universidade de aveiro

SISTEMAS OPERATIVOS

Relatório Trabalho 01 – Turma P01

Prof. José Nuno Panelas Nunes Lau

2020/2021

João Reis (98474)

Ricardo Rodriguez (98388)

Índice

Introdução 3
Metodologia 4
Primeira parte 4
DIRETÓRIO /PROC 4
ESTRUTURAS DE DADOS 4
ARMAZENAMENTO DOS PROCESSOS VÁLIDOS 5
ARMAZENAMENTO DE RCHAR E WCHAR INICIAIS 5
PROCURA E ARMAZENAMENTO DAS INFORMAÇÕES DE CADA PROCESSO 6
Segunda parte 7
MÚLTIPLOS ARGUMENTOS DE ENTRADA 7
VARIÁVEIS INICIALIZADAS 8
ATUALIZACÃO DO ARMAZENAMENTO DOS PROCESSOS VÁLIDOS 8
TRATAMENTO DE EXCEÇÕES 10
PRINT DA TABELA 10
VALIDAÇÃO DA SOLUÇÃO 11
ERROS E WARNINGS 11
TABELA DE PROCESSOS 12
CONCLUSÃO 14
ANEXO 15

Introdução

É-nos proposto, como projeto de Sistemas Operativos, desenvolver um script em bash, designado "procstat.sh", com o objetivo de apresentar algumas informações dos processos ativos no nosso computador, formatados numa tabela.

Este script permite a visualização de algumas estatísticas, tais como a quantidade de memória total e de memória residente em memória física usada pelos processos, o número total de bytes de Input/Output e a respetiva taxa de leitura/escrita, apresentada em bytes por segundo, para além de apresentar o nome do processo, o nome de utilizador que o está a executar e a data de início de cada processo.

Para calcular as taxas de leitura/escrita, é necessário introduzir, como argumento final obrigatório no script, o número de segundos (s) usados para determinar a diferença entre número de caracteres lidos e escritos nesse intervalo de tempo e, de seguida, a respetiva taxa ao dividir esses valores pelo número de segundos introduzidos.

Para além disto, o script permite uma visualização diferente da tabela consoante os argumentos de ordenação inseridos, entre as quais a ordenação por memória total (-m), por memória residente em memória física (-t), por taxa de leitura (-d) e por taxa de escrita (-w). Caso se pretenda inverter estas ordem anteriores, basta inserir o parâmetro de entrada reverse (-r). Por defeito, a tabela é ordenada por ordem alfabética do nome dos processos.

Também deverá ser possível pesquisar processos num intervalo de tempo específico (parâmetros starting date "-s" e ending date "-e") e quantos processos serão impressos na tabela (parâmetro "-p").

Ao longo deste relatório, explicamos a nossa metodologia para resolver o problema descrito.

Metodologia

Nesta parte do relatório, descrevemos a abordagem utilizada para solucionar este problema, bem como demonstrar os métodos usados para resolver os problemas que surgiram ao longo da realização do trabalho e, por consequência, a evolução do código.

Este trabalho foi divido em 2 partes principais: uma teve o objetivo de apresentar numa tabela todas as informações de cada processo formatados corretamente, a outra incidiu na ordenação e pesquisa de processos da respetiva tabela.

Primeira parte

Tal como referido anteriormente, o nosso primeiro objetivo foi conseguir apresentar a tabela com todos os dados dos processos corretamente formatados.

Para isso foram seguidos os seguintes passos:

DIRETÓRIO /PROC

Inicialmente, para resolver o problema pedido, acessamos o diretório do Linux **proc** através do comando "*cd /proc*". Este diretório, comum a todos os utilizadores do sistema operativo, é um sistema de ficheiros que possui várias informações sobre todo o sistema. Desta forma, este guarda, também, informação sobre vários processos ativos do sistema, o que permite recolher os dados cruciais para a resolução deste problema.

ESTRUTURAS DE DADOS

Para armazenar os diferentes tipos de informação que são essenciais para obtermos o resultado pretendido, declarámos quatro listas indexadas: "*processID*", "*infoProcess*", "*allRchar*" e "*allWchar*". O "*processID*", tal como vamos ver já a seguir, destina-se exclusivamente a guardar o número identificador (PID) de cada processo válido. O "*infoProcess*" é um *array* que vai armazenar todos os dados de todos os processos válidos, para depois apresentar toda a informação numa tabela. Por último, as listas "*allRchar*" e "*allWchar*" vão guardar,

respetivamente, o número de caracteres lidos da memória pelo processo e o número de caracteres escritos no disco pelo processo antes de executar o comando *sleep*.

ARMAZENAMENTO DOS PROCESSOS VÁLIDOS

Uma das primeiras fases na realização deste projeto foi destinada a armazenar todos os processos no diretório proc.

Para tal, já estando no diretório */proc*, recorremos a um *for loop* que vai iterar todos os ficheiros listados de */proc* que sejam números, ou seja, todos os processos de */proc* que são representados pelo *process ID* (PID).

Inicialmente, o primeiro for *loop* desenvolvido adicionava o valor do PID ao *array processID* caso uma série de condições se verificassem. Se os ficheiros de *status*, *io* e *comm* de um determinado PID existissem e se o ficheiro de *io* fosse legível (se tivermos permissão de leitura dos seus dados), então esse processo considerava-se válido e adicionava-se o respetivo PID ao final de *processID*, devido à variável *index*, inicializada como '0', que é incrementada a cada processo válido.

```
index=0
for k in $(ls -a | grep -Eo '[0-9]{1,5}'); do
    if [[ -f "$k/status" ]]; then
        if [[ -f "$k/io" ]]; then
            if [[ -f "$k/comm" ]]; then
                if [[ -r "$k/io" ]]; then
                    processID[index]=$k
                    ((index++))
                fi
            fi
        fi
    fi
done
```

Fig. 1 | *For loop* inicial que armazena todos os PID's pretendidos

ARMAZENAMENTO DE RCHAR E WCHAR INICIAIS

Noutro ciclo for, guarda-se nas variáveis var1 e var2 as linhas com informação sobre os valores de *rchar* e *wchar* de cada processo em *processID*, sendo que as variáveis *rchar* e *wchar* são variáveis que vão armazenar exclusivamente os caracteres numéricos de var1 e var2, tal como pretendemos. De seguida, armazena-se *rchar* e *wchar* nos *arrays allRchar* e *allWchar*, respetivamente. Este passo é importante porque, posteriormente, estes dados serão necessários para o cálculo da taxa de leitura/escrita.

De seguida, utilizamos o comando sleep com a variável sleepTime para parar o processo pela quantidade de segundos introduzidas.

PROCURA E ARMAZENAMENTO DAS INFORMAÇÕES DE CADA PROCESSO

Após o *sleep* terminar, recorremos novamente a um *for loop* para todos os PID's, onde todas as informações pretendidas de cada PID serão adicionadas a "*infoProcess*".

Dentro do *for loop* guardamos os valores de *VmSize* e de *VmRSS*, que são produtos da remoção dos caracteres não numéricos de var1 e var2. Para calcular as taxas de leitura/escrita, busca-se o respetivo valor inicial de *rchar* e *wchar* do processo nas listas *allRchar* e *allWchar*. De seguida, guarda-se em *rchar2* e *wchar2* os novos valores de caracteres lidos e escritos após a execução do comando *sleep*.

Agora, a taxa de escrita/leitura calcula-se ao subtrair o *rchar2*/*wchar2* atual pelo *rchar*/*wchar* antigo, seguidos de uma divisão deste resultado por *sleepTime* através duma calculadora precisa (*bc*). Note-se que estes valores de taxa de escrita/leitura vão ter, no máximo, duas casas decimais *("scale=2; …"*) e que todos os números que não tiverem um '0' antes do ponto vão passar a tê-lo através da substituição das expressões "*#.*" por "*0.*".

Posteriormente, guarda-se o nome do processo (*comm*), substituindo todos os espaços em branco por um sublinhado, a data (*date*) do processo como string em língua inglesa e, por último, o utilizador (*user*) do processo.

Terminado já o cálculo de todas os dados do processo, adiciona-se ao *array* "*infoProcess*" todos estas estas informações.

Por cada iteração realizada, incrementa-se os valores nulos iniciais de *index*, para permitir aceder aos arrays *allRchar* e *allWchar.*

A nossa primeira parte está finalmente concluída, dado que conseguíamos visualizar todos os processos e as suas respetivas informações depois de usar o comando *printf* com a formatação adequada.

Segunda parte

A segunda fase incide na ordenação e formatação da tabela.

Contudo, alguns problemas surgiram ao tentar implementar a pesquisa e ordenação de processos. Um deles foi como iriamos imprimir só os processos com um determinado utilizador, com uma determinada letra ou subtring contida no seu nome, com um intervalo específico de datas. Implementar tudo isto de uma forma o máximo eficiente.

MÚLTIPLOS ARGUMENTOS DE ENTRADA

Quando executamos o script *procstat.sh*, podemos inserir uma combinação diferente de parâmetros de entrada que permitem visualizar a tabela de uma forma diferente, consoante o pretendido. Para tal, usamos o *while getopts*, uma função embutida que permite processar a informação introduzida no terminal pela pessoa que executa o script.

Os parâmetros de entrada aceites neste script são os que estão contidos entre as aspas. Se um parâmetro estiver delimitado à sua direita por dois pontos (:), significa que este precisa de ser seguido por um argumento. Caso contrário, este não precisará de qualquer argumento e desempenhará um papel específico. A variável *opt* vai guardar a opção atual que está a ser analisada pelo *getopts*.

De seguida, usamos um case *switch* para desempenhar funções diferentes consoante a variável *opt* atual:

- Se esta for **p** (número de processos que pretendemos visualizar), guardamos o argumento que se segue em *numProcess* e verificamos se o mesmo é um inteiro positivo;
- Se for **u** (nome do utilizador), guardamos o próximo argumento em *user*;
- Se for **c** (nome do processo), armazena-se o valor seguinte em *comm*;
- No caso de ser **s** (data de início do intervalo) ou **e** (data de fim do intervalo) e os seus valores não forem inválidos, então guarda-se em *sDate* ou *eDate*, respetivamente, a marca temporal (*timestamp*) da data de início e da data de fim do intervalo;
- Caso seja **m**, **t**, **d** ou **r**, guardamos na variável *order* o valor da coluna que queremos ordenar quando formos imprimir a tabela e incrementamos uma unidade à variável *total*, inicializada com valor nulo. Isto é útil uma vez que só podemos introduzir uma destas variáveis e, caso o valor de *total* for superior a 1, o script acaba com uma mensagem de erro devido ao *if* que se encontra depois deste *while loop*.

- Se o parâmetro de entrada for **r**, a variável *reverse* terá o valor '1'. Assim, a tabela será apresentada de forma alfabeticamente decrescente caso não for introduzido nenhum dos argumentos anteriores ou, caso se selecionou um dos parâmetros anteriores, a tabela será apresentada segundo o argumento introduzido de forma crescente.
- Caso tenha sido introduzido um argumento diferente dos apresentados anteriormente, o script será encerrado com uma mensagem de "*command not found*".

É importante, também, introduzir o comando "*shift $((OPTIND - 1))*", uma vez que queremos excluir os parâmetros de entrada já processados pelo *getopts*.

VARIÁVEIS INICIALIZADAS

Caso alguns parâmetros de entrada não forem selecionados, dispomos já de uma série de variáveis pré-inicializadas antes do *while getopts*. Ambas as variáveis *reverse* e *order* são inicializadas com o valor nulo, tal como pretendemos.

Se o parâmetro c não for escolhido, *comm* terá um valor que permite demonstrar todos os tipos de nomes de processo diferentes, bem como *user* caso não seja introduzido o argumento u. O *numProcess*, o número de processos a serem impressos, inicializado como "null", para mais tarde ser atualizado.

De seguida, o *sDate* será inicializado com um *timestamp* nulo, enquanto que o *eDate* terá o *timestamp* atual, tal como pretendemos. No final, inicializamos o contador total com o valor nulo e guardamos em *sleepTime* o último argumento introduzido e obrigatório, o número de segundos que pretendemos usar para calcular *rater* e *ratew*, para ser posteriormente usado.

ATUALIZACÃO DO ARMAZENAMENTO DOS PROCESSOS VÁLIDOS

Para tentar solucionar o problema descrito na introdução da 2ª parte, achamos pertinente adicionar mais algumas condições para adicionar exclusivamente os processos que são acessíveis e, ao mesmo tempo, que correspondem aos argumentos introduzidos quando o script é executado.

Desta forma, atualizamos o ciclo *for* da figura 1 com mais algumas condições, com a intenção de salvaguardar apenas os PID's que satisfazem as opções inseridas como argumento, para além das condições já previamente estabelecidas.

Assim, decidimos verificar se existiam as informações VmSize e VmRSS em */status* e rchar e wchar *em /io* para serem posteriormente guardadas. Caso não exista qualquer uma dessas informações, esse PID é ignorado e avançamos para o próximo.

Vamos também verificar se o nome do processo atual (*pComm*) e o utilizador que está a correr esse processo (*pUser*) coincidem com os valores de *comm* e *user* introduzidos. Igualmente, verificamos se a data de início do processo (*startDate*), guardada inicialmente como *string* em linguagem *EN/US* e, de seguida, transformada numa marca temporal (*dateTS*), está contida no intervalo entre *sDate* e *eDate*, parâmetros de entrada do script. Caso todas estas condições anteriores se verifiquem, então o processo é válido e será adicionado ao *array processID*, incrementando-se o valor de *index*.

Por exemplo, *"./procstat.sh -u root 10"* só irá guardar os PID's cujo respetivo *user* coincida com o *user* indicado como argumento (*root*). Assim, estas condições abrangem o nome do utilizador, o nome do processo e o intervalo de tempo.

```
index=0
for k in $(ls -a | grep -Eo '[0-9]{1,5}'); do
    if [[ -f "$k/status" && -f "$k/io" && -f "$k/comm" ]]; then
        if [[ -r "$k/status" && -r "$k/io" && -r "$k/comm" ]]; then
            if $(cat $k/status | grep -q 'VmSize\|VmRSS') && $(cat $k/io | grep -q 'rchar\|wchar') ; then

                pComm=$(cat $k/comm)

                pUser=$(ps -o user= -p $k)

                LANG=en_us_8859_1
                startDate=$(ps -o lstart= -p $k)
                startDate=$(date +"%b %d %H:%M" -d "$startDate")
                dateTS=$(date --date="$startDate" +"%s")

                if [[ ($pComm =~ $comm) && ($pUser == $user) && ($dateTS -gt $sDate) && ($dateTS -lt $eDate) ]]; then
                    if ! [[ "${processID[@]}" =~ "$k" ]]; then
                        processID[index]=$k
                        ((index++))
                    fi
                fi
            fi
        fi
    fi
done
```

Fig. 2 | Ciclo *for* da figura 1 atualizado de modo a armazenar os PID's de acordo com as opções inseridas

PRINT DA TABELA

Sendo que temos armazenados todos os processos válidos com a respetiva informação correta, bem como os respetivos de argumentos de entrada introduzidos quando se corre o *script*, podemos agora imprimir os resultados de forma apropriada.

Para imprimir a tabela, usamos comandos diferentes para condições diferentes:

- A variável *order* ser diferente do valor por omissão ('1') e não for inserido o parâmetro de *reverse* ('0'), onde ordenamos a coluna pretendida de forma decrescente;
- A variável *order* ser diferente do valor por omissão ('1') e for inserido o parâmetro de *reverse* ('1'), onde ordenamos a coluna pretendida de forma crescente;

- A variável *order* ser o valor padrão (‘1’) e for inserido o parâmetro de *reverse* (‘1’), onde ordenamos a coluna pretendida de forma descrescente alfabeticamente (usando -f para sortear de forma *case-insensitive*);
- Caso nenhuma destas condições se verifique, então significa que ambas as variáveis *order* e *reverse* têm os valores padrão e a lista é apresentada de forma crescente alfabeticamente;

Cada um destes comandos referidos anteriormente seguem o comando “*head –n ${numProcess}*” de forma a apresentar um número limitado de processos caso tenha o parâmetro p tenha sido introduzido.

TRATAMENTO DE EXCEÇÕES

Depois da ordenação e formatação dos processos na tabela, o último passo foi tratar possíveis erros que poderiam acontecer em determinadas ações, por exemplo, não colocar o argumento obrigatório.

Alguns exemplos onde foram tratadas as exceções são:

- Comandos diferente dos disponíveis;
- Comandos que não são compatíveis entre si;
- Inserir uma data, tanto de início como de fim, inválida;
- O número processos a serem impressos seja superior ao número de processos total;
- A data de início seja superior à data de fim;
- O sleep time não seja um número inteiro positivo ou não exista.

Para cada um destes casos, aparecerá uma mensagem de erro e sairá do programa, tal como iremos verificar no tópico “Validação da solução”.

Termina assim o desenvolvimento final do *script* procstat.sh.

VALIDAÇÃO DA SOLUÇÃO

Realizamos alguns testes ao código para averiguar a existência de possíveis erros e falhas. Os resultados foram os seguintes:

ERROS E WARNINGS

```
joao@reis:~/SO/projeto$ ./procstat.sh
Error: Invalid options (sleep time is not a positive integer or don't exist)
```

Erro por não conter o argumento obrigatório, o nº de segundos

```
joao@reis:~/SO/projeto$ ./procstat.sh -m -r -c "d.*"
Error: Invalid options (sleep time is not a positive integer or don't exist)
```

Erro por não ter o sleep time, desta vez com mais parâmetros inseridos

```
joao@reis:~/SO/projeto$ ./procstat.sh -t -m 10
Error: incompatible commands
```

Erro por conter dois ou mais comandos imcompatíveis (-m, -t, -d, -w)

```
joao@reis:~/SO/projeto$ ./procstat.sh -s "Dec 06 23:00" -e "Dec 06 22:00" 10
Error: Invalid options (ending date smaller or equal than the starting date)
```

Erro porque a data de inicio é superior à data de fim

```
joao@reis:~/SO/projeto$ ./procstat.sh -p -2 10
Error: Invalid options (number of -p must be a positive integer)
```

Erro porque o número de processos a ser imprimidos (-p) têm de ser um número inteiro positivo

```
joao@reis:~/SO/projeto$ ./procstat.sh -s "data" 10
Error: Invalid starting date
```

Erro porque a data inserida não é válida

```
joao@reis:~/SO/projeto$ ./procstat.sh -m -q 10
Error: command not found
```

Erro por ter um comando que não existe (-q)

```
joao@reis:~/SO/projeto$ ./procstat.sh 0
Error: Invalid options (sleep time is not a positive integer or don't exist)
```

Erro porque o nº de segundos (sleep time) é nulo

```
joao@reis:~/SO/projeto$ ./procstat.sh -p 10000 10
Error: You selected a greater number of processes than the available ones
```

Erro porque o nº de processos a serem imprimidos é maior do que os disponíves

```
joao@reis:~/SO/projeto$ ./procstat.sh -p 0 10
Warning: No valid processes found
```

Como é pedido para imprimir um total de 0 processos, aparece um warning para avisar que não há processos encontrados

```
joao@reis:~/SO/projeto$ ./procstat.sh -u user 10
Warning: No valid processes found
```

Como não tem nenhum usurário “user”, aparece novamente o warning para avisar que não há processos válidos encontrados

TABELA DE PROCESSOS

```
joao@reis:~/SO/projeto$ ./procstat.sh 10
```

Output: Output 1 (Em anexo)

Impressão de todos os processos após um sleep de 10 segundos

```
joao@reis:~/SO/projeto$ ./procstat.sh -m -p 10 10
```

Output: Output 2 (Em anexo)

Impressão de 10 processos por ordem decrescente de memória após um sleep de 10 segundos

```
joao@reis:~/SO/projeto$ ./procstat.sh -c "chr.*" -d -r 10
```

Output: Output 3 (Em anexo)

Impressão de todos os processos que contenham “chr” no seu nome, ordenados crescentemente pelo seu RATER após um sleep de 10 segundos

```
joao@reis:~/SO/projeto$ ./procstat.sh -s "Dec 07 13:00" -e "Dec 07 15:00" 10
```

Output: Output 4 (Em anexo)

Impressão de todos os processos que começaram entre as datas indicadas, após um sleep de 10 segundos

```
joao@reis:~/SO/projeto$ ./procstat.sh -e "Dec 07 11:15" -r -u joao -c "gnome.*" -t 10
```

Output: Output 5 (Em anexo)

Impressão de todos os processos com “gnome” no seu nome do utilizador “joao”, que começaram até à data indicada, por ordem decrescente de RSS, após um sleep de 10 segundos

```
joao@reis:~/SO/projeto$ ./procstat.sh -c "gnome.*" -t -r -u joao -e "Dec 07 11:15" 10
```

Output: Output 6 (Em anexo)

As opções são as mesmas que o exemplo anterior, porém a ordem em que aparecem varia. Como podemos verificar o seu Output não varia em relação ao Output anterior (Output 5)

CONCLUSÃO

A resolução do problema descrito na introdução foi um sucesso, dado que os resultados coincidem com o que era espectável.

Tivemos vários obstáculos/problemas menos complexos aos apresentados neste relatório. Contudo, para nós, todo este trabalho foi um desafio do qual saímos satisfeitos com o resultado final.

Para além disso, enriqueceu o nosso conhecimento sobre *bash* e aumentou a curiosidade sobre a mesma, uma vez que foi uma pesquisa intensa apoiada no material prático e teórico fornecido nas aulas de Sistemas Operativos.

ANEXOS

```
joao@reis:~/SO/projeto$ ./procstat.sh 10
COMM            USER        PID         MEM         RSS       READB       WRITEB       RATER       RATEW             DATE
at-spi-bus-laun joao       1800      305548        6092       22375        1091           0           0    Dec 07 11:02
at-spi2-registr joao       1885      162820        7460       31159       98314           0           0    Dec 07 11:02
bash            joao       5446       11108        5432   681497750     1190523           0           0    Dec 07 11:17
cat             joao       2256        8228         584        4292           0           0           0    Dec 07 11:02
cat             joao       2257        8228         584        4496         204           0           0    Dec 07 11:02
chrome          joao       2250      651840      220500   219818351    61676811      470.20    11874.00    Dec 07 11:02
chrome          joao       2261      276056       63604      179881          21           0           0    Dec 07 11:02
chrome          joao       2263      276056       64244      179895          56           0           0    Dec 07 11:02
chrome          joao       2268      276056       16224     2587332     5257812           0           0    Dec 07 11:02
chrome          joao       2288      586508      126972      633233     3141387      138.50      137.80    Dec 07 11:02
chrome          joao       2292      369888       96852    30255728    56454271      464.80    52018.00    Dec 07 11:02
chrome          joao       2311      336884       36148     5927874      458641           0      123.60    Dec 07 11:02
chrome          joao       2364     4601888       86496      139506         232           0           0    Dec 07 11:02
chrome          joao       2890     8870864      122584      267584      180639           0           0    Dec 07 11:03
chrome          joao       4478     4648824      184396     2114136      398936      1429.00           0    Dec 07 11:04
chrome          joao       4582     9103944      369436    241479027     5954616    113074.10        5.80    Dec 07 11:04
chrome          joao       4543     4640020       95144      748347       43372           0           0    Dec 07 11:04
chrome          joao       4556     4628524       89428      702587         308           0           0    Dec 07 11:04
chrome          joao       4796     4787864      226120    11881580     4912598      1003.20        0.10    Dec 07 11:07
chrome          joao       5049     4581140       54148       61378           9           0           0    Dec 07 11:07
code            joao       3309     4715036      135400     5479807     5025859       16.90       51.30    Dec 07 11:03
code            joao       3372      186240       40832       81837          20           0           0    Dec 07 11:03
code            joao       3373      186240       40000       81837           4           0           0    Dec 07 11:03
code            joao       3408      426288       97988      249109      805431        0.50           0    Dec 07 11:03
code            joao       3425      257064       59344      750283       71530           0           0    Dec 07 11:03
code            joao       3441    15241076      216756    39528090     2958297      1015.40       43.10    Dec 07 11:03
code            joao       3491     4531564      128020    15055797      217915       30.10       18.00    Dec 07 11:03
code            joao       3521     4457108       67184      936421         859           0           0    Dec 07 11:03
code            joao       3620     4618048      114264    10105038     4354552      140.70           0    Dec 07 11:03
dbus-daemon     joao       1695        9380        6592      122891        7975           0           0    Dec 07 11:02
dbus-daemon     joao       1805        7580        4436       42401        4244           0           0    Dec 07 11:02
dconf-service   joao       1919      156356        5856       35964       19728           0           0    Dec 07 11:02
Discord         joao       2452     4749908      146200   145048223      324362    49935.80        4.40    Dec 07 11:02
Discord         joao       2456      202332       45092       95981          20           0           0    Dec 07 11:02
Discord         joao       2457      202332       45472       95982          50           0           0    Dec 07 11:02
Discord         joao       2459      202332        9532           1           4           0           0    Dec 07 11:02
Discord         joao       2495      423224       94168      231545      125483        1.60       22.70    Dec 07 11:03
Discord         joao       2504      275088       66644     7515118     1583571     2908.40      452.20    Dec 07 11:03
Discord         joao       2574    19751552      227652     8329030      575980     2490.40      193.80    Dec 07 11:03
Discord         joao       2605      519812       52004      133492        7902           0           0    Dec 07 11:03
evolution-addre joao       1925      747864       30460      128703       37361           0           0    Dec 07 11:02
evolution-alarm joao       2013      638156       59772      385874        9209           0           0    Dec 07 11:02
evolution-calen joao       1910      839612       30024      152706        7850           0           0    Dec 07 11:02
evolution-sourc joao       1900      391196       26200      138699        2257           0           0    Dec 07 11:02
gdm-x-session   joao       1689      164200        6896       38799         742           0           0    Dec 07 11:02
gjs             joao       1940     2930780       27232       58381        1409           0           0    Dec 07 11:02
gnome-calendar  joao       5322      858476       56228      377776        9354           0           0    Dec 07 11:17
gnome-session-b joao       1707      188652       14348     5101300       22961           0           0    Dec 07 11:02
gnome-session-b joao       1822      485940       17180      660970       21367           0           0    Dec 07 11:02
gnome-session-c joao       1815       90188        4288       24460          64           0           0    Dec 07 11:02
gnome-shell     joao       1836     4727956      288396    13985472    10580255       30.40      406.40    Dec 07 11:02
gnome-shell-cal joao       1893      581548       20372      127243        5098           0           0    Dec 07 11:02
gnome-terminal- joao       5327      814156       50284      971704     6402582           0      790.40    Dec 07 11:17
goa-daemon      joao       1647      546628       30772      123785        3458           0           0    Dec 07 11:02
goa-identity-se joao       1658      315264        9228      128109       11281           0           0    Dec 07 11:02
gsd-a11y-settin joao       1964      310192        6736       23450        2105           0           0    Dec 07 11:02
gsd-color       joao       1965      574996       26904      303894       15974           0           0    Dec 07 11:02
gsd-datetime    joao       1966      374140       16388       84778        1740           0           0    Dec 07 11:02
gsd-disk-utilit joao       1998      231792        6108       25103        2052           0           0    Dec 07 11:02
gsd-housekeepin joao       1968      312388        8200      891581        2449           0           0    Dec 07 11:02
gsd-keyboard    joao       1970      342576       25460      287872        7804           0           0    Dec 07 11:02
gsd-media-keys  joao       1971     1021952       30104     2036067     1129614           0           0    Dec 07 11:02
gsd-power       joao       1974      410988       26388      292734       12214           0           0    Dec 07 11:02
gsd-print-notif joao       1976      248840       11800       54336        1989           0           0    Dec 07 11:02
gsd-printer     joao       2003      342456       15260       77391        1000           0           0    Dec 07 11:02
gsd-rfkill      joao       1979      457308        6280       24802        3481           0           0    Dec 07 11:02
gsd-screensaver joao       1981      235780        6372       22172        1852           0           0    Dec 07 11:02
gsd-sharing     joao       1989      465312       11036       49650       30707        4.80        8.00    Dec 07 11:02
gsd-smartcard   joao       1994      389552        8868       34909        2037           0           0    Dec 07 11:02
gsd-sound       joao       1996      319876        9024       44178        2073           0           0    Dec 07 11:02
gsd-usb-protect joao       1999      385492        7252       26098        2706           0           0    Dec 07 11:02
gsd-wacom       joao       2003      341852       24680      425475        6864           0           0    Dec 07 11:02
gsd-wwan        joao       2006      314516        8976       28402        2136           0           0    Dec 07 11:02
gsd-xsettings   joao       2008      343108       26240      464868       10609           0           0    Dec 07 11:02
gvfs-afc-volume joao       1609      310996        9172       32580        3470           0           0    Dec 07 11:02
gvfs-goa-volume joao       1643      236144        6324       22612        3742           0           0    Dec 07 11:02
gvfs-gphoto2-vo joao       1675      238336        6852       37310        3380           0           0    Dec 07 11:02
gvfs-mtp-volume joao       1664      235936        6284       33190        3398           0           0    Dec 07 11:02
gvfs-udisks2-vo joao       1631      314680        9884     1669805        4939           0           0    Dec 07 11:02
gvfsd           joao       1624      239976        7904       40999        5160           0           0    Dec 07 11:02
gvfsd-fuse      joao       1629      378336        6500       32412         888           0           0    Dec 07 11:02
gvfsd-metadata  joao       3758      162420        6508       26812         669           0           0    Dec 07 11:03
gvfsd-trash     joao       1952      314500        8220      109291        3432           0           0    Dec 07 11:02
ibus-daemon     joao       1861      311640        8820       48049        5854           0           0    Dec 07 11:02
ibus-dconf      joao       1865      237032        7584       23924         697           0           0    Dec 07 11:02
ibus-engine-sim joao       2116      163188        7424       26692        1688           0           0    Dec 07 11:02
ibus-extension- joao       1866      273088       29036      864243        7189           0           0    Dec 07 11:02
ibus-portal     joao       1872      230844        7552       30184       12769           0           0    Dec 07 11:02
ibus-x11        joao       1868      194464       24788      279131        6100           0           0    Dec 07 11:02
nacl_helper     joao       2265       10712        4252       10212          50           0           0    Dec 07 11:02
nautilus        joao       8926     1064528       73600     6336773     5686796           0           0    Dec 07 11:22
oosplash        joao      13772       90312        5020      118590         498           0           0    Dec 07 11:43
procstat.sh     joao      17612        9632        3704    92951135       14235     2043437.90     7972.00    Dec 07 11:54
pulseaudio      joao       1599     2793860       20444      752869      373852      108.00      108.00    Dec 07 11:02
snap-store      joao       2076     1229924      214556    14579941     5439835           0           0    Dec 07 11:02
soffice.bin     joao      13789     1589776      291912    25487442    35158230           0           0    Dec 07 11:43
systemd         joao       1586       19364       10652    59383206     3827352           0           0    Dec 07 11:02
tracker-miner-f joao       1601      659976       25736     7929171       31977           0           0    Dec 07 11:02
update-notifier joao       3761      417876       28768   570579247    15306134           0           0    Dec 07 11:03
xdg-document-po joao       2117      457764        6800       40581       10930           0           0    Dec 07 11:02
xdg-permission- joao       1889      235936        4516       20620         603           0           0    Dec 07 11:02
Xorg            joao       1691      950372       82192    16891228    50085413       54.40      695.20    Dec 07 11:02
```

Output 1

```
joao@reis:~/SO/projeto$ ./procstat.sh -m -p 10 10
COMM            USER        PID         MEM         RSS       READB       WRITEB       RATER       RATEW            DATE
Discord         joao       2574    30931436      786764    43513838    10629491     2486.90      218.40    Dec 07 11:03
code            joao       3441    19461164      229300    63231811     5616267     1157.40       43.10    Dec 07 11:03
chrome          joao       4502     9185284      383432   642975721     7245964   113661.90        1.20    Dec 07 11:04
chrome          joao       2890     8951436      134148     2357800      193773           0           0    Dec 07 11:03
spotify         joao      77903     6179728      273916      589666        2718     1422.00           0    Dec 07 15:36
Discord         joao       2452     5292304      153816   716108626     1182859    50872.90        4.50    Dec 07 11:02
gnome-shell     joao       1836     4790256      314476   107474767    29608209      189.60     1474.90    Dec 07 11:02
chrome          joao       4796     4787608      221160    34862487     4914425     1290.70        0.10    Dec 07 11:07
code            joao       3369     4715036      139320     6244729     6855701       15.20       20.00    Dec 07 11:03
chrome          joao      40220     4676540      126428      739792      476990           0           0    Dec 07 12:46
```

Output 2

```
joao@reis:~/SO/projeto$ ./procstat.sh -c "chr.*" -d -r 10
COMM            USER        PID         MEM         RSS       READB       WRITEB       RATER       RATEW            DATE
chrome          joao       2261     276056       63604      179881          21           0           0    Dec 07 11:02
chrome          joao       2263     276056       64244      179895          56           0           0    Dec 07 11:02
chrome          joao       2268     276056       16224    23549817    28210244           0           0    Dec 07 11:02
chrome          joao       2311     344956       37472    13790943     8356842           0           0    Dec 07 11:02
chrome          joao       2364    4610084       86956      568963         399           0           0    Dec 07 11:02
chrome          joao       2890    8951436      134396     2389326      193773           0           0    Dec 07 11:03
chrome          joao       4543    4643100       98708     2272159       91697           0           0    Dec 07 11:04
chrome          joao      21266    4675024      133556      822242      337713           0           0    Dec 07 12:01
chrome          joao      21784     559004       59720      182499       42854           0           0    Dec 07 12:06
chrome          joao      40220    4676540      126428      752733      476990           0           0    Dec 07 12:46
chrome          joao      41044    4615432       81084      301909          52           0           0    Dec 07 13:02
chrome          joao      41149    4617864       87112      291810          63           0           0    Dec 07 13:03
chrome          joao      65709    4646828      101860      116942        3053           0           0    Dec 07 15:27
chrome          joao      77710    4581140       53856       25863          12           0           0    Dec 07 15:35
chrome          joao       2250    1082108      268572   830401211   230653280        7.30        5.70    Dec 07 11:02
chrome          joao       2292     367844       98768    67391882   269464587       13.70        0.10    Dec 07 11:02
chrome          joao       2288     644332      143712     1785313     8793452      125.90      125.60    Dec 07 11:02
chrome          joao       4478    4648728      105800     6924701      400807      143.10           0    Dec 07 11:04
chrome          joao       4796    8989592      223896    35903069     4914507      860.40           0    Dec 07 11:07
chrome          joao      41135    4669464      113468     9421410       93391      865.50        2.70    Dec 07 13:03
chrome          joao      21302    4667568      122444     9561039      351086     1006.60           0    Dec 07 12:02
chrome          joao       4502    9186692      402436   683521083     7255353     2109.90        0.90    Dec 07 11:04
```

Output 3

```
joao@reis:~/SO/projeto$ ./procstat.sh -s "Dec 07 13:00" -e "Dec 07 15:00" 10
COMM            USER        PID         MEM         RSS       READB       WRITEB       RATER       RATEW            DATE
chrome          joao      41044    4615432       81084      330597          52           0           0    Dec 07 13:02
chrome          joao      41135    4669464      112684    10139903       95569      866.20        2.80    Dec 07 13:03
chrome          joao      41149    4617864       87108      320478          63           0           0    Dec 07 13:03
code            joao      41298    4431496       64468      333386        1389           0           0    Dec 07 13:06
gvfsd-dnssd     joao      41753     315248        8556       35859        4088           0           0    Dec 07 13:08
gvfsd-network   joao      41724     388360        8788       29789        2729           0           0    Dec 07 13:08
```

Output 4

```
joao@reis:~/SO/projeto$ ./procstat.sh -e "Dec 07 11:15" -r -u joao -c "gnome.*" -t 10
COMM            USER        PID         MEM         RSS       READB       WRITEB       RATER       RATEW            DATE
gnome-session-c joao       1815       90188        4288       24460          64           0           0    Dec 07 11:02
gnome-session-b joao       1707      188652       14348     5101300       22961           0           0    Dec 07 11:02
gnome-session-b joao       1822      485940       17180      675178       43559           0           0    Dec 07 11:02
gnome-shell-cal joao       1893      581548       20372      157875        9634           0           0    Dec 07 11:02
gnome-shell     joao       1836     4791064      315800   210510543    41211040      167150.00      823.60    Dec 07 11:02
```

Output 5

```
joao@reis:~/SO/projeto$ ./procstat.sh -c "gnome.*" -t -r -u joao -e "Dec 07 11:15" 10
COMM            USER        PID         MEM         RSS       READB       WRITEB       RATER       RATEW            DATE
gnome-session-c joao       1815       90188        4288       24460          64           0           0    Dec 07 11:02
gnome-session-b joao       1707      188652       14348     5101300       22961           0           0    Dec 07 11:02
gnome-session-b joao       1822      485940       17180      675178       43559           0           0    Dec 07 11:02
gnome-shell-cal joao       1893      581548       20372      157875        9634           0           0    Dec 07 11:02
gnome-shell     joao       1836     4789048      314644   214527138    41285228      167146.00      806.40    Dec 07 11:02
```

Output 6