arrays
write_rate_array=()
read_io # Ler rchar e wchar pela 1ª vez
first_rchar_array="${rchar_array[@]}") # Copiar array da 1ª Leitura uma
vez que
first_wchar_array="${wchar_array[@]}") # vai ser overwritten
sleep $1 # Esperar s segundos
read_io # Ler rchar e wchar pela 2ª vez

for i in ${!rchar_array[@]}; do # Calcular read rate e write rate em
Bytes/s
    op1=${rchar_array[i]}
    op2=${first_rchar_array[i]}
    read_rate=$(echo "scale=2; ($op1 - $op2)/$1" | bc)
    op1=${wchar_array[i]}
    op2=${first_wchar_array[i]}
    write_rate=$(echo "scale=2; ($op1 - $op2)/$1" | bc)

    read_rate_array+=($read_rate)
    write_rate_array+=($write_rate)
done
```

São feitas duas leituras dos valores “rchar” e “wchar” recorrendo à função “read\_io()”, com um intervalo de tempo definido pelo utilizador. Após a primeira leitura, o array inicializado pela função é copiado para outra variável de modo a não ser reescrito aquando da segunda execução da função “read\_io()”.

Para efetuar os cálculos, é percorrido cada elemento dos arrays e utiliza-se a seguinte fórmula para obter as rates de read e write:

$$\text{rate} = (\text{leitura2} - \text{leitura1}) / \text{intervalo de tempo}$$

Por fim, os valores calculados são guardados em arrays que serão acedidos pela função “process\_list()”.

```
process_list () {
    counter=0
    for pid in ${pid_list[@]}; do
# verificar se a diretoria ainda existe
        if [ -e /proc/$pid ]; then
# Substituir espaços por underscores, de modo a preservar a formatação
no print
            comm=$(cat $pid/comm | tr " " "_")
            uid=$( stat -c "%u" /proc/${pid} )
            user=$( id -nu ${uid} )
# Obter todas as Linhas com VmSize
            VmSize=$(grep 'VmSize' $pid/status)
# Obter apenas o valor numérico
            VmSize_value=$(echo $VmSize | grep -o -E '[0-9]+')
            VmRSS=$(grep 'VmRSS' $pid/status)
            VmRSS_value=$(echo $VmRSS | grep -o -E '[0-9]+')
            rchar=$(grep 'rchar' $pid/io)
            rchar_value=$(echo $rchar | grep -o -E '[0-9]+')
            wchar=$(grep 'wchar' $pid/io)
            wchar_value=$(echo $wchar | grep -o -E '[0-9]+')
            process_date=$(ls -ld /proc/$pid)
            process_date=$(echo $process_date | awk '{ print $6" "$7"
"$8}')
            rater=${read_rate_array[counter]}
            ratew=${write_rate_array[counter]}

# Se o valor for "" alterar para 0
            if [[ $VmSize_value == "" ]]; then
                VmSize_value="0"
            fi

# Se o valor for "" alterar para 0
            if [[ $VmRSS_value == "" ]]; then
                VmRSS_value="0"
            fi

            printf '%-30s\t %-20s\t %10s\t %10s\t %10s\t %10s\t %9s\t
%10s\t %10s\t %5s\n' "$comm" "$user" "$pid" "$VmSize_value"
"$VmRSS_value" "$rchar_value" "$wchar_value" "$rater" "$ratew"
```

```
"$process_date"
    (( counter++ ))
fi
done
}
```

É na função “process\_list()” que são procurados todos os valores que aparecem no ecrã. Em alguns casos utilizam-se os arrays declarados em porções de código anteriores para obter os valores pretendidos (utilizando um contador “counter” que vai incrementando a cada iteração do ciclo for), e noutras casos, é feita a leitura direta do valor com o auxílio dos comandos “cat”, “grep”, “awk” e “echo”. Após obter todos valores necessários, é feito o print de linhas que contêm esses valores.

Podemos selecionar os processos a visualizar recorrendo a opções de filtragem aquando da inicialização do script.

É de salientar que existem combinações incompatíveis, como por exemplo utilizar várias opções de sort ao mesmo tempo. Para impedir a inclusão de várias opções de sort, declaramos uma variável “sort\_parameter=1” no início do script e verificamos se o valor de “sort\_parameter” já foi alterado: if [ \$sort\_parameter -ne 1 ].

```
while getopts "c:s:e:u:p:mtdwr" OPTION; do
if [[ ${OPTARG} == -* ]]; then
    echo "Missing argument for -$OPTARG"
    exit 1
fi
case $OPTION in
c)
    flag_c=${OPTARG}
    ;;
s)
    flag_s=${OPTARG}
    ;;
e)
    flag_e=${OPTARG}
    ;;
u)
    flag_u=${OPTARG}
    ;;
p)
    flag_p=${OPTARG}
    ;;
m)
    sort_parameter=4
    ;;
esac
done
```

```
;;
t)
if [ $sort_parameter -ne 1 ]; then
    echo "Invalid argument: multiple sort parameters detected"
    exit 1
fi
sort_parameter=5
;;
d)
if [ $sort_parameter -ne 1 ]; then
    echo "Invalid argument: multiple sort parameters detected"
    exit 1
fi
sort_parameter=8
;;
w)
if [ $sort_parameter -ne 1 ]; then
    echo "Invalid argument: multiple sort parameters detected"
    exit 1
fi
sort_parameter=9
;;
r)
sort_reverse="-r"
;;
*)
echo "Invalid options provided"
exit 1
;;
esac
done

shift $((OPTIND-1))
```

As opções de sort e a opção “-p” são aplicadas dentro da função “process\_list()”, que faz o print dos processos.

```
process_list () {
    counter=0
    for pid in ${pid_list[@]}; do
        ...
        if [ $counter -ge $flag_p ]; then
            break
        fi
        ...
        (( counter++ ))
    done | sort -n -k $sort_parameter $sort_reverse
}
```

As opções de filtragem são aplicadas no ciclo for analisado anteriormente, que percorre as subdiretorias da diretoria /proc.

No caso da opção “-c”, verificamos se o valor de comm não corresponde ao filtro passado na opção e se esse for o caso, guardamos o pid num array para posteriormente procedermos à sua eliminação do array de pids a listar na tabela. O mesmo raciocínio é aplicado às opções “-u”, “-s” e “-e”.

```
pid_list=()
pid_list2=()
pid_list3=()
pid_list4=()
min_date=$(date -d "${flag_s}" +"%s")
max_date=$(date -d "${flag_e}" +"%s")
for entry in /proc/*; do
    ...
    # Lista com PIDs que não contêm a FLAG -c
    if [[ $flag_c != "" ]]; then
        comm=$(cat $entry/comm)
        if ! [[ $comm =~ $flag_c ]]; then
            pid_list2+=($entry_basename)
        fi
    fi

    # Lista com PIDs que não contêm a flag -u
    if [[ $flag_u != "" ]]; then
        uid=$( stat -c "%u" /proc/${entry_basename} )"
```

```

        user=$( id -nu ${uid} )
        if ! [[ $user =~ $flag_u ]]; then
            pid_list3+=($entry_basename)
        fi
    fi

    # Lista com PIDs que estão no intervalo de tempo definido pelas
    # flags -s e -e
    if [[ $flag_s != "" || $flag_e != "" ]]; then
        pid_date=$(ls -ld /proc/$entry_basename)
        pid_date=$(echo $pid_date | awk '{ print $6" "$7" "$8}')
        pid_date=$(date -d "${pid_date}" +"%s")

        if [[ $flag_s != "" ]]; then
            if [ $pid_date -lt $min_date ]; then
                pid_list4+=($entry_basename)
            fi
        fi
        if [[ $flag_e != "" ]]; then
            if [ $pid_date -gt $max_date ]; then
                pid_list4+=($entry_basename)
            fi
        fi
    fi
    fi
    ...
done

```

Para obtermos o nosso array final de pids a listar na tabela subtraímos os pids\_to\_remove à pid\_list antes de chamarmos a função process\_list() que lê a pid\_list final e faz print dos processos para o ecrã.

```

# Subtrair arrays com PIDS que não contêm as flags ao array com os PIDs
# todos
pids_to_remove_list=()
pids_to_remove_list=( "${pid_list2[@]}" "${pid_list3[@]}"
"${pid_list4[@]}" )
for i in "${pids_to_remove_list[@]}"; do
    for k in "${!pid_list[@]}"; do
        if [[ ${pid_list[k]} = $i ]]; then
            unset 'pid_list[k]'
        fi
    done
done

```

## Demonstrações e testes de robustez

Realizamos alguns testes para verificar a eficiência e robustez do script. Começamos por correr o script normalmente, sem qualquer parâmetro, e podemos ver que apenas passando um argumento válido para o sleep (em segundos) é que o script é executado.

procstat.sh -c "d.*"									
COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
applet.py	pedromonteiro	2545	249940	32604	3319257	126705	0	0	Dec 7 08:51
at-spi2-registr	pedromonteiro	1985	220780	6908	29711	21650	0	0	Dec 7 08:50
at-spi-bus-laun	pedromonteiro	1978	349416	6432	20943	1349	0	0	Dec 7 08:50
bamfdaemon	pedromonteiro	2273	415348	24608	691014	53453	0	0	Dec 7 08:50
bash	pedromonteiro	2291	22792	5036	253366	2285	0	0	Dec 7 08:50
bash	pedromonteiro	6929	22852	5504	240550581	656435	0	0	Dec 7 10:04
blueberry-obex-	pedromonteiro	2206	433740	31200	1305609	4537	0	0	Dec 7 08:50
blueberry-tray	pedromonteiro	2279	28988	8496	210094	0	0	0	Dec 7 08:50
cat	pedromonteiro	2662	7612	844	1948	0	0	0	Dec 7 08:52
cat	pedromonteiro	2663	7612	768	2285	337	0	0	Dec 7 08:52

Através da opção -c, verificamos que, ao passarmos um argumento, são selecionados os processos que apenas contêm no seu nome a string passada no argumento da opção -c.  
De notar que, caso não seja passado um argumento para o sleep o programa não é executado.

procstat.sh -c "d.*"									
COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
bamfdaemon	pedromonteiro	2273	415348	24872	709861	69713	0	0	Dec 7 08:50
cinnamon2d	pedromonteiro	2157	12888	3124	4041	0	0	0	Dec 7 08:50
code	pedromonteiro	8438	4954076	145056	38400608	13723809	0	0	Dec 7 10:07
code	pedromonteiro	8442	338180	39668	77946	20	0	0	Dec 7 10:07
code	pedromonteiro	8443	338180	39480	77946	4	0	0	Dec 7 10:07
code	pedromonteiro	8468	564056	141152	1658834	269759	0	0	Dec 7 10:07
code	pedromonteiro	8481	422944	58012	802234	106628	0	0	Dec 7 10:07
code	pedromonteiro	8494	15382248	189204	33818041	2038023	1397.00	0	Dec 7 10:07
code	pedromonteiro	8525	4780972	119812	11792957	7069450	0	0	Dec 7 10:07
code	pedromonteiro	8573	4711728	150720	20448156	205076	8.00	83.00	Dec 7 10:07
code	pedromonteiro	8589	4611696	65376	940928	576	0	0	Dec 7 10:07
code	pedromonteiro	8758	15397588	206868	38079529	2512267	1414.00	37.00	Dec 7 10:07
code	pedromonteiro	8784	4701392	136392	18869581	955044	13.00	0	Dec 7 10:07
code	pedromonteiro	8801	4611696	65632	943872	4385	0	0	Dec 7 10:07
csd-ally-keyboa	pedromonteiro	2029	396632	18964	537384	8365	0	0	Dec 7 08:50
csd-ally-settin	pedromonteiro	2039	395980	19140	537872	8905	0	0	Dec 7 08:50
csd-automount	pedromonteiro	2036	474128	19936	541301	10750	0	0	Dec 7 08:50
csd-background	pedromonteiro	2026	479472	29276	1696107	10326	0	0	Dec 7 08:50
csd-clipboard	pedromonteiro	2032	319772	19144	536504	9796	0	0	Dec 7 08:50
csd-color	pedromonteiro	2043	573464	21108	470944	13369	0	0	Dec 7 08:50
csd-cursor	pedromonteiro	2017	319788	18856	536656	8444	0	0	Dec 7 08:50
csd-housekeepin	pedromonteiro	2027	396612	19172	1513434	8322	0	0	Dec 7 08:50
csd-keyboard	pedromonteiro	2034	446240	21796	527630	68161	0	0	Dec 7 08:50
csd-media-keys	pedromonteiro	2014	778628	22332	625228	21836	0	0	Dec 7 08:50
csd-mouse	pedromonteiro	2038	396636	19192	537440	12057	0	0	Dec 7 08:50
csd-orientation	pedromonteiro	2025	543412	19128	537672	8705	0	0	Dec 7 08:50
csd-power	pedromonteiro	2020	700584	29552	639916	25858	0	0	Dec 7 08:50
csd-print-notif	pedromonteiro	2096	501680	12616	73116	584	0	0	Dec 7 08:50
csd-screensaver	pedromonteiro	2041	444484	22228	566299	8374	0	0	Dec 7 08:50
csd-sound	pedromonteiro	2028	393556	18676	536520	8748	0	0	Dec 7 08:50
csd-wacom	pedromonteiro	2040	765784	21124	731848	8850	0	0	Dec 7 08:50
csd-xrandr	pedromonteiro	2021	378704	20652	402884	9236	0	0	Dec 7 08:50
csd-xsettings	pedromonteiro	2022	482276	18896	400019	10367	0	0	Dec 7 08:50
dbus-daemon	pedromonteiro	1889	51160	5484	185574	328	0	0	Dec 7 08:50
dbus-daemon	pedromonteiro	1983	50312	4636	49144	331	0	0	Dec 7 08:50
dconf-service	pedromonteiro	2084	188132	5388	34849	793488	0	0	Dec 7 08:50
Discord	pedromonteiro	2652	5027776	138676	301875779	858314	91172.00	7.00	Dec 7 08:51

Semelhante à opção -c temos também a opção -u, que faz uma pesquisa por todos os USERS associados aos processos que estão a executar e apenas retorna aqueles que são passados como argumento.

COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
agent	fortnyce	2315	236892	4760	22078	192	74892.00	152.00	Dec 7 20:39
apple.py	fortnyce	3397	59480	34520	3046661	73272	0	0	Dec 7 20:40
at-spi2-registr	fortnyce	2097	162908	6500	33128	29318	0	0	Dec 7 20:39
at-spi-bus-laun	fortnyce	2089	305564	6844	22299	1027	0	0	Dec 7 20:39
bamdaemon	fortnyce	2441	269088	26360	703579	44337	0	0	Dec 7 20:39
bash	fortnyce	2418	10872	5056	247731	1737	0	0	Dec 7 20:39
bash	fortnyce	4475	11088	5340	36630602	105292	0	2724.00	Dec 7 20:42
blueberry-obex-	fortnyce	4502	10876	5096	247731	1902	4457.00	8.00	Dec 7 20:42
cat	fortnyce	2324	278892	33456	1512858	4721	0	0	Dec 7 20:39
cat	fortnyce	4812	8224	520	4292	0	0	0	Dec 7 20:43
chrome	fortnyce	4813	8224	520	4364	72	1490.00	15.00	Dec 7 20:43
chrome	fortnyce	12133	4579812	53228	33135	11	1446.00	0	Dec 7 20:47
chrome	fortnyce	4805	909660	252080	14996082	55456293	0	0	Dec 7 20:43
chrome	fortnyce	4816	273732	64116	184324	21	0	0	Dec 7 20:43
chrome	fortnyce	4817	273732	64692	184325	56	0	0	Dec 7 20:43
chrome	fortnyce	4821	273732	16048	2381320	3454849	0	0	Dec 7 20:43
chrome	fortnyce	4843	624288	158584	1029165	148496	0	0	Dec 7 20:43
chrome	fortnyce	4845	356136	97356	17844433	14546425	0	0	Dec 7 20:43
chrome	fortnyce	4858	356064	37956	9533353	6819773	0	0	Dec 7 20:43
chrome	fortnyce	4871	345220	32740	0	0	0	0	Dec 7 20:43
chrome	fortnyce	4952	4589288	81372	76169	84	0	0	Dec 7 20:43
chrome	fortnyce	4966	4706740	167572	3124933	390130	0	0	Dec 7 20:43
chrome	fortnyce	5203	13450332	468324	42671653	3613021	0	0	Dec 7 20:43
chrome	fortnyce	5219	4625656	95236	303548	23011	0	0	Dec 7 20:43
chrome	fortnyce	5271	548324	58204	111791	158	1446.00	0	Dec 7 20:43
chrome	fortnyce	5298	4646444	119356	283789	59966	0	0	Dec 7 20:43
cinnamon	fortnyce	2285	3789956	209088	11965142	4096143	0	0	Dec 7 20:39
cinnamon-killer	fortnyce	2352	276544	31236	966008	4989	0	0	Dec 7 20:39
cinnamon-launch	fortnyce	2283	54772	30508	961126	3968	0	0	Dec 7 20:39
cinnamon-screen	fortnyce	3278	468508	51880	2209282	23973	0	0	Dec 7 20:39
cinnamon-sessio	fortnyce	2084	375604	25948	9301768	75294	0	0	Dec 7 20:39
code	fortnyce	4379	4721140	139644	6637849	5934705	0	0	Dec 7 20:41

Executando o script com ambas as opções, -c e -u, são impressos no terminal os processos correspondentes à string associada à opção -c, que tenha sido executados pelo utilizador associado à opção -u.

COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
code	fortnyce	4379	4721140	139644	6637849	5934705	0	0	Dec 7 20:41
code	fortnyce	4383	186260	41076	81845	20	0	0	Dec 7 20:41
code	fortnyce	4384	186260	41032	81845	4	0	0	Dec 7 20:41
code	fortnyce	4412	460268	125132	1169164	219623	0	0	Dec 7 20:42
code	fortnyce	4423	260024	62672	2302577	251678	0	0	Dec 7 20:42
code	fortnyce	4444	19489660	239468	44320275	3566092	1503.00	8.00	Dec 7 20:42
code	fortnyce	4509	4559728	157776	29620908	834081	0	13.00	Dec 7 20:42
code	fortnyce	4525	4457144	66380	954025	1676	0	0	Dec 7 20:42
code	fortnyce	4538	4625252	124112	30622421	7079398	0	0	Dec 7 20:42
code	fortnyce	4559	4456120	65152	736531	2320	0	0	Dec 7 20:42
code	fortnyce	4658	4422312	59796	332662	601	0	0	Dec 7 20:42

As opções “-s”, que corresponde à data mínima de início de um processo, e “-e”, que corresponde à data máxima de início de um processo, permitem-nos filtrar os processos pelo momento em que começaram.

Exemplificando,

COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
applet.py	pedromonteiro	2545	249940	32604	3325649	149993	0	0	Dec 7 08:51
at-sp1-registr	pedromonteiro	1985	220780	6908	31039	45274	0	0	Dec 7 08:50
at-sp1-bus-laun	pedromonteiro	1978	349416	6432	20943	1349	0	0	Dec 7 08:50
bamfdaemon	pedromonteiro	2273	415348	24872	752616	88589	0	0	Dec 7 08:50
bash	pedromonteiro	2291	22792	5036	253266	2285	0	0	Dec 7 08:50
blueberry-obex-	pedromonteiro	2206	433740	31200	1305673	4729	0	0	Dec 7 08:50
blueberry-tray	pedromonteiro	2279	28988	8496	210694	0	0	0	Dec 7 08:56
cat	pedromonteiro	2662	7612	844	1948	0	0	0	Dec 7 08:52
cat	pedromonteiro	2663	7612	768	244778	242830	0	0	Dec 7 08:52
chrome	pedromonteiro	2656	1099624	232748	428553273	151479189	66.00	34.00	Dec 7 08:51
chrome	pedromonteiro	2711	461708	61248	164660	21	0	0	Dec 7 08:52
chrome	pedromonteiro	2712	461708	61588	164670	56	0	0	Dec 7 08:52
chrome	pedromonteiro	2747	461708	16584	6011754	11799279	0	0	Dec 7 08:52
chrome	pedromonteiro	2771	930100	233348	2361160	2667729	5.00	0	Dec 7 08:52
chrome	pedromonteiro	2773	552660	87088	311479679	326263113	1111.00	2.00	Dec 7 08:52
chrome	pedromonteiro	2801	538732	33388	0	0	0	0	Dec 7 08:52
chrome	pedromonteiro	2817	521100	37848	5207216	4827943	0	0	Dec 7 08:52
chrome	pedromonteiro	2838	4808868	92856	333386	14564	0	0	Dec 7 08:52
chrome	pedromonteiro	2849	4808628	88112	416004	69381	0	0	Dec 7 08:52

Observamos que apenas foram listados os processos que começaram entre as datas inseridas, ou seja, entre as 8h00 e as 9h00.

Caso não haja processos iniciados dentro desse tempo, deverá apenas aparecer no terminal o cabeçalho, onde a lista de processos estará vazia.

```
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -s "Dec 7 08:00" -e "Dec 7 08:30" 1
```

Em termos de escolha e pesquisa de processos temos ainda a opção **-p**, que imprime no terminal o número de processos representado pelo seu argumento (número inteiro maior que 0).

```
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -p 5 1
COMM           USER     PID      MEM      RSS      READB      WRITEB      RATER      RATEW      DATE
bash          pedromonteiro   11427    22852     5328    67815430     302723        0         0 Dec 7 10:47
cinnamon-sessio  pedromonteiro   1874     476712    23804    17681794    62682889        0         0 Dec 7 08:50
dbus-daemon    pedromonteiro   1880     51160     5484    288417        328        0         0 Dec 7 08:50
gnome-terminal- pedromonteiro   11421    557132    37416    822825    8374060        0      1056.00 Dec 7 10:47
systemd        pedromonteiro   1853     77056     8044    306102614   16958399        0         0 Dec 7 08:50
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -p 3 1
COMM           USER     PID      MEM      RSS      READB      WRITEB      RATER      RATEW      DATE
bash          pedromonteiro   11427    22852     5328    74800426     333486        0         0 Dec 7 10:47
gnome-terminal-  pedromonteiro   11421    557132    37416    824057    8619891        0      856.00 Dec 7 10:47
procstat.sh    pedromonteiro   1671     13048     3472    6369795    26991  2297509.00     8464.00 Dec 7 11:04
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ 
```

Desenvolvemos também opções que permitem ordenar os processos, pela memória (opção -m), pelo RSS (opção -t), pela taxa de leitura (opção -d) e pela taxa de escrita (opção -w).

Ao executarmos o script com a opção -m verificamos que os processos são listados consoante a sua memória, de forma crescente, ou seja, o processo com menor valor de memória aparecerá listado em primeiro lugar.

COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
sh	pedromonteiro	2170	4628	920	1948	0	0	0	Dec 7 12:05
sh	pedromonteiro	2285	4628	876	1948	0	0	0	Dec 7 12:05
cat	pedromonteiro	15380	7612	772	1948	0	0	0	Dec 7 12:08
cat	pedromonteiro	15381	7612	884	50491	48543	0	0	Dec 7 12:08
rfkill	pedromonteiro	2188	9748	936	8457	130	0	0	Dec 7 12:05
cinnamon2d	pedromonteiro	2052	12888	3192	4041	0	0	0	Dec 7 12:04
procstat.sh	pedromonteiro	7039	13048	3560	6239708	26474	2243434.00	8332.00	Dec 7 15:50
bash	pedromonteiro	2198	22792	5180	254734	2285	0	0	Dec 7 12:05
bash	pedromonteiro	7011	22852	5352	262811	2763	0	0	Dec 7 15:49
python2	pedromonteiro	2187	26268	7692	467408	0	0	0	Dec 7 12:05
nac1_helper	pedromonteiro	15391	27272	4012	7957	56	0	0	Dec 7 12:08
blueberry-tray	pedromonteiro	2168	28988	8548	210694	0	0	0	Dec 7 12:05
cinnamon-subpro	pedromonteiro	29268	50272	12036	727959	0	0	0	Dec 7 13:15
dbus-daemon	pedromonteiro	1876	50312	4608	57776	320	0	0	Dec 7 12:04
dbus-daemon	pedromonteiro	1784	51344	5660	506998	328	0	0	Dec 7 12:04
python3	pedromonteiro	29269	67700	20768	2184748	517	0	0	Dec 7 13:15
systemd	pedromonteiro	1749	77056	8184	4430124340	329303873	0	0	Dec 7 12:04
obexd	pedromonteiro	2152	82740	6756	23630	268	0	0	Dec 7 12:04
dconf-service	pedromonteiro	1975	188136	5412	36521	1692859	0	0	Dec 7 12:04
gvfsd-metadata	pedromonteiro	2194	197580	6104	49953	551065	0	0	Dec 7 12:05
ibus-engine-sim	pedromonteiro	1896	197912	6664	80718608	122837024	0	0	Dec 7 12:04
at-spi2-registr	pedromonteiro	1878	220780	6820	60831	229506	0	0	Dec 7 12:04
applet.py	pedromonteiro	2397	249940	32408	3416705	435433	0	0	Dec 7 12:05
gvfs-goa-volume	pedromonteiro	2014	257000	5912	20769	3406	0	0	Dec 7 12:04
gvfs-mtp-volume	pedromonteiro	2005	268772	6040	31657	3078	0	0	Dec 7 12:04
ibus-portal	pedromonteiro	1886	271588	6648	70664	83089	0	0	Dec 7 12:04
pxgsettings	pedromonteiro	2286	273348	6064	18559	936	0	0	Dec 7 12:05
ibus-dconf	pedromonteiro	1858	273652	6584	22108	697	0	0	Dec 7 12:04
gvfs-gphoto2-volume	pedromonteiro	2072	281704	6668	35817	3026	0	0	Dec 7 12:04
gvfsd	pedromonteiro	1851	284972	7332	194478	25753	0	0	Dec 7 12:04

Podemos organizar a forma como os processos são listados a partir do seu valor de RSS, isto é, a partir do valor residente em memória física, com recurso à opção -t.

COMM	USER	PID	MEM	RSS	READB	WRITEB	RATER	RATEW	DATE
cat	pedromonteiro	15380	7612	772	1948	0	0	0	Dec 7 12:08
sh	pedromonteiro	2285	4628	876	1948	0	0	0	Dec 7 12:05
cat	pedromonteiro	15381	7612	884	109434	107486	0	0	Dec 7 12:08
sh	pedromonteiro	2170	4628	920	1948	0	0	0	Dec 7 12:05
rfkill	pedromonteiro	2188	9748	936	8457	130	0	0	Dec 7 12:05
cinnamon2d	pedromonteiro	2052	12888	3192	4041	0	0	0	Dec 7 12:04
procstat.sh	pedromonteiro	13543	13020	3588	6249614	26584	2243478.00	8353.00	Dec 7 15:54
nac1_helper	pedromonteiro	15391	27272	4012	7957	56	0	0	Dec 7 12:08
dbus-daemon	pedromonteiro	1876	50312	4608	58227	320	0	0	Dec 7 12:04
bash	pedromonteiro	2198	22792	5180	254734	2285	0	0	Dec 7 12:05
gvfsd-fuse	pedromonteiro	1856	416116	5300	27308	1024	0	0	Dec 7 12:04
bash	pedromonteiro	7011	22852	5356	18274855	108331	0	0	Dec 7 15:49
dconf-service	pedromonteiro	1975	188136	5412	36665	1767375	0	0	Dec 7 12:04
ibus-daemon	pedromonteiro	1784	51344	5660	514118	328	0	0	Dec 7 12:04
gvfs-goa-volume	pedromonteiro	2014	267000	5912	29927	3670	0	0	Dec 7 12:04
gvfs-mtp-volume	pedromonteiro	2005	268772	6040	31793	3382	0	0	Dec 7 12:04
pxgsettings	pedromonteiro	2286	273348	6064	18559	936	0	0	Dec 7 12:05
gvfsd-metadata	pedromonteiro	2194	197580	6104	50641	593569	0	0	Dec 7 12:05
at-spi-bus-laun	pedromonteiro	1871	349416	6264	20951	1381	0	0	Dec 7 12:04
ibus-dconf	pedromonteiro	1858	273652	6584	22108	697	0	0	Dec 7 12:04
ibus-portal	pedromonteiro	1886	271588	6648	71432	84369	0	0	Dec 7 12:04
ibus-engine-sim	pedromonteiro	1896	197912	6664	80758792	122897792	0	0	Dec 7 12:04

Observamos, então, que o primeiro processo a aparecer é o que apresenta menor valor de RSS, sendo os seguintes processos listados por ordem crescente desse valor.

Quanto aos valores das taxas de leitura e escrita (representados nas colunas RATER e RATEW, respetivamente), podemos também fazer uma ordenação de processos, através da opção -d, para o caso de querermos ordenar consoante os valores da taxa de leitura, e através da opção -w, para o caso de considerarmos a taxa de escrita.

Utilizamos a opção -p, adicionalmente, para ser mais notória a ordenação, visto que existem muitos valores 0.

```
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -d -p 10 1
COMM           USER     PID     MEM     RSS    READB   WRITEB   RATER   RATEW   DATE
cat       pedromonteiro 15380   7612    772    1948      0      0      0      0 Dec 7 12:08
cat       pedromonteiro 15381   7612    884   229097   227149      0      0      0 Dec 7 12:08
chrome   pedromonteiro 15389  461708  61944   4797167   2703573      0      0      0 Dec 7 12:08
chrome   pedromonteiro 15390  461708  61796   164689      56      0      0      0 Dec 7 12:08
chrome   pedromonteiro 15394  461708  16452   32224243   24343980      0      0      0 Dec 7 12:08
chrome   pedromonteiro 15439  529516  39368   11506338  12410686      0      0      0 Dec 7 12:08
nacl_helper    pedromonteiro 15391   27272   4012    7957      56      0      0      0 Dec 7 12:08
chrome   pedromonteiro 15373  1053740 254968   636740977  222190055    228.00   100.00 Dec 7 12:08
chrome   pedromonteiro 15418  562388  96896   485081013  579267709   197435.00  212043.00 Dec 7 12:08
procstat.sh    pedromonteiro 1265   13020   3572   6234085   26571   2243527.00   8368.00 Dec 7 16:03
```

```
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -w -p 15 1
COMM           USER     PID     MEM     RSS    READB   WRITEB   RATER   RATEW   DATE
cat       pedromonteiro 15380   7612    772    1948      0      0      0      0 Dec 7 12:08
cat       pedromonteiro 15381   7612    884   243380   241432      0      0      0 Dec 7 12:08
chrome   pedromonteiro 15389  461708  61944   4797167   2703573      0      0      0 Dec 7 12:08
chrome   pedromonteiro 15390  461708  61796   164689      56      0      0      0 Dec 7 12:08
chrome   pedromonteiro 15453  4816620  90476   1438765   23196 1419.00      0      0 Dec 7 12:08
chrome   pedromonteiro 15455  4808628  88804   935311  69384 1419.00      0      0 Dec 7 12:08
chrome   pedromonteiro 15463  5103972  353756  232100726  2730267 1426.00      0      0 Dec 7 12:08
chrome   pedromonteiro 15472  4785580  83568   556558  17323 1418.00      0      0 Dec 7 12:08
nacl_helper    pedromonteiro 15391   27272   4912    7957      56      0      0      0 Dec 7 12:08
chrome   pedromonteiro 15373  1069108 254356   667034012  228986748    315.00 127.00 Dec 7 12:08
chrome   pedromonteiro 15653  1039816 58808   9374992  6471747 1934.00 1446.00 Dec 7 12:08
procstat.sh    pedromonteiro 11535   13020   3628   6249261  26707  2243542.00  8371.00 Dec 7 16:08
chrome   pedromonteiro 15439  529516  39580   13682604  14775926    1.00  42023.00 Dec 7 12:08
chrome   pedromonteiro 15394  461708  16452   32686449  25133132 100799.00  84552.00 Dec 7 12:08
chrome   pedromonteiro 15418  562900  96960   497814869  594091368 193449.00  202717.00 Dec 7 12:08
```

Todas estas opções para ordenar processos têm a possibilidade de serem executadas com a opção -r, que irá listar os processos de forma decrescente, consoante a opção com a qual é executada.

```
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -m -r 1
COMM           USER     PID     MEM     RSS    READB   WRITEB   RATER   RATEW   DATE
Discord   pedromonteiro 2784  20240820 463772  23211141 2122317 191108.00 205454.00 Dec 7 08:52
code       pedromonteiro 8758  15397588 207036  42452870 2543757      0      0 Dec 7 10:07
code       pedromonteiro 8494  15382248 188412  38164681 2072549      0      0 Dec 7 10:07
chrome    pedromonteiro 2974  5308188 556572  109366320 3189846      0      0 Dec 7 08:52
cppools-srv pedromonteiro 8684  5188928 13500   411211 512 75.00      0      0 Dec 7 10:07
Discord   pedromonteiro 2652  5138584 148972  454405649 2294725      0      0 Dec 7 08:51
chrome    pedromonteiro 6733  5032180 303992  213198392 2744821      0      0 Dec 7 10:03
chrome    pedromonteiro 3319  4959486 180856  19489837 2113948      0      0 Dec 7 08:53
code       pedromonteiro 8438  4954076 145236  38413007 13741768      0      0 Dec 7 10:07
chrome    pedromonteiro 3358  4941856 178484  10641003 3840 4254.00      0      0 Dec 7 08:53
chrome    pedromonteiro 3307  4868956 117540  555253 850711 0 0 Dec 7 08:53
chrome    pedromonteiro 8254  4829532 113140  433824 136519 1423.00      0      0 Dec 7 10:05
chrome    pedromonteiro 3313  4924664 102776  407983 244424      0      0 Dec 7 08:53
chrome    pedromonteiro 4824  4813992 88924   1660242 22942      0      0 Dec 7 08:55
```

```
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -w -r 1
COMM           USER     PID     MEM     RSS    READB   WRITEB   RATER   RATEW   DATE
chrome   pedromonteiro 2773  553172  87556  415811440 434555429 29370.00 97379.00 Dec 7 08:52
procstat.sh    pedromonteiro 19969   13020   3624   6367911  26909  2297604.00 8482.00 Dec 7 11:32
cinnamon   pedromonteiro 2158  4142036 223448  18277005 33436711 2097.00 1188.00 Dec 7 08:50
gnome-terminal- pedromonteiro 6570  565000 43272  956053 9092783 0 1056.00 Dec 7 11:25
Discord   pedromonteiro 2784  20216232 464040  23352638 2137325 5335.00 720.00 Dec 7 08:52
pulseaudio   pedromonteiro 2102  2266528 18652   2167351 1517470 432.00 432.00 Dec 7 08:50
chrome    pedromonteiro 2656  1000404 234652  605654803 231836175 324.00 100.00 Dec 7 08:51
code       pedromonteiro 8573  4719924 151492  20481002 328032 21.00 83.00 Dec 7 10:07
code       pedromonteiro 8758  15397588 206896  42652753 2544541 2812.00 37.00 Dec 7 10:07
code       pedromonteiro 8494  15382248 189268  38367381 2073341 2812.00 37.00 Dec 7 10:07
chrome    pedromonteiro 2974  5340972 553644  109473988 3190294 4459.00 25.00 Dec 7 08:52
Discord   pedromonteiro 2652  5138584 148008  457254123 2205416 133924.00 8.00 Dec 7 08:51
code       pedromonteiro 8438  4954076 145236  38413664 13742263 1.00 1.00 Dec 7 10:07
chrome    pedromonteiro 6733  5032180 302492  213241226 2744845 2847.00 1.00 Dec 7 10:03
```

Caso seja passado mais do que uma opção de sort, o programa não será executado e aparecerá uma mensagem de erro.

```
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -m -t 1
Invalid argument: multiple sort parameters detected
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -m -w 1
Invalid argument: multiple sort parameters detected
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -t -w 1
Invalid argument: multiple sort parameters detected
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -t -d 1
Invalid argument: multiple sort parameters detected
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -w -d 1
Invalid argument: multiple sort parameters detected
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -w -d -m 1
Invalid argument: multiple sort parameters detected
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -d -w -r 1
Invalid argument: multiple sort parameters detected
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ █
```

O formato da data também é validado:

```
fortnyce@fortnyce:~/Desktop/S0/avaliacoes$ ./procstat.sh -s "5 22:00 Dec" 1
Invalid Date Format! Date Format: mmm/(d)d/hh:mm
fortnyce@fortnyce:~/Desktop/S0/avaliacoes$ ./procstat.sh -s "Dec 32 22:00" 1
date: invalid date 'Dec 32 22:00'
Invalid Date Format! Date Format: mmm/(d)d/hh:mm
fortnyce@fortnyce:~/Desktop/S0/avaliacoes$ ./procstat.sh -s "Dec 1 22:00" 1
COMM           USER          PID        MEM
```

A validez dos resultados impressos foi verificada com a utilização do comando “top”.

Alguns exemplos com a chamada de múltiplos parâmetros:

```
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -m -r -c "d.*" -p 1 10
COMM           USER      PID      MEM      RSS      READB      WRITEB      RATER      RATEW      DATE
COMM           pedromonteiro 1749    77056    8248    1355559    867011      0          0       Dec 7 21:13
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ 
```

```
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -w -r -c "d.*" 10
COMM           USER      PID      MEM      RSS      READB      WRITEB      RATER      RATEW      DATE
Discord        pedromonteiro 2568   20176204  364180    8866639    6006265    7230.90    6228.70    Dec 7 21:14
pulseaudio     pedromonteiro 2022   2267068   20496    7401669    6106487    6695.90    6106.40    Dec 7 21:13
Discord        pedromonteiro 2530   477168   65384    9543970    6094015    2784.60    3462.10    Dec 7 21:14
Discord        pedromonteiro 2511   658168   195572   1503980    2693900      64.30    2735.60    Dec 7 21:14
Discord        pedromonteiro 2469   4952156   139184   118885607   1280549    126365.70   999.80    Dec 7 21:14
goa-identity-se pedromonteiro 2067   296340   7552     198282    68713      135.20    103.20    Dec 7 21:13
code           pedromonteiro 4145   15404448  207652    36311828   2984592      986.70     9.00     Dec 7 21:20
code           pedromonteiro 4221   4697356   136660   17504262    742677      2.60      2.60     Dec 7 21:20
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ 
```

```
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ ./procstat.sh -s "Dec 7 21:00" -e "Dec 7 21:20" 10
COMM           USER      PID      MEM      RSS      READB      WRITEB      RATER      RATEW      DATE
applet.py       pedromonteiro 2423   249940   32196    3342225   199193      15.20    45.60    Dec 7 21:14
at-spi2-registr pedromonteiro 1883   220792   6844     30647    43858      0          0       Dec 7 21:13
at-spi-bus-laun pedromonteiro 1873   349416   6260     20927    1341      0          0       Dec 7 21:13
bamfdaemon     pedromonteiro 2170   430484   40140    805219    75061      0          0       Dec 7 21:13
blueberry-obex- pedromonteiro 2100   433744   31480    1305649    4681      0          0       Dec 7 21:13
blueberry-tray  pedromonteiro 2181   28988    8548     210094    0          0          0       Dec 7 21:13
cat             pedromonteiro 2490   7612     768      1948      0          0          0       Dec 7 21:15
cat             pedromonteiro 2491   7612     764      16288    14340      0          0       Dec 7 21:15
chrome          pedromonteiro 2484   1271916   255004   176814058   95440764    473.20    25.20    Dec 7 21:14
chrome          pedromonteiro 2517   461708   61560    164648     21          0          0       Dec 7 21:14
chrome          pedromonteiro 2518   461740   61956    164661      56          0          0       Dec 7 21:14
chrome          pedromonteiro 2563   461740   16456    6667833    5729179      0          0       Dec 7 21:15
chrome          pedromonteiro 2615   997128   219232   1249193    559618      1.60      1.10     Dec 7 21:14
chrome          pedromonteiro 2622   555464   96432    8325272    96077774    3665.30    3059.60    Dec 7 21:14
chrome          pedromonteiro 2650   538732   33512      0          0          0          0       Dec 7 21:15
chrome          pedromonteiro 2774   512416   38336    2979555    2445430      0          0       Dec 7 21:15
pedromonteiro@pedro:~/Desktop/so-avaliacoes$ 
```

```
fortnyce@fortnyce:~/Desktop/SO/avaliacoes$ sudo ./procstat.sh -u root -c "sys" -m -r 1
COMM           USER      PID      MEM      RSS      READB      WRITEB      RATER      RATEW      DATE
COMM           root      1      168144   12140    1654567160   3339525654      0          0       Dec 7 20:37
systemd         root      435    76784    41324    1358351    43634      0          0       Dec 7 20:37
systemd-journal root      470    23820    7264    101700067    218949      0          0       Dec 7 20:37
systemd-udevd  root      1011   17008    8512    6975273    24563      0          0       Dec 7 20:37
systemd-logind root      1013   16524    6088    33155      88          0          0       Dec 7 20:37
systemd-machine root      □      0          0          0          0          0          0       Dec 7 20:37
fortnyce@fortnyce:~/Desktop/SO/avaliacoes$ 
```