Sistemas Operativos

2022/2023

Taxas de Leitura/Escrita de processos em bash

Hugo Correia 108215 P6

Sara Almeida 108796 P6

Professor:

Nuno Lau (nunolau@ua.pt)

Índice

1 – Introdução

No contexto da disciplina de Sistemas Operativos, foi-nos proposta a realização deste projeto que tem como objetivo a visualização e tratamento dos processos a decorrer atualmente no nosso computador, mais especificamente, o número de bytes escritos e lidos num certo intervalo de tempo bem como as suas respetivas taxas de leitura e escrita (*bytes/segundo*).

Para a realização deste trabalho foi utilizado o **VSCODE**, um editor conhecido por ambos os elementos deste grupo e com o qual estamos habituados a trabalhar no seguimento de outras disciplinas. A execução de todo o trabalho foi sendo suportada por um repositório de GitHub, para facilitar a sua manipulação por ambos os elementos do grupo e para serem salvas cópias de segurança caso haja algum erro ao longo do processo do trabalho.

O script *rwstat.sh* terá como base a matéria lecionada nas aulas práticas de Sistemas Operativos onde foi abordada a base da programação em bash-shell. Neste caso, foi aplicada na listagem de processos que se encontram a correr no computador do utilizador. De acordo com o guião, haverá várias opções passadas como argumento pelo utilizador que permitirão filtrar a informação listada. A nossa ideia inicial passou por armazenar todas as informações num **array associativo** (função *listar_processos()* em que a cada **key** seria o **$pid** associado a todas as informações relativas ao processo. Antes mesmo do programa fazer este processamento passará por uma validação de argumentos (função *arguments_validation()*). Após a criação do array, o mesmo será filtrado (função *PID_filter()*) e por fim escrito no terminal com a ordem devida e desejada pelo utilizador (função *sort_process()*)

A realização deste projeto permitiu claramente o alargamento do nosso conhecimento relativo à linguagem de bash e clarificou a nossa perceção dos processos do sistema operativo Linux.

2 – Estrutura do código

2.1 - Inicializar o Script

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

Figura 1 - linguagem utilizada e diretório

Primeiramente, iniciámos o script dizendo à shell que o interpretador que está a ser utilizado é ***"# !/bin/bash"***. Como iremos estar a trabalhar na diretoria **/proc** ao longo de todo o script, decidimos que o mais correto a fazer seria direcionar o nosso script para a mesma logo de início.

2.1. – Declaração de variáveis globais

```
declare -A READBI  #Array para armazenar os valores iniciais de READB
declare -A WRITEBI #Array onde irá ser guardado o valor inicial dos WriteB
declare -A PID_array=() #Array onde irá ser guardado o PID de todos os processos

#----------------------------------Declaracao das variaveis---------------------------------
#Valor default da filtragem por regex '-c'
flag_c="NULL"
#Valor default da filtragem por user '-u'
flag_u="NULL"
#Valor default do gama de pids minima '-m'
flag_m="NULL"
#Valor default do gama de pids maxima '-M'
flag_M="NULL"
#Valor default do numero de processos '-p'
flag_p="NULL"
#("" or "-r") Se a flag reverse não for acionadada,
# os processos são ordenados por ordem inversa a taxa de leitura
reverse=""
#(NULL or -w) Se a flag -w for acionada,
# os processos são ordenados por ordem inversa a taxa de escrita
flag_w="NULL"
#data minima default '-s'
min_date="NULL"
#data maxima default '-e'
#se não for passado argumento -e com data máxima, o programa assume a data atual
max_date="NULL"
min_date_s="NULL"
max_date_s="NULL"
#associar o último argumento a uma variável chamada last para ser mais fácil de manipular e utilizar
last=${@: -1}
n_argumentos=0 #numero de argumentos passados
n_opcoes=0 #numero de opções passadas

#regex para a flag -c
re='^[0-9]+([.][0-9]+)?$'
#----------------------------------Fim da declaracao das variaveis--------------------------------
```

Figura 2 - Declaração de variáveis

De seguida, decidimos usar **arrays associativos**, para o tratamento da informação (Fig.2), por estes serem estruturas de dados que permitem associar determinados valores a "keys" facilitando a manipulação dos dados. Sendo assim, começámos por inicializar os arrays *READBI*, *READBF*, *WRITEBI*, *WRITEBF* e *PID_array*.

Algumas variáveis foram inicializados com valor "NULL" (valor default) para ser mais fácil a sua manipulação, ou seja, dos argumentos de entrada do programa. À medida que as opções forem sendo passadas como argumento pelo utilizador os seus valores vão sendo alterados para posteriormente serem lidos e utilizados pelas várias funções do programa.

Mais detalhadamente, temos as variáveis *last*, *n_argumentos, n_opcoes* e *re.* A variável *last* irá guardar o último argumento, para facilitar a sua manipulação, visto que é um argumento muito importante e muito utilizado. As variáveis *n_argumentos* e *n_opcoes* serão utilizadas mais tarde, ao longo do tratamento das opções, com a finalidade de garantir que o número de argumentos e as regras em relação a cada opção são respeitadas. A variável *re* tem um valor de uma expressão que representa uma expressão geral regex que será útil mais tarde no tratamento da opção *-c.*

2.2. – Funções

Grande parte do nosso código fonte da solução encontrada é baseado em funções, por questões de organização e de mais fácil manipulação de dados.

2.2.1 – Função num_Int()

```
#esta função é chamada sempre que for preciso validar se um argumento é um número inteiro
num_Int() {
    if [[ $1 =~ ^[0-9]+$ ]]; then # ^[0-9]+$ - verifica se o argumento é um número inteiro
        return 0 #retorna 0 se for um número inteiro
    else
        return 1 #retorna 1 se não for um número inteiro
    fi
}
```

Figura 3 - Função num_Int()

Neste campo, começámos por criar uma função que será chamada sempre que for necessário validar se um argumento é ou não um **número inteiro** (Fig.3), pois vimos ser útil ao longo da resolução do guião.

2.2.2 – Função validate_date()

```
#esta função é chamada sempre que são passadas as opções -s e/ou -e para validar o formato das datas passadas como argumentos
validate_date() {

    #data passada como argumento
    date1=$1;
    #data passada como argumento, mas formatada com o formato dado no guião do projeto
    date2=$(date "+%b %d %H:%M" -d "$date1");

    #se a data passada como argumento não for igual à mesma formatada corretamente significa que o formato é inválido
    if [ "$date1" != "$date2" ]; then
        echo "Data inválida! Formato correto 'Month Day Hour:Minute'";
        exit 1;
    fi
}
```

Figura 4 - Função validate_date()

Criámos também esta função (Fig.4) que será utilizada mais tarde, no tratamento das opções, para validar o **formato das datas** possivelmente passadas como argumento, mais especificamente, às opções *-s* e *-e*, pois queríamos que apenas datas com o formato apresentado no guião pudessem ser passadas como argumento.

2.2.3 – Função usage()

```
#função chamada caso o argumento seja inválido
#imprime todas os argumentos possíveis
usage() {
    echo " OPÇÕES VÁLIDAS:"
    echo "      -c : filtra por expressão regular"
    echo "      -s : data mínima para início do processo"
    echo "      -e : data máxima para início do processo"
    echo "      -u : seleção dos processos através do user name"
    echo "      -m -M : gama de pids"
    echo "      -p : número de processos a visualizar"
    echo "      -r : ordenação da tabela pela ordem inversa"
    echo "      -w : ordenação da tabela por valores escritos"
    echo " O último argumento tem de ser um número inteiro! (segundos)"
    exit 1
}
```

Figura 5 - Função usage()

Aqui (Fig.5), criámos a função *usage()* que imprime no terminal todas as opções válidas e a sua finalidade, sendo estas : *-c, -s, -e, -u, -m, -M, -p, -r e -w*.

Esta função será chamada aquando da validação dos argumentos e opções caso algum seja inválido, como iremos aprofundar mais à frente.

- Opção *-c* : Permite filtrar os processos através de uma expressão *regex* passada como argumento à mesma, imprimindo no terminal apenas os processos que obedeçam a essa expressão.

- Opções *-s* e *-e* : Permitem a seleção dos processos através da especificação de um período temporal com data mínima e data máxima para o início do processo.

- Opção *-u* : A utilização desta opção permite a impressão no terminal unicamente dos processos que estejam a ser realizados pelo utilizador passado como argumento à mesma.

- Opções *-m* e *-M* : Permitem a filtração dos processos consoante uma gama mínima e máxima de pids, respetivamente.

- Opção *-p* : Esta opção permite ao utilizador escolher a quantidade de processos a visualizar no terminal.

- Opção *-r* : Esta opção tem como finalidade imprimir no terminal todos os processos, ou os filtrados pelo utilizador através das outras opções, por ordem inversa à ordem default.

- Opção *-w* : Por outro lado, esta opção irá ordenar os processos a ser visualizados consoante os *write values,* por ordem decrescente dos mesmos.

2.2.4 – Função arguments_validation()

```
#função de validação de todos os argumentos
arguments_validation(){

    if [[ $# -lt 1 ]]; then  #se o número de argumentos for menor que 1, significa que não foi passado nenhum argumento
        echo "Número de argumentos inválido! Passe, pelo menos, 1 argumento."
        exit 1

    #verificar se o último argumento é um número inteiro
    elif ! num_Int ${@: -1}; then  #se o último argumento não for um número inteiro
        echo "O último argumento deverá ser um número inteiro!"
        exit 1
    fi
```

Figura 6 - Validações obrigatórias

Nesta função (Fig.6), criada para a validação de todos os argumentos, incluindo os passados como argumento às opções, começámos por fazer duas verificações óbvias e obrigatórias para o funcionamento do programa: primeiro, a validação de que o **número de argumentos** é maior que um, pois é obrigatório que o utilizador passe pelo menos o número de segundos do *sleep time* como argumento e, de seguida, que o último argumento, que representa os segundos, é um **número inteiro.**

De seguida, passamos para o **tratamento das opções**.

```
while getopts ":c:s:e:u:m:M:p:rw" opt; do #verificar se os argumentos passados são válidos
```

Figura 7 - comando getopts

Para tal, utilizámos o comando *getopts* (Fig.7) ao qual podemos passar as várias opções, neste caso, para a validação dos argumentos que podem ser passados às mesmas. Como podemos observar acima, algumas opções são seguidas de ":", cujo uso indica que essa opção precisará obrigatoriamente de um argumento passado a si.

```
case $opt in
    c) #verificar se a flag ja foi acionada anteriormente
        if [ "$flag_c" != "NULL" ]; then
            echo "A flag -c já foi acionada anteriormente!"
            exit 1
        fi
        if [[ $# != $(($OPTIND-1)) ]] && ! [[ $OPTARG =~ ^-[a-zA-Z] ]] && ! [[ $OPTARG =~ $re ]] ; then #verificar que o OPTARG não é o último argumento nem começa com "-"
            flag_c=$OPTARG #se for válido, guardar o valor do argumento na variável flag_c
            n_argumentos=$(($n_argumentos+1)) #incrementar o número de argumentos
            n_opcoes=$(($n_opcoes+1)) #incrementar o número de opções
        else #se não for válido
            echo "O argumento -c nao pode ser o último argumento nem pode ser seguido de outro argumento '-#'!"
            usage
            exit 1
        fi
    ;;
```

Figura 8 - Opção -c

Acima (Fig.8), no caso da opção *-c,* podemos observar inicialmente uma validação que será feita em **todas** as opções que verifica que a mesma opção não é passada mais que uma vez. Para além desta, podemos observar outra validação que aparece na **maioria** das opções que consiste em verificar que o argumento passado à opção nem é o último argumento nem começa com "-", ou seja, não é outra opção. Dentro da mesma, iremos incrementar o número de argumentos e de opções. Se ambas as validações tiverem sucesso, a variável associada a essa opção passará a ter o valor do argumento passado à mesma.

Caso contrário, será imprimida no terminal uma mensagem alertando o utilizador do funcionamento **inválido** da opção e será chamada a função *usage*(), explicada acima. Este raciocínio, como dito anteriormente, será utilizado no tratamento de todas as opções que necessitam obrigatoriamente de um argumento.

```
s)
    #converter a data mínima para segundos para poder comparar com a data máxima, caso esta seja passada como argumento
    if [ "$min_date" != "NULL" ]; then
        echo "A flag -s já foi acionada anteriormente!"
        exit 1
    fi
    min_date_s=$(date -d "$OPTARG" +%s)
    if [[ $# != $(($OPTIND-1)) ]] && ! [[ $OPTARG =~ ^-[a-zA-Z] ]]; then #verificar que o OPTARG não é o último argumento nem começa com "-"
        if validate_date $OPTARG; then #verificar se a data é válida através da função date
            min_date=$OPTARG #se for válida, guardar o valor do argumento na variável min_date
            n_argumentos=$(($n_argumentos+1)) #incrementar o número de argumentos
            n_opcoes=$(($n_opcoes+1)) #incrementar o número de opções
        fi
    else
        echo "O argumento -s nao pode ser o último argumento nem pode ser seguido de outro argumento '-#'!"
        usage #funcão criada para mostrar as opcões de argumentos válidos
        exit 1
    fi

;;

e)
    #converter a data máxima para segundos para poder comparar com a data mínima
    if [ "$max_date" != "NULL" ]; then
        echo "A flag -e já foi acionada anteriormente!"
        exit 1
    fi
    max_date_s=$(date -d "$OPTARG" +%s)
    if [[ $# != $(($OPTIND-1)) ]] && ! [[ $OPTARG =~ ^-[a-zA-Z] ]]; then
        if [[ $min_date_s -ge $max_date_s ]]; then #validar que ,caso a data máxima exista, esta é maior que a data mínima
            echo "Data máxima inválida! A data máxima deve ser maior que a data mínima, ou não existir!"
            usage #funcão criada para mostrar as opcões de argumentos possíveis
            exit 1
        else
            if validate_date $OPTARG; then #verificar se a data é válida através da função date
                max_date=$OPTARG #se for válida, guardar o valor do argumento na variável max_date
                n_argumentos=$(($n_argumentos+1)) #incrementar o número de argumentos
                n_opcoes=$(($n_opcoes+1)) #incrementar o número de opções
            fi
        fi
    else
        echo "O argumento -e nao pode ser o último argumento nem pode ser seguido de outro argumento '-#'!"
        usage
        exit 1
    fi
;;
```

Figura 9 – Opções -s e -e

No caso das opções *-s* e *-e* (Fig.9*),* quisemos verificar que a data mínima é **menor** que a data máxima. Para isso, atribuímos às variáveis *min_date_s* e *max_date_s* o valor em **segundos** das datas mínima e máxima, respetivamente, através do comando date e os seus formatos e, depois, comparámos ambas no tratamento da opção *-e*. Para além disto, quisemos verificar que as datas passadas como argumento estavam no **formato** dado pelo professor no guião chamando, para isto, a função *validate_date()* explicada acima.

```
u)
    if [ "$flag_u" != "NULL" ]; then
        echo "A flag -u já foi acionada anteriormente!"
        exit 1
    fi
    if [[ $# != $(($OPTIND-1)) ]] && ! [[ $OPTARG =~ ^-[a-zA-Z] ]]; then
            #verificar se o argumento passadon é um username válido
            if id -u $OPTARG >/dev/null 2>&1; then
                flag_u=$OPTARG #se for válido, guardar o valor do argumento na variável flag_u
                n_argumentos=$(($n_argumentos+1)) #incrementar o número de argumentos
                n_opcoes=$(($n_opcoes+1)) #incrementar o número de opções
            else
                echo "O argumento -u não é um username válido!"
                usage
                exit 1
            fi
        else
        echo "O argumento -u nao pode ser o último argumento nem pode ser seguido de outro argumento '-#'!"
        usage
        exit 1
    fi
;;
```

Figura 10 - Opção -u

No caso da opção *-u* (Fig.10), utilizámos duas condições *if* que encontrámos após alguma pesquisa e adaptámos ao nosso programa para verificar se o **user** passado como argumento existe.

```
m)
    if [ "$flag_m" != "NULL" ]; then
        echo "A flag -m já foi acionada anteriormente!"
        exit 1
    fi
    if [[ $# != $(($OPTIND-1)) ]] && ! [[ $OPTARG =~ ^-[a-zA-Z] ]]; then
        if num_Int $OPTARG; then #verificar se o argumento é um número inteiro
            flag_m=$OPTARG #se for válido, guardar o valor do argumento na variável flag_m
            n_argumentos=$(($n_argumentos+1)) #incrementar o número de argumentos
            n_opcoes=$(($n_opcoes+1)) #incrementar o número de opções
        else
            echo "O argumento -m deve ser seguido de um número inteiro!"
            usage #função criada para mostrar as opções de argumentos possíveis
            exit 1
        fi
    else
        echo "O argumento -m nao pode ser o último argumento nem pode ser seguido de outro argumento '-#'!"
        usage
        exit 1
    fi
;;

M)
    if [ "$flag_M" != "NULL" ]; then
        echo "A flag -M já foi acionada anteriormente!"
        exit 1
    fi
    if [[ $# != $(($OPTIND-1)) ]] && ! [[ $OPTARG =~ ^-[a-zA-Z] ]]; then
        if num_Int $OPTARG; then #verificar se o argumento é um número inteiro
            if [[ $OPTARG -gt $flag_m ]]; then #verificar que o valor de gama de pids M é maior que o valor de m
                flag_M=$OPTARG #se for válido, guardar o valor do argumento na variável flag_M
                n_argumentos=$(($n_argumentos+1)) #incrementar o número de argumentos
                n_opcoes=$(($n_opcoes+1)) #incrementar o número de opções
            else
                echo "O argumento -M deve ser maior que o argumento -m!"
                usage
                exit 1
            fi
        else
            echo "O argumento -M deve ser seguido de um número inteiro!"
            usage
            exit 1
        fi
    else
        echo "O argumento -M nao pode ser o último argumento nem pode ser seguido de outro argumento '-#'!"
        usage
        exit 1
    fi
```

Figura 11 - Opção -m e -M

No caso das opções *-m* e *-M* (Fig.11), como se referem à **gama de pids**, chamámos a função *num_Int()* para validarmos que os argumentos passados são números inteiros e, de seguida, na opção *-M* verificamos através de uma pequena condição if que o número passado como argumento a esta opção deverá ser **maior** que o número passado à opção *-m*.

```
p)
    if [ "$flag_p" != "NULL" ]; then
        echo "A flag -p já foi acionada anteriormente!"
        exit 1
    fi
    if [[ $# != $(($OPTIND-1)) ]] && ! [[ $OPTARG =~ ^-[a-zA-Z] ]]; then #verificar que o OPTARG não é o último argumento nem começa com "-"
        if num_Int $OPTARG; then #verificar se o argumento é um número inteiro
            flag_p=$OPTARG
            n_argumentos=$(($n_argumentos+1)) #incrementar o número de argumentos
            n_opcoes=$(($n_opcoes+1)) #incrementar o número de opções
        else
            echo "O argumento -p deve ser seguido de um número inteiro!"
            usage
            exit 1
        fi
    else
        echo "O argumento -p nao pode ser o último argumento nem pode ser seguido de outro argumento '-#'!"
        usage
        exit 1
    fi
;;
```

Figura 12 - Opção -p

Na opção *-p* (Fig.12), que determina o **número de processos a visualizar**, para além das verificações comuns a todas as opções que necessitam obrigatoriamente de argumento, apenas precisámos de verificar que o argumento passado é um número inteiro, novamente através da função *num_int().*

```
r)
    if [ "$reverse" != "" ]; then
        echo "A flag -r já foi acionada anteriormente!"
        exit 1
    fi
    reverse="-r" #se o argumento -r for passado, guardar o valor "-r" na variável reverse e o programa vai ordenar os resultados por ordem normal da taxa de leitura
    n_opcoes=$(($n_opcoes+1)) #incrementar o número de opções
;;

w)
    if [ "$flag_w" != "NULL" ]; then
        echo "A flag -w já foi acionada anteriormente!"
        exit 1
    fi
    flag_w="-w" #se a fkag -w for passada, guardar o valor "-w" na variável flag_w e o programa vai ser ordenado tendo por base a coluna dos WriteB
    n_opcoes=$(($n_opcoes+1)) #incrementar o número de opções
;;
```

Figura 13 - Opção -r e -w

A utilização das opções -r e -w (Fig.13) será bem explicada mais abaixo, aquando da explicação da função *sort_process()*.

```
*) #caso o argumento não seja válido
    echo "Opção inválida: -$OPTARG" >&2
    usage
    exit 1
;;
```

*Figura 14 - Opção **

Quanto à última opção *, esta representa qualquer opção passada como argumento que não seja uma das tratadas acima, ou seja, qualquer opção **inválida**.

2.2.5 - Função listar_processos()

```
listar_processos(){
    printf "%-9s\t\t %8s\t\t %10s\t %10s\t %9s\t %10s\t %10s %16s\n" "COMM" "USER" "PID" "READB" "WRITEB" "RATER" "RATEW" "DATE"
```

Figura 15 - Cabeçalho

Inicialmente e antes de mesmo de ir recolher os dados necessários de cada processo o **cabeçalho será escrito no terminal** por questão de organização de código como é observado na figura acima (Fig.15).

```
for pid in $(ps -e -o pid=); do #percorrer todos os pids existentes no diretório proc
    #verificar se os ficheiros io, status e comm são readables
    if [[ -r /proc/$pid/io && -r /proc/$pid/status && -r /proc/$pid/comm ]]; then
        #guardar o valor de rchar do ficheiro io do pid em questão na posição do pid no array READBI
        READBI[$pid]=`cat $pid/io | grep rchar | awk '{print $2}'`
        #guardar o valor de wchar do ficheiro io do pid em questão na posição do pid no array WRITEBI
        WRITEBI[$pid]=`cat $pid/io | grep wchar | awk '{print $2}'`
    fi
done
```

Figura 16 - Loop for para recolha de dados

De seguida, através de um **"loop for"** (Fig.16) o sistema percorrerá todos os processos contidos dentro do diretório ***/proc*** validando sempre se cada um dos processos contém os diretórios ***/io***, ***/status***, ***/comm*** possíveis com a permissão de leitura ativada. Caso contrário, o processo não será percorrido e consequentemente não entrará mais tarde para a lista de processos em estudo.

Após a validação, o programa vai proceder com a leitura inicial e escrita inicial dos bytes de cada processo através dos comandos *cat*, *grep* e *awk*. Ao aceder ao ficheiro ***io*** do processo, o programa procura os respetivos valores e guarda-los num dos *arrays* associativos apresentados no início do relatório sendo a sua **key** os **pids** *(id's dos processos)* respetivos.

```
    sleep $last #esperar o tempo que o utilizador passou como argumento para o programa
```

Figura 17 - sleep

Após este procedimento, o programa é colocado em *sleep* (Fig.17) durante **$last** segundos (variável definida anteriormente*)* definidos pelo utilizador como **último argumento.** Isto porque à posteriori será necessário calcular de novo os valores de leitura e escrita as suas respetivas taxas.

```
    for pid in $(ps -e -o pid=); do #percorrer todos os pids existentes no diretório proc outra vez
        if [[ -r /proc/$pid/io && -r /proc/$pid/status && -r /proc/$pid/comm ]]; then #verificar se os ficheiros io, status e comm são readables
            if [[ ! ${!READBI[*]} =~ "${pid}" ]]; then
                continue
            fi
```

Figura 18 - Loop for a seguir ao sleep

Como referido anteriormente, percorremos de novo todos os processos fazendo **a mesma verificação** que se fez anteriormente, com a única diferença de que adicionámos uma nova **condição if para o programa não ler os processos que sejam criados a meio do tempo de sleep** (Fig.18).

Estes não nos interessam, pois não cumprem os requisitos pedidos no problema que dependem completamente do tempo da variação do tempo entre uma leitura e a outra.

```
READBF=`cat $pid/io | grep rchar | awk '{print $2}'`  #guardar o valor de rchar do ficheiro io do pid em questão numa variável
WRITEBF=`cat $pid/io | grep wchar | awk '{print $2}'` #guardar o valor de wchar do ficheiro io do pid em questão numa variável
#calcular a diferenca entre o READBF e o READBI e guardar o valor na posição do pid no array READB
READB[$pid]=$(($READBF-${READBI[$pid]}))
#calcular a diferenca entre o WRITEBF e o WRITEBI e guardar o valor na posição do pid no array WRITEB
WRITEB[$pid]=$(($WRITEBF-${WRITEBI[$pid]}))

COMM=$(cat $pid/comm | tr -s ' ' '_') #guardar o valor de comm do ficheiro comm do processo em questão numa variável

USER=$(ps -u -p $pid | awk '{print $1}' | tail -1) #guarda o valor do user do processo em questão numa variável
#Data de criacao do processo
process_date=$(ls -ld /proc/$pid)
process_date=$(echo $process_date | awk '{ print $6" "$7" "$8}') #guardar a data de criação do processo no formato desejado numa variável
#calcular a taxa de leitura usando a fórmula dada no enunciado (READBF - READBI) / last
RATER=$(echo "scale=2; ($READBF - ${READBI[$pid]}) / $last" | bc)
#calcular a taxa de escrita usando a fórmula dada no enunciado (WRITEBF - WRITEBI) / last
RATEW=$(echo "scale=2; ($WRITEBF - ${WRITEBI[$pid]}) / $last" | bc)
```

Figura 19 - Continuação do loop for e recolha de informação

Assim, o programa vai recolher todos os dados que precisa sendo estes de novo os bytes escritos e lidos (**$READBF** e **$WRITEBF**), o nome do processo (**$COMM**), o nome de utilizador do mesmo (**$USER**), a sua data de criação (**$process_date**) e a sua taxa de leitura e escrita respetivamente (**$RATER** e **$RATEW**), como podemos ver na figura acima (Fig.19).

No $COMM, adicionámos um pequeno detalhe que, caso os nomes contenham espaços, estes serão transformados num underscore através do comando *tr*, facilitando a sua organização.

Entretanto, podemos observar alguns cálculos matemáticos na figura acima que vamos passar a explicar com figuras mais específicas.

Para calcular o **número de bytes escritos e lidos** durante o sleep time fizemos a diferença entre os respetivos anteriores ao sleep time e a seguir ao mesmo, como podemos observar na Fig.20. Estes valores serão guardados para, mais a frente, serem escritos no terminal.

```
READB[$pid]=$(($READBF-${READBI[$pid]}))
#calcular a diferenca entre o WRITEBF e o WR
WRITEB[$pid]=$(($WRITEBF-${WRITEBI[$pid]}))
```

Figura 20 - Número de bytes escritos e lidos durante o sleep

```
RATER=$(echo "scale=2; ($READBF - ${READBI[$pid]}) / $last" | bc)
#calcular a taxa de escrita usando a fórmula dada no enunciado (WRI
RATEW=$(echo "scale=2; ($WRITEBF - ${WRITEBI[$pid]}) / $last" | bc)
```

Figura 21 - Taxas de leitura e escrita durante o sleep

O mesmo se verifica para as respetivas **taxas** sendo estas o número de bytes lidos e escritos por segundo durante o espaço de tempo em que o **$last** atua (Fig.21). Calculam-se usando a seguinte expressão**,** aplicada ao caso em estudo**:**

$$Rate = \frac{(leitura2 - leitura1)}{intervalo\ de\ tempo}$$

```
#Guarda o processo em questão num array associativo com o pid como chave
#Cada valor é guardado numa coluna separada por tabs e formatada para que fique igual ao que foi pedido no enunciado
PID_array[$pid]=$(printf "\n%-20s\t\t %8s\t\t %10s\t %10s\t %9s\t %10s\t %10s %16s\n" "$COMM" "$USER" "$pid" "${READB[$pid]}" "${WRITEB[$pid]}" "$RATER" "$RATEW" "$process_date")
```

Figura 22 - PID_array

Por fim todos estes valores são armazenados no **PID_array[$pid]** já formatados consoante o cabeçalho para facilitar a sua ordenação e organização mais tarde. Recordando que, como estamos a usar **arrays associativos** cada **$pid** terá um conjunto de valores associado.

Mais tarde iremos filtrar estes respetivos **$pids** através da função **PID_filter*()*** tendo também por base os argumentos validados na função *arguments_validation().*

2.2.6 - Função PID_filter()

Esta função tem como objetivo filtrar os processos definidos na função *listar_processos()* **retirando** do *PID_array[$pid]* os $pids dos processos que **não** cumpram os valores definidos nas flags validadas na função *arguments_validation().*

Inicialmente são percorridos **todos os $pids** existentes no *PID_array[$pid].* O bash-shell tem a capacidade de distinguir colunas de um array associativo logo, como as variáveis foram armazenadas em colunas, usando o comando *awk* conseguimos aceder a essas respetivas colunas alcançando os valores pretendidos em cada uma.

```
#Acede ao valor COMM associado ao pid em questão e guarda-o numa variável
COMM=$(echo ${PID_array[$pid]} | awk '{print $1}')
#Acede ao valor USER associado ao pid em questão e guarda-o numa variável
USER=$(echo ${PID_array[$pid]} | awk '{print $2}')
#Acede ao valor DATE associado ao pid em questão e guarda-o numa variável
DATE=$(echo ${PID_array[$pid]} | awk '{print $8" "$9" "$10}')
```

Figura 23 - Exemplos da utilização do comando awk

Na Fig.23 podemos observar alguns exemplos do funcionamento do comando ***awk***. Através do mesmo conseguimos aceder ao valor do $COMM e do $USER acedendo à primeira e à segunda colunas respetivamente. No caso do $DATE a única pequena diferença perante os restantes elementos do array é que ocupa três colunas no array, pois está formatada da seguinte forma “Month Day Hour:Minute”, sendo as colunas separadas por um whitespace.

De seguida, vamos filtrar os valores recolhidos pelas variáveis diante dos argumentos recolhidos pelo programa. Cada flag tem uma filtragem individual, como poderemos ver de seguida.

```
#Compara o valor de COMM com o valor de COMM passado como argumento
if [[ $flag_c != "NULL" ]]; then #se o utilizador passou o argumento -c
    if ! [[ $COMM =~ $flag_c ]]; then #se o valor de COMM não contém o valor de COMM passado como argumento
        unset PID_array[$pid] #remove o pid do array PID_array
    fi
fi
```

Figura 24 - Filtragem pelo $comm

Primeiramente, e este será um processo recorrente em todos os elementos de filtragem presentes nesta função, vamos verificar se a flag em questão foi **passada como argumento** pelo utilizador, como podemos ver na figura acima. Recordando que todas as flags foram inicializadas com o valor default 'NULL', portanto basta verificar se valor da flag é diferente de 'NULL', pois caso esta flag tenha outro valor significa que o mesmo foi atualizado na função *arguments_validation()*, ou seja, significa que o utilizador a passou como argumento.

Caso o mesmo se verifique, no caso da *flag_c* (Fig.24) iremos avançar para outra condição que verifica se $COMM não contém o valor de COMM passado como argumento. Todos os $pids que **cumpram** esta condição serão **removidos** do *PID_array[$pid]* seguindo o comando *unset*.

```
#Compara o valor de USER com o valor de USER passado como argumento
if [[ $flag_u != "NULL" ]]; then #se o utilizador passou o argumento -u
    if ! [[ $USER =~ $flag_u ]]; then  #se o valor de USER não contém o valor de USER passado como argumento
        unset PID_array[$pid] #remove o pid do array PID_array
    fi
fi
```

Figura 25 - Filtragem pelo $user

Seguindo a mesma lógica da flag anterior, verificamos se a *flag_u* foi utilizada. Caso o mesmo se verifique, o programa prosseguirá a verificar se o $USER não contém o valor de USER passado como argumento. Consequente deste acontecimento o respetivo $pid será removido do *PID_array[$pid]* (Fig.25).

```
if [[ $flag_m != "NULL" || $flag_M != "NULL" ]]; then #se o utilizador passou o argumento -m ou -M
    if [[ $flag_m != "NULL" && $flag_M != "NULL" ]]; then #se o utilizador passou os argumentos -m e -M
        if [[ $pid -lt $flag_m || $pid -gt $flag_M ]]; then #verificar se o PID está fora dos valores de -m e -M
            unset PID_array[$pid] #remove o pid do array PID_array
        fi
    elif [[ $flag_m != "NULL" && $flag_M == "NULL" ]]; then #se o utilizador passou o argumento -m
        if [[ $pid -lt $flag_m ]]; then #verificar se o PID está fora do valor de -m
            unset PID_array[$pid] #remove o pid do array PID_array
        fi
    elif [[ $flag_m == "NULL" && $flag_M != "NULL" ]]; then #se o utilizador passou o argumento -M
        if [[ $pid -gt $flag_M ]]; then #verificar se o PID está fora do valor de -M
            unset PID_array[$pid] #remove o pid do array PID_array
        fi
    fi
fi
```

Figura 26 - Filtragem pela gama de pids

Por outro lado, a filtragem pela gama de pids terá que ter um conjunto de verificações um pouco mais elevado, sendo que haverá três diferentes cenários caso alguma ou ambas as flags *(flag_m* e/ou *flag_M*) seja utilizada. Caso tanto a *flag_m* quanto a *flag_M* sejam acionadas ficaremos com um conjunto completamente definido pelo utilizador de gama de pids. Seguindo a lógica do código, todos os $pids que não se encontrem nesse intervalo serão eliminados do *PID_array[$pid].* Caso apenas uma das flags seja acionada, apenas será necessário verificar que a gama de pids a imprimir é menor ou maior que a definida pelo utilizador. Se tudo tiver de acordo com as condições impostas, os pids que estiverem fora dos intervalos estabelecidos serão eliminados do *PID_array[$pid].*

```
if [[ $min_date != "NULL" ]]; then #se o utilizador passou o argumento -d
    if [[ $DATE_s -lt $min_date_s ]]; then #verificar se a data do processo está fora do valor de -d
        unset PID_array[$pid] #remove o pid do array PID_array
    fi
fi
if [[ $max_date != "NULL" ]]; then #se o utilizador passou o argumento -D
    if [[ $DATE_s -gt $max_date_s ]]; then #verificar se a data do processo está fora do valor de -D
        unset PID_array[$pid] #remove o pid do array PID_array
    fi
fi
```

Figura 27 - Filtragem pelo período temporal

Seguindo o mesmo raciocínio de todas as filtragens anteriores, é verificada a ativação das flags *min_date* e *max_date (*Fig.27*)*.

```
#Data do processo mas em segundos
DATE_s=$(date -d "$DATE" +%s)
```

Figura 28 - data em segundos

Em adição e para facilitar o processo de filtragem, a variável *DATE* foi passada para segundos como é visível na imagem ao lado(Fig.28).

Após reunidas as condições necessárias, verifica-se se a data do processo, já em segundos (*$DATE_s*), é inferior à data mínima estabelecida pelo utilizador *($min_date*). De seguida, dentro de outra condição é verificado se a mesma é superior à data máxima estabelecida *($max_date*). Caso as mesma se verifiquem para certo $pid, o mesmo será também removido do *PID_array[$pid].*

No final da função, o **PID_array[$pid]** estará completamente **filtrado** e pronto para ser escrito no terminal pela ordem definida pela função *sort_process(),* explicada de seguida.

2.2.7 - Função sort_process()

Todos as listas de processos terão uma certa **ordem** que será garantida pela função *sort_process().* Esta tem por base o **PID_array[$pid]** já antes definido e filtrado consoante as opções passadas.

Na função arguments_validation(), foram definidas três opções já explicadas anteriormente que iram auxiliar e definir o processo de ordenação, são estas as opções *-p, -r* e *-w.*

```
sort_process(){
    if [[ $flag_p != "NULL" ]]; then #se o utilizador passou o argumento -p
        if [[ $reverse == "-r" ]]; then #se o utilizador passou o argumento -r
            if [[ $flag_w == "-w" ]]; then #se o utilizador passou o argumento -w
                printf "%s" "${PID_array[@]}" | sort -k7 -n | head -n $(($flag_p+1)) #ordena o array PID_array pelo valor de WRITEB e imprime os primeiros $flag_p processos
            else
                printf "%s" "${PID_array[@]}" | sort -k6 -n | head -n $(($flag_p+1)) #ordena o array PID_array pelo valor de RATER e imprime os primeiros $flag_p processos
            fi
        else
            if [[ $flag_w == "-w" ]]; then #se o utilizador passou o argumento -w
                printf "%s" "${PID_array[@]}" | sort -k7 -n -r | head -n $flag_p #ordena o array PID_array pelo valor de WRITEB e imprime os primeiros $flag_p processos
            else
                printf "%s" "${PID_array[@]}" | sort -k6 -n -r | head -n $flag_p #ordena o array PID_array pelo valor de RATER e imprime os primeiros $flag_p processos
            fi
        fi
    else #se o utilizador não passou o argumento -p
        if [[ $reverse == "-r" ]]; then #se o utilizador passou o argumento -r
            if [[ $flag_w == "-w" ]]; then #se o utilizador passou o argumento -w
                printf "%s" "${PID_array[@]}" | sort -k7 -n #ordena o array PID_array pelo valor de WRITEB
            else
                printf "%s" "${PID_array[@]}" | sort -k6 -n #ordena o array PID_array pelo valor de RATER
            fi
        else
            if [[ $flag_w == "-w" ]]; then #se o utilizador passou o argumento -w
                printf "%s" "${PID_array[@]}" | sort -k7 -n -r #ordena o array PID_array pelo valor de WRITEB
            else
                printf "%s" "${PID_array[@]}" | sort -k6 -n -r #ordena o array PID_array pelo valor de RATER
            fi
        fi
    fi
```

Figura 29 - Função sort_process()

Sort_process() será, portanto, um conjunto vasto de condições para abranger **todos os casos possíveis de ordenação**. Teremos, inicialmente, duas grandes condições possíveis que verificam a existência ou não de algum valor atribuído à *flag_p*. Seja qual for o caso, o sistema vai verificar se *–r* e/ou *–w* foram passados como argumento.

Caso nem *-r* nem *-w* sejam passadas como argumento, o programa vai ordenar os processos pela **ordem inversa da taxa de leitura**, ou seja, no nosso caso, por ordem descendente dos valores de **$RATER,** sejam $flag_p processos caso *-p* seja utilizada, ou todos os processos, caso tal não se verifique.

Por outro lado, a opção *-w* irá definir a ordenação de acordo com os *write values,* ou seja, no nosso caso, irá ordenar de forma decrescente com base nos valores de **$WRITEB.**

Quanto à opção *-r,* esta vai ordenar os processos pela ordem **contrária** à ordem default, caso passada sozinha, ou pela ordem contrária da associada à opção *-w*. Assim, irá ordenar os valores de **$RATER** e/ou **$WRITEB** mas, desta vez, de forma crescente.

```
sort -k7 -n -r
```

Figura 30 - Exemplo da utilização do comando sort

Todas estas ordenações são conseguidas através de um pequeno comando ***sort***, como podemos ver exemplificado na Fig.30, onde são especificada(s) a(s) coluna(s) a ordenar através da opção *-k*, seguida da opção *-n* que indica a ordenação por número inteiro, por sua vez seguida ou não da opção *-r* que ordena inversamente.

2.2.8 – Chamar todas as funções

```
arguments_validation "$@" #chama a função arguments_validation passando todos os argumentos passados pelo utilizador
listar_processos #chama a função listar_processos
PID_filter  #chama a função PID_filter
sort_process  #chama a função sort_process
```

Figura 31 - Chamar todas as funções

Para que todas as funções previamente definidas possam funcionar, o programa tem de as chamar. Em cima (Fig. 31) vemos isso mesmo, o programa chama pela ordem correta todas as funções permitindo assim o bom funcionamento de cada. Pequeno detalhe: dentro de algumas funções nomeadamente a função *arguments_validation()*, serão chamadas outras duas funções. Em baixo (Fig.32), conseguimos observar melhor este funcionamento através do fluxo de execução do programa.

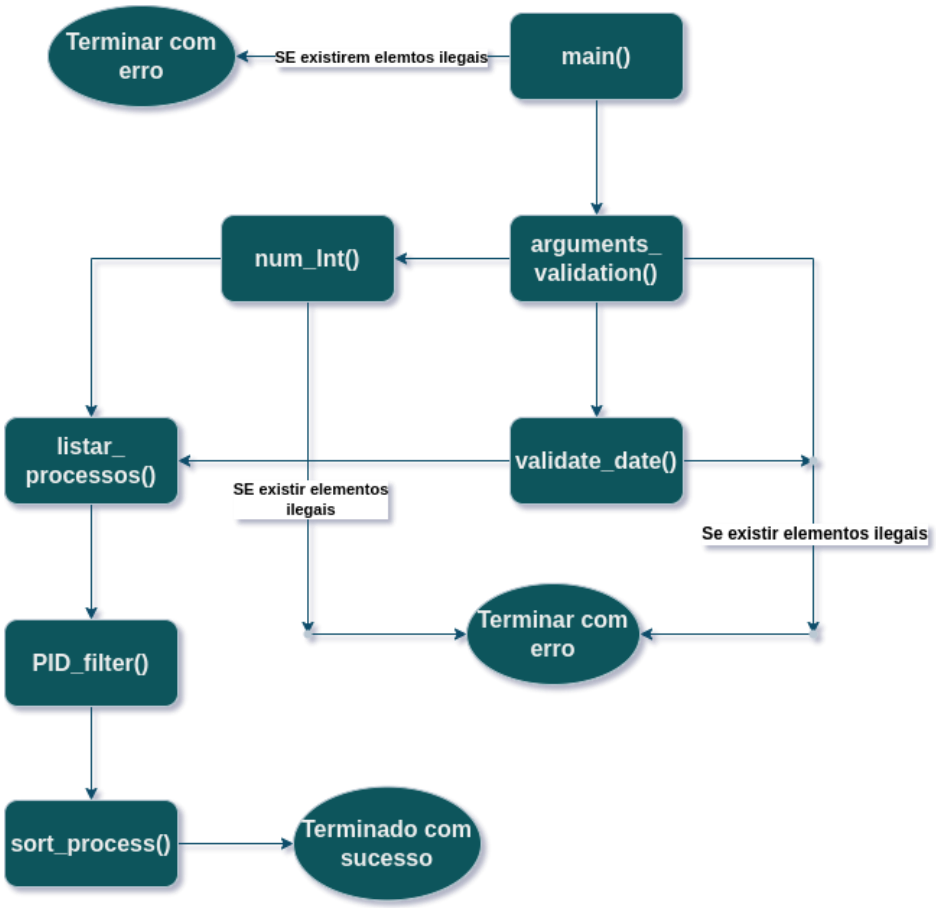


Figura 32 -Fluxo de execução do programa

3 - Demonstrações e testes

Para terminar, realizámos alguns testes para verificar a eficiência e bom funcionamento do script.

Iremos passar à explicação e demonstração dos testes dados como exemplo no guião do projeto, acompanhados, no final, de alguns outros testes extras para demonstração de algumas mensagens de erros e funcionalidades que considerámos importantes.

```
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh
Número de argumentos inválido! Passe, pelo menos, 1 argumento sendo esse o sleep time.
```

Figura 34 - ./rwstat.sh

```
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh a
O último argumento deverá ser um número inteiro correspondente ao sleep time!
```

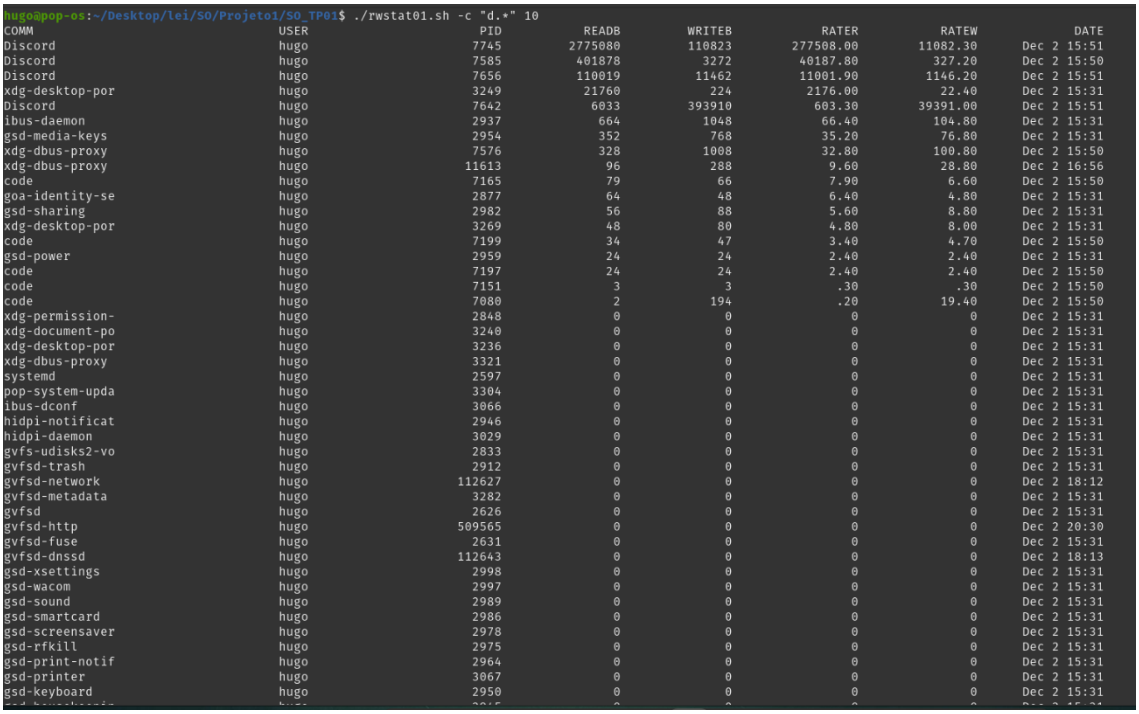
Figura 33 - ./rwstat.sh a

Começámos por correr o script, primeiro, sem qualquer parâmetro (Fig.33) e, de seguida, com um argumento inválido (Fig.34) para nos certificarmos que as mensagens de erro devidas são impressas no terminal.

```
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh 1
COMM                 USER                  PID         READB         WRITEB         RATER         RATEW          DATE
bash                 hugo                25268     134770815         144538     134770815.00     144538.00     Dec 2 17:45
Discord              hugo                 7585         98824             24         98824.00        24.00      Dec 2 15:50
Discord              hugo                 7745         75108          27786         75108.00     27786.00      Dec 2 15:51
spotify              hugo                11620         10649           5104         10649.00      5104.00      Dec 2 16:56
chrome               hugo                14261         10460              0         10460.00            0      Dec 2 17:27
pipewire-pulse       hugo                 2614          9024           2624          9024.00      2624.00      Dec 2 15:31
chrome               hugo                14020          5910            528          5910.00       528.00      Dec 2 17:27
pipewire             hugo                 2607          4960           1512          4960.00      1512.00      Dec 2 15:31
Discord              hugo                 7642           463            463           463.00       463.00      Dec 2 15:51
Discord              hugo                 7656            90              0            90.00            0      Dec 2 15:51
gnome-shell          hugo                 2808            80            632           80.00       632.00      Dec 2 15:31
chrome               hugo                 3555            34              1           34.00         1.00      Dec 2 15:31
chrome               hugo                14022            34              1           34.00         1.00      Dec 2 17:27
code                 hugo                 7165            29             29           29.00        29.00      Dec 2 15:50
gsd-sharing          hugo                 2982            24             40           24.00        40.00      Dec 2 15:31
spotify              hugo                11693            15             15           15.00        15.00      Dec 2 16:56
code                 hugo                 7199            13             13           13.00        13.00      Dec 2 15:50
gsd-color            hugo                 2940             8             16            8.00        16.00      Dec 2 15:31
code                 hugo                 7197             8              8            8.00         8.00      Dec 2 15:50
chrome               hugo                 3550             5              0            5.00            0      Dec 2 15:31
chrome               hugo                13974             4              4            4.00         4.00      Dec 2 17:27
chrome               hugo                 3325             3           8723            3.00      8723.00      Dec 2 15:31
spotify              hugo                11729             1              1            1.00         1.00      Dec 2 16:57
xdg-permission-      hugo                 2848             0              0               0            0      Dec 2 15:31
xdg-document-po      hugo                 3240             0              0               0            0      Dec 2 15:31
xdg-desktop-por      hugo                 3269             0              0               0            0      Dec 2 15:31
xdg-desktop-por      hugo                 3249             0              0               0            0      Dec 2 15:31
xdg-desktop-por      hugo                 3236             0              0               0            0      Dec 2 15:31
xdg-dbus-proxy       hugo                 7576             0              0               0            0      Dec 2 15:50
xdg-dbus-proxy       hugo                 3321             0              0               0            0      Dec 2 15:31
xdg-dbus-proxy       hugo                11613             0              0               0            0      Dec 2 16:56
wireplumber          hugo                 2612             0              0               0            0      Dec 2 15:31
tracker-miner-f      hugo                 2687             0              0               0            0      Dec 2 15:31
touchegg             hugo                 3017             0              0               0            0      Dec 2 15:31
systemd              hugo                 2597             0              0               0            0      Dec 2 15:31
spotify              hugo                11688             0              0               0            0      Dec 2 16:57
spotify              hugo                11640             0              0               0            0      Dec 2 16:56
spotify              hugo                11627             0              0               0            0      Dec 2 16:56
spotify              hugo                11626             0              0               0            0      Dec 2 16:56
socat                hugo                 7582             0              0               0            0      Dec 2 15:50
sh                   hugo                 2934             0              0               0            0      Dec 2 15:31
pop-system-upda      hugo                 3304             0              0               0            0      Dec 2 15:31
```

Figura 35 - ./rwstat.sh 10

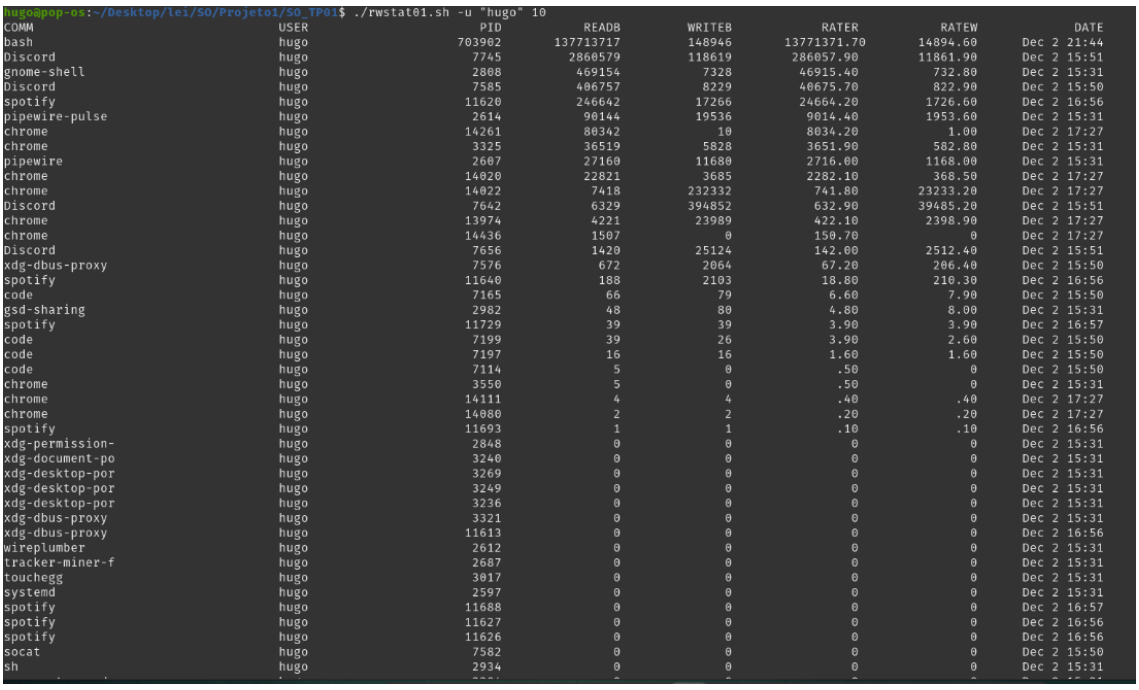
Na figura acima (Fig.35) podemos observar que apenas passando um argumento válido para o sleep, ou seja, um número inteiro é que o script é executado, tal como pretendido.



```
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh -c "d.*" 10
COMM                            USER                     PID          READB          WRITEB          RATER          RATEW             DATE
Discord                         hugo                    7745        2775080         110823         277508.00       11082.30      Dec 2 15:51
Discord                         hugo                    7585         401878           3272          40187.80         327.20      Dec 2 15:50
Discord                         hugo                    7656         110019          11462          11001.90        1146.20      Dec 2 15:51
xdg-desktop-por                 hugo                    3249          21760            224           2176.00          22.40      Dec 2 15:31
Discord                         hugo                    7642           6033         393910            603.30       39391.00      Dec 2 15:51
ibus-daemon                     hugo                    2937            664           1048             66.40         104.80      Dec 2 15:31
gsd-media-keys                  hugo                    2954            352            768             35.20          76.80      Dec 2 15:31
xdg-dbus-proxy                  hugo                    7576            328           1008             32.80         100.80      Dec 2 15:50
xdg-dbus-proxy                  hugo                   11613             96            288              9.60          28.80      Dec 2 16:56
code                            hugo                    7165             79             66              7.90           6.60      Dec 2 15:50
goa-identity-se                 hugo                    2877             64             48              6.40           4.80      Dec 2 15:31
gsd-sharing                     hugo                    2982             56             88              5.60           8.80      Dec 2 15:31
xdg-desktop-por                 hugo                    3269             48             80              4.80           8.00      Dec 2 15:31
code                            hugo                    7199             34             47              3.40           4.70      Dec 2 15:50
gsd-power                       hugo                    2959             24             24              2.40           2.40      Dec 2 15:31
code                            hugo                    7197             24             24              2.40           2.40      Dec 2 15:50
code                            hugo                    7151              3              3               .30            .30      Dec 2 15:50
code                            hugo                    7080              2            194               .20          19.40      Dec 2 15:50
xdg-permission-                 hugo                    2848              0              0                 0              0      Dec 2 15:31
xdg-document-po                 hugo                    3240              0              0                 0              0      Dec 2 15:31
xdg-desktop-por                 hugo                    3236              0              0                 0              0      Dec 2 15:31
xdg-dbus-proxy                  hugo                    3321              0              0                 0              0      Dec 2 15:31
systemd                         hugo                    2597              0              0                 0              0      Dec 2 15:31
pop-system-upda                 hugo                    3304              0              0                 0              0      Dec 2 15:31
ibus-dconf                      hugo                    3066              0              0                 0              0      Dec 2 15:31
hidpi-notificat                 hugo                    2946              0              0                 0              0      Dec 2 15:31
hidpi-daemon                    hugo                    3029              0              0                 0              0      Dec 2 15:31
gvfs-udisks2-vo                 hugo                    2833              0              0                 0              0      Dec 2 15:31
gvfsd-trash                     hugo                    2912              0              0                 0              0      Dec 2 15:31
gvfsd-network                   hugo                  112627              0              0                 0              0      Dec 2 18:12
gvfsd-metadata                  hugo                    3282              0              0                 0              0      Dec 2 15:31
gvfsd                           hugo                    2626              0              0                 0              0      Dec 2 15:31
gvfsd-http                      hugo                  509565              0              0                 0              0      Dec 2 20:30
gvfsd-fuse                      hugo                    2631              0              0                 0              0      Dec 2 15:31
gvfsd-dnssd                     hugo                  112643              0              0                 0              0      Dec 2 18:13
gsd-xsettings                   hugo                    2998              0              0                 0              0      Dec 2 15:31
gsd-wacom                       hugo                    2997              0              0                 0              0      Dec 2 15:31
gsd-sound                       hugo                    2989              0              0                 0              0      Dec 2 15:31
gsd-smartcard                   hugo                    2986              0              0                 0              0      Dec 2 15:31
gsd-screensaver                 hugo                    2978              0              0                 0              0      Dec 2 15:31
gsd-rfkill                      hugo                    2975              0              0                 0              0      Dec 2 15:31
gsd-print-notif                 hugo                    2964              0              0                 0              0      Dec 2 15:31
gsd-printer                     hugo                    3067              0              0                 0              0      Dec 2 15:31
gsd-keyboard                    hugo                    2950              0              0                 0              0      Dec 2 15:31
```

Figura 36 - ./rwstat.sh -c "d." 10*

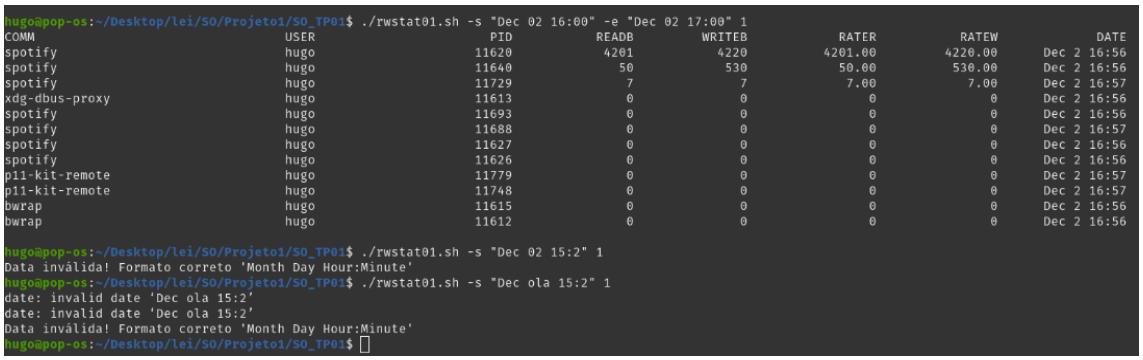
Aqui (Fig.36), testamos a opção *-c* com uma expressão regex que deverá imprimir no terminal apenas os processos com “d” no comm. Como podemos observar este teste é bem sucedido.



```
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh -u "hugo" 10
COMM                            USER                     PID          READB          WRITEB          RATER          RATEW             DATE
bash                            hugo                  703902       137713717         148946       13771371.70      14894.60      Dec 2 21:44
Discord                         hugo                    7745         2860579         118619         286057.90      11861.90      Dec 2 15:51
gnome-shell                     hugo                    2808          469154           7328          46915.40        732.80      Dec 2 15:31
Discord                         hugo                    7585          406757           8229          40675.70        822.90      Dec 2 15:50
spotify                         hugo                   11620          246642          17266          24664.20       1726.60      Dec 2 16:56
pipewire-pulse                  hugo                    2614           90144          19536           9014.40       1953.60      Dec 2 15:31
chrome                          hugo                   14261           80342             10           8034.20          1.00      Dec 2 17:27
chrome                          hugo                    3325           36519           5828           3651.90        582.80      Dec 2 15:31
pipewire                        hugo                    2607           27160          11680           2716.00       1168.00      Dec 2 15:31
chrome                          hugo                   14020           22821           3685           2282.10        368.50      Dec 2 17:27
chrome                          hugo                   14022            7418         232332            741.80      23233.20      Dec 2 17:27
Discord                         hugo                    7642            6329         394852            632.90      39485.20      Dec 2 15:51
chrome                          hugo                   13974            4221          23989            422.10       2398.90      Dec 2 17:27
chrome                          hugo                   14436            1507              0            150.70             0      Dec 2 17:27
Discord                         hugo                    7656            1420          25124            142.00       2512.40      Dec 2 15:51
xdg-dbus-proxy                  hugo                    7576             672           2064             67.20        206.40      Dec 2 15:50
spotify                         hugo                   11640             188           2103             18.80        210.30      Dec 2 16:56
code                            hugo                    7165              66             79              6.60          7.90      Dec 2 15:50
gsd-sharing                     hugo                    2982              48             80              4.80          8.00      Dec 2 15:31
spotify                         hugo                   11729              39             39              3.90          3.90      Dec 2 16:57
code                            hugo                    7199              39             26              3.90          2.60      Dec 2 15:50
code                            hugo                    7197              16             16              1.60          1.60      Dec 2 15:50
code                            hugo                    7114               5              0               .50             0      Dec 2 15:50
chrome                          hugo                    3550               5              0               .50             0      Dec 2 15:31
chrome                          hugo                   14111               4              4               .40           .40      Dec 2 17:27
chrome                          hugo                   14080               2              2               .20           .20      Dec 2 17:27
spotify                         hugo                   11693               1              1               .10           .10      Dec 2 16:56
xdg-permission-                 hugo                    2848               0              0                 0             0      Dec 2 15:31
xdg-document-po                 hugo                    3240               0              0                 0             0      Dec 2 15:31
xdg-desktop-por                 hugo                    3269               0              0                 0             0      Dec 2 15:31
xdg-desktop-por                 hugo                    3249               0              0                 0             0      Dec 2 15:31
xdg-desktop-por                 hugo                    3236               0              0                 0             0      Dec 2 15:31
xdg-dbus-proxy                  hugo                    3321               0              0                 0             0      Dec 2 15:31
xdg-dbus-proxy                  hugo                   11613               0              0                 0             0      Dec 2 16:56
wireplumber                     hugo                    2612               0              0                 0             0      Dec 2 15:31
tracker-miner-f                 hugo                    2687               0              0                 0             0      Dec 2 15:31
touchegg                        hugo                    3017               0              0                 0             0      Dec 2 15:31
systemd                         hugo                    2597               0              0                 0             0      Dec 2 15:31
spotify                         hugo                   11688               0              0                 0             0      Dec 2 16:57
spotify                         hugo                   11627               0              0                 0             0      Dec 2 16:56
spotify                         hugo                   11626               0              0                 0             0      Dec 2 16:56
socat                           hugo                    7582               0              0                 0             0      Dec 2 15:50
sh                              hugo                    2934               0              0                 0             0      Dec 2 15:31
```

Figura 37 - ./rwstat.sh -u "hugo" 10

A opção *-u* fará com que sejam apresentados apenas os processos correspondentes ao user passado como argumento à mesma (Fig.37).

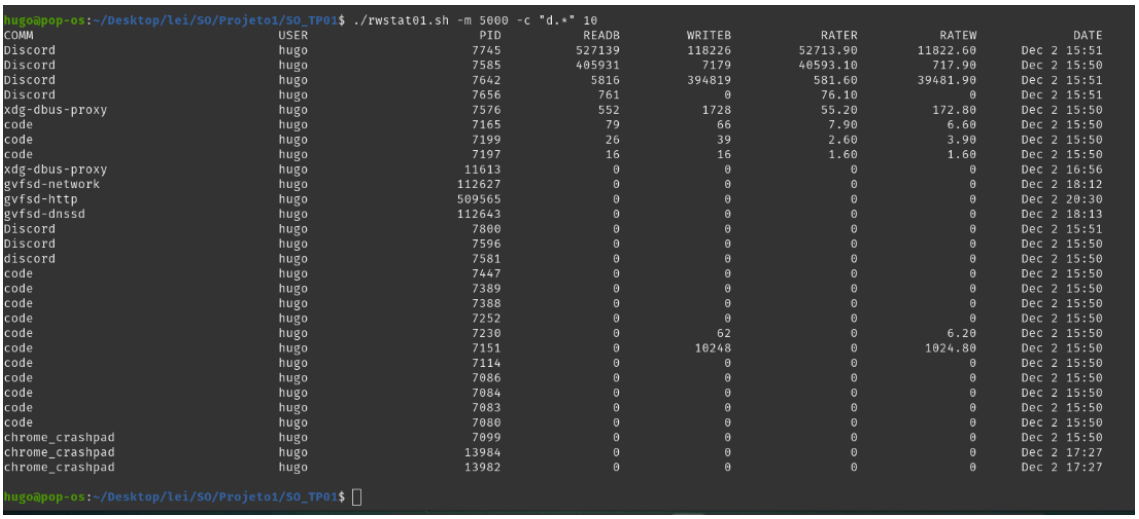


```
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh -s "Dec 02 16:00" -e "Dec 02 17:00" 1
COMM                            USER                     PID            READB           WRITEB            RATER            RATEW             DATE
spotify                         hugo                   11620             4201             4220          4201.00          4220.00      Dec 2 16:56
spotify                         hugo                   11640               50              530            50.00           530.00      Dec 2 16:56
spotify                         hugo                   11729                7                7             7.00             7.00      Dec 2 16:57
xdg-dbus-proxy                  hugo                   11613                0                0                0                0      Dec 2 16:56
spotify                         hugo                   11693                0                0                0                0      Dec 2 16:56
spotify                         hugo                   11688                0                0                0                0      Dec 2 16:57
spotify                         hugo                   11627                0                0                0                0      Dec 2 16:56
spotify                         hugo                   11626                0                0                0                0      Dec 2 16:56
p11-kit-remote                  hugo                   11779                0                0                0                0      Dec 2 16:57
p11-kit-remote                  hugo                   11748                0                0                0                0      Dec 2 16:57
bwrap                           hugo                   11615                0                0                0                0      Dec 2 16:56
bwrap                           hugo                   11612                0                0                0                0      Dec 2 16:56

hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh -s "Dec 02 15:2" 1
Data inválida! Formato correto 'Month Day Hour:Minute'
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh -s "Dec ola 15:2" 1
date: invalid date ‘Dec ola 15:2’
date: invalid date ‘Dec ola 15:2’
Data inválida! Formato correto 'Month Day Hour:Minute'
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$
```

Figura 38 - ./rwstat.sh -s "Dec 02 16:00" -e "Dec 02 17:00" 1

Na figura acima (Fig.38), podemos observar o filtro do período temporal para o início do processo a funcionar. Para além disso, podemos observar algumas mensagens de erro que são impressas no terminal caso o formato da data não seja correto.

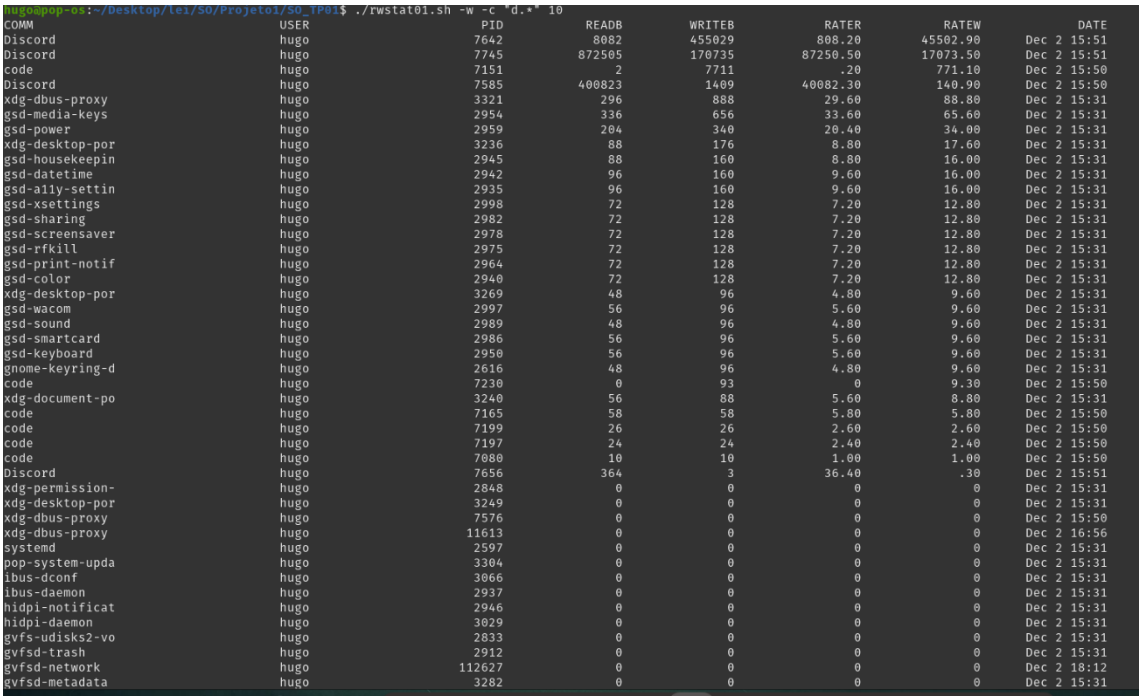


```
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh -m 5000 -c "d.*" 10
COMM                            USER                     PID            READB           WRITEB            RATER            RATEW             DATE
Discord                         hugo                    7745           527139           118226         52713.90         11822.60      Dec 2 15:51
Discord                         hugo                    7585           405931             7179         40593.10           717.90      Dec 2 15:50
Discord                         hugo                    7642             5816           394819           581.60         39481.90      Dec 2 15:51
Discord                         hugo                    7656              761                0            76.10             0.00      Dec 2 15:51
xdg-dbus-proxy                  hugo                    7576              552             1728            55.20           172.80      Dec 2 15:50
code                            hugo                    7165               79               66             7.90             6.60      Dec 2 15:50
code                            hugo                    7199               26               39             2.60             3.90      Dec 2 15:50
code                            hugo                    7197               16               16             1.60             1.60      Dec 2 15:50
xdg-dbus-proxy                  hugo                   11613                0                0                0                0      Dec 2 16:56
gvfsd-network                   hugo                  112627                0                0                0                0      Dec 2 18:12
gvfsd-http                      hugo                  509565                0                0                0                0      Dec 2 20:30
gvfsd-dnssd                     hugo                  112643                0                0                0                0      Dec 2 18:13
Discord                         hugo                    7800                0                0                0                0      Dec 2 15:51
Discord                         hugo                    7596                0                0                0                0      Dec 2 15:50
discord                         hugo                    7581                0                0                0                0      Dec 2 15:50
code                            hugo                    7447                0                0                0                0      Dec 2 15:50
code                            hugo                    7389                0                0                0                0      Dec 2 15:50
code                            hugo                    7388                0                0                0                0      Dec 2 15:50
code                            hugo                    7252                0                0                0                0      Dec 2 15:50
code                            hugo                    7230                0               62                0             6.20      Dec 2 15:50
code                            hugo                    7151                0            10248                0          1024.80      Dec 2 15:50
code                            hugo                    7114                0                0                0                0      Dec 2 15:50
code                            hugo                    7086                0                0                0                0      Dec 2 15:50
code                            hugo                    7084                0                0                0                0      Dec 2 15:50
code                            hugo                    7083                0                0                0                0      Dec 2 15:50
code                            hugo                    7080                0                0                0                0      Dec 2 15:50
chrome_crashpad                 hugo                    7099                0                0                0                0      Dec 2 15:50
chrome_crashpad                 hugo                   13984                0                0                0                0      Dec 2 17:27
chrome_crashpad                 hugo                   13982                0                0                0                0      Dec 2 17:27

hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$
```

Figura 39 - ./rwstat.sh -m 5000 -c "d." 10*

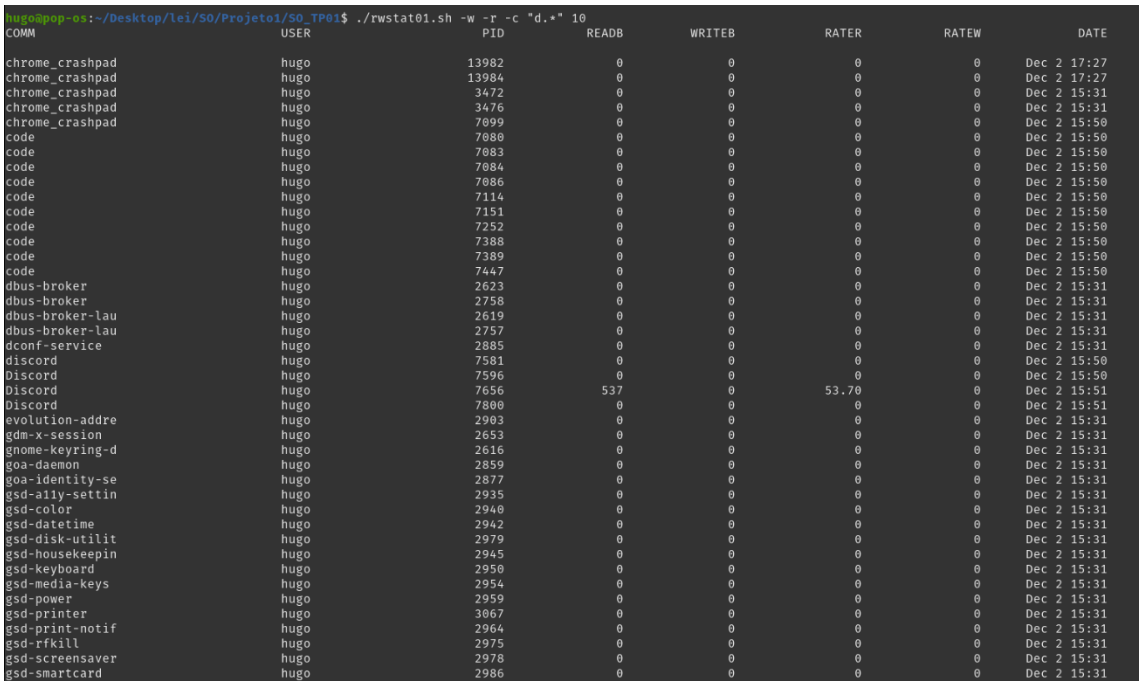
Semelhante ao que foi verificado anteriormente a partir da opção *-c* são apresentados apenas os processos que contenham “d” no seu comm. Desta vez acrescentámos a opção *-m* que, como já foi explicado anteriormente, é o número mínimo da gama de pids, tendo, neste caso, o valor de 5000. Como é verificado na Fig.39 todos os pids da tabela têm valor superior ao referido.



```
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh -w -c "d.*" 10
COMM                            USER                    PID         READB         WRITEB         RATER         RATEW          DATE
Discord                         hugo                   7642          8082         455029        808.20      45502.90      Dec 2 15:51
Discord                         hugo                   7745        872505         170735      87250.50      17073.50      Dec 2 15:51
code                            hugo                   7151             2           7711           .20        771.10      Dec 2 15:50
Discord                         hugo                   7585        400823           1409      40082.30        140.90      Dec 2 15:50
xdg-dbus-proxy                  hugo                   3321           296            888         29.60         88.80      Dec 2 15:31
gsd-media-keys                  hugo                   2954           336            656         33.60         65.60      Dec 2 15:31
gsd-power                       hugo                   2959           204            340         20.40         34.00      Dec 2 15:31
xdg-desktop-por                 hugo                   3236            88            176          8.80         17.60      Dec 2 15:31
gsd-housekeepin                 hugo                   2945            88            160          8.80         16.00      Dec 2 15:31
gsd-datetime                    hugo                   2942            96            160          9.60         16.00      Dec 2 15:31
gsd-a11y-settin                 hugo                   2935            96            160          9.60         16.00      Dec 2 15:31
gsd-xsettings                   hugo                   2998            72            128          7.20         12.80      Dec 2 15:31
gsd-sharing                     hugo                   2982            72            128          7.20         12.80      Dec 2 15:31
gsd-screensaver                 hugo                   2978            72            128          7.20         12.80      Dec 2 15:31
gsd-rfkill                      hugo                   2975            72            128          7.20         12.80      Dec 2 15:31
gsd-print-notif                 hugo                   2964            72            128          7.20         12.80      Dec 2 15:31
gsd-color                       hugo                   2940            72            128          7.20         12.80      Dec 2 15:31
xdg-desktop-por                 hugo                   3269            48             96          4.80          9.60      Dec 2 15:31
gsd-wacom                       hugo                   2997            56             96          5.60          9.60      Dec 2 15:31
gsd-sound                       hugo                   2989            48             96          4.80          9.60      Dec 2 15:31
gsd-smartcard                   hugo                   2986            56             96          5.60          9.60      Dec 2 15:31
gsd-keyboard                    hugo                   2950            56             96          5.60          9.60      Dec 2 15:31
gnome-keyring-d                 hugo                   2616            48             96          4.80          9.60      Dec 2 15:31
code                            hugo                   7230             0             93             0          9.30      Dec 2 15:50
xdg-document-po                 hugo                   3240            56             88          5.60          8.80      Dec 2 15:31
code                            hugo                   7165            58             58          5.80          5.80      Dec 2 15:50
code                            hugo                   7199            26             26          2.60          2.60      Dec 2 15:50
code                            hugo                   7197            24             24          2.40          2.40      Dec 2 15:50
code                            hugo                   7080            10             10          1.00          1.00      Dec 2 15:50
Discord                         hugo                   7656           364              3         36.40           .30      Dec 2 15:51
xdg-permission-                 hugo                   2848             0              0             0             0      Dec 2 15:31
xdg-desktop-por                 hugo                   3249             0              0             0             0      Dec 2 15:31
xdg-dbus-proxy                  hugo                   7576             0              0             0             0      Dec 2 15:50
xdg-dbus-proxy                  hugo                  11613             0              0             0             0      Dec 2 16:56
systemd                         hugo                   2597             0              0             0             0      Dec 2 15:31
pop-system-upda                 hugo                   3304             0              0             0             0      Dec 2 15:31
ibus-dconf                      hugo                   3066             0              0             0             0      Dec 2 15:31
ibus-daemon                     hugo                   2937             0              0             0             0      Dec 2 15:31
hidpi-notificat                 hugo                   2946             0              0             0             0      Dec 2 15:31
hidpi-daemon                    hugo                   3029             0              0             0             0      Dec 2 15:31
gvfs-udisks2-vo                 hugo                   2833             0              0             0             0      Dec 2 15:31
gvfsd-trash                     hugo                   2912             0              0             0             0      Dec 2 15:31
gvfsd-network                   hugo                 112627             0              0             0             0      Dec 2 18:12
gvfsd-metadata                  hugo                   3282             0              0             0             0      Dec 2 15:31
```

Figura 40 - ./rwstat.sh -w -c "d." 10*

Seguindo a mesma ideia em relação à opção -c referida acima, a tabela será novamente impressa mas, agora com a ativação da opção *-w*, por ordem decrescente das taxas de escrita (Fig.40).



```
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh -w -r -c "d.*" 10
COMM                            USER                    PID         READB         WRITEB         RATER         RATEW          DATE

chrome_crashpad                 hugo                  13982             0              0             0             0      Dec 2 17:27
chrome_crashpad                 hugo                  13984             0              0             0             0      Dec 2 17:27
chrome_crashpad                 hugo                   3472             0              0             0             0      Dec 2 15:31
chrome_crashpad                 hugo                   3476             0              0             0             0      Dec 2 15:31
chrome_crashpad                 hugo                   7099             0              0             0             0      Dec 2 15:50
code                            hugo                   7080             0              0             0             0      Dec 2 15:50
code                            hugo                   7083             0              0             0             0      Dec 2 15:50
code                            hugo                   7084             0              0             0             0      Dec 2 15:50
code                            hugo                   7086             0              0             0             0      Dec 2 15:50
code                            hugo                   7114             0              0             0             0      Dec 2 15:50
code                            hugo                   7151             0              0             0             0      Dec 2 15:50
code                            hugo                   7252             0              0             0             0      Dec 2 15:50
code                            hugo                   7388             0              0             0             0      Dec 2 15:50
code                            hugo                   7389             0              0             0             0      Dec 2 15:50
code                            hugo                   7447             0              0             0             0      Dec 2 15:50
dbus-broker                     hugo                   2623             0              0             0             0      Dec 2 15:31
dbus-broker                     hugo                   2758             0              0             0             0      Dec 2 15:31
dbus-broker-lau                 hugo                   2619             0              0             0             0      Dec 2 15:31
dbus-broker-lau                 hugo                   2757             0              0             0             0      Dec 2 15:31
dconf-service                   hugo                   2885             0              0             0             0      Dec 2 15:31
discord                         hugo                   7581             0              0             0             0      Dec 2 15:50
Discord                         hugo                   7596             0              0             0             0      Dec 2 15:50
Discord                         hugo                   7656           537              0         53.70             0      Dec 2 15:51
Discord                         hugo                   7800             0              0             0             0      Dec 2 15:51
evolution-addre                 hugo                   2903             0              0             0             0      Dec 2 15:31
gdm-x-session                   hugo                   2653             0              0             0             0      Dec 2 15:31
gnome-keyring-d                 hugo                   2616             0              0             0             0      Dec 2 15:31
goa-daemon                      hugo                   2859             0              0             0             0      Dec 2 15:31
goa-identity-se                 hugo                   2877             0              0             0             0      Dec 2 15:31
gsd-a11y-settin                 hugo                   2935             0              0             0             0      Dec 2 15:31
gsd-color                       hugo                   2940             0              0             0             0      Dec 2 15:31
gsd-datetime                    hugo                   2942             0              0             0             0      Dec 2 15:31
gsd-disk-utilit                 hugo                   2979             0              0             0             0      Dec 2 15:31
gsd-housekeepin                 hugo                   2945             0              0             0             0      Dec 2 15:31
gsd-keyboard                    hugo                   2950             0              0             0             0      Dec 2 15:31
gsd-media-keys                  hugo                   2954             0              0             0             0      Dec 2 15:31
gsd-power                       hugo                   2959             0              0             0             0      Dec 2 15:31
gsd-printer                     hugo                   3067             0              0             0             0      Dec 2 15:31
gsd-print-notif                 hugo                   2964             0              0             0             0      Dec 2 15:31
gsd-rfkill                      hugo                   2975             0              0             0             0      Dec 2 15:31
gsd-screensaver                 hugo                   2978             0              0             0             0      Dec 2 15:31
gsd-smartcard                   hugo                   2986             0              0             0             0      Dec 2 15:31
```

Figura 41 - ./rwstat.sh -w -r -c "d." 10*

Seguindo a mesma ideia em relação à opção *-c*, a tabela será novamente impressa sendo que agora com a ativação da opção *-w* e da opção *-r*, passará a ser ordenada inversamente em relação à ordem default da opção *-w*, ou seja, por ordem crescente das taxas de escrita (Fig.41).

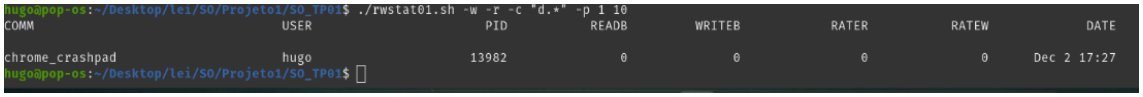


Figura 42 - ./rwstat.sh -w -r -c "d." -p 1 10*

Por fim, acrescentamos ao teste anterior a opção -p com o valor 1 associado (Fig.42). Portanto, será impresso o processo com a taxa de escrita mais pequena.

De seguida, apresentamos, como referido no início desta fase do relatório, alguns testes que considerámos relevantes:

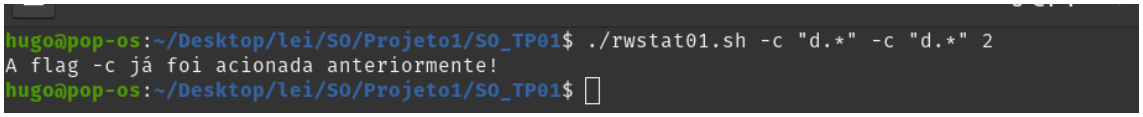


Figura 44 - ERRO: Opção passada mais que uma vez

Figura 43 -ERRO: user inválido

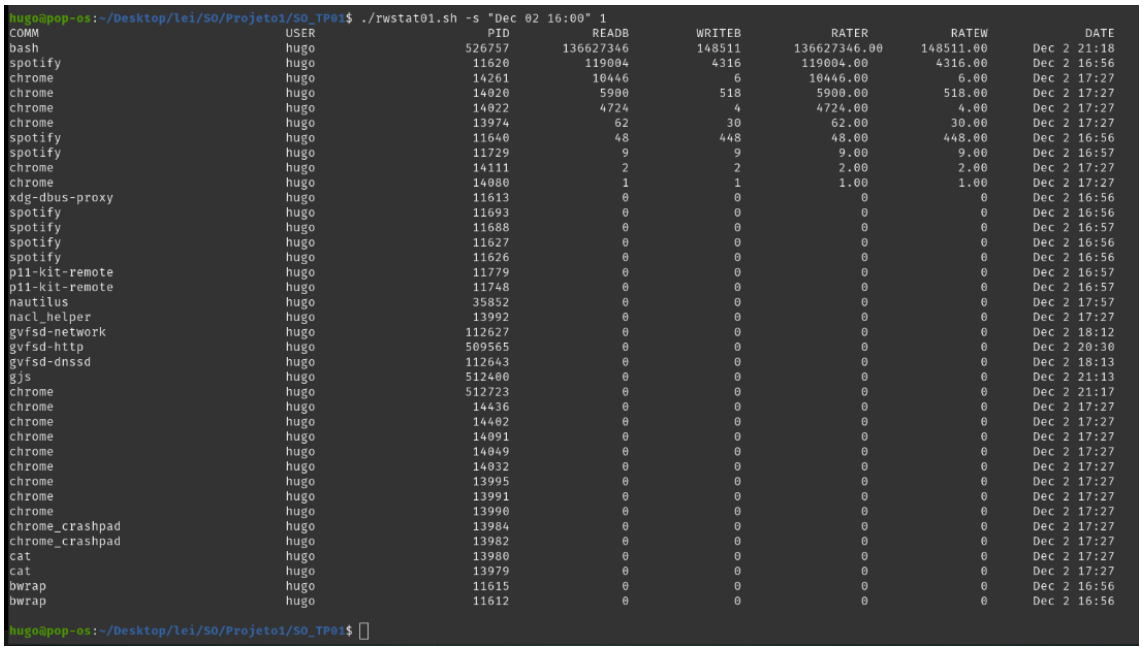


Figura 45 – Filtragem: Apenas data mínima

```
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh -e "Dec 02 15:40" 1
COMM                            USER                 PID          READB         WRITEB          RATER          RATEW           DATE
pipewire-pulse                  hugo                2614          12656            528       12656.00         528.00      Dec 2 15:31
pipewire                        hugo                2607            512            256         512.00         256.00      Dec 2 15:31
gnome-shell                     hugo                2808             72           1032          72.00        1032.00      Dec 2 15:31
chrome                          hugo                3555              1              1           1.00           1.00      Dec 2 15:31
xdg-permission-                 hugo                2848              0              0              0              0      Dec 2 15:31
xdg-document-po                 hugo                3240              0              0              0              0      Dec 2 15:31
xdg-desktop-por                 hugo                3269              0              0              0              0      Dec 2 15:31
xdg-desktop-por                 hugo                3249              0              0              0              0      Dec 2 15:31
xdg-desktop-por                 hugo                3236              0              0              0              0      Dec 2 15:31
xdg-dbus-proxy                  hugo                3321              0              0              0              0      Dec 2 15:31
wireplumber                     hugo                2612              0              0              0              0      Dec 2 15:31
tracker-miner-f                 hugo                2687              0              0              0              0      Dec 2 15:31
touchegg                        hugo                3017              0              0              0              0      Dec 2 15:31
systemd                         hugo                2597              0              0              0              0      Dec 2 15:31
sh                              hugo                2934              0              0              0              0      Dec 2 15:31
pop-system-upda                 hugo                3304              0              0              0              0      Dec 2 15:31
pipewire                        hugo                2613              0              0              0              0      Dec 2 15:31
p11-kit-server                  hugo                3303              0              0              0              0      Dec 2 15:31
p11-kit-remote                  hugo                3661              0              0              0              0      Dec 2 15:32
io.elementary.a                 hugo                3035              0              0              0              0      Dec 2 15:31
ibus-x11                        hugo                3074              0              0              0              0      Dec 2 15:31
ibus-portal                     hugo                3093              0              0              0              0      Dec 2 15:31
ibus-extension-                 hugo                3072              0              0              0              0      Dec 2 15:31
ibus-engine-sim                 hugo                3155              0              0              0              0      Dec 2 15:31
ibus-dconf                      hugo                3066              0              0              0              0      Dec 2 15:31
ibus-daemon                     hugo                2937              0              0              0              0      Dec 2 15:31
hidpi-notificat                 hugo                2946              0              0              0              0      Dec 2 15:31
hidpi-daemon                    hugo                3029              0              0              0              0      Dec 2 15:31
gvfs-udisks2-vo                 hugo                2833              0              0              0              0      Dec 2 15:31
gvfs-mtp-volume                 hugo                2842              0              0              0              0      Dec 2 15:31
gvfs-gphoto2-vo                 hugo                2838              0              0              0              0      Dec 2 15:31
gvfs-goa-volume                 hugo                2856              0              0              0              0      Dec 2 15:31
gvfsd-trash                     hugo                2912              0              0              0              0      Dec 2 15:31
gvfsd-metadata                  hugo                3282              0              0              0              0      Dec 2 15:31
gvfsd                           hugo                2626              0              0              0              0      Dec 2 15:31
gvfsd-fuse                      hugo                2631              0              0              0              0      Dec 2 15:31
gvfs-afc-volume                 hugo                2847              0              0              0              0      Dec 2 15:31
gsd-xsettings                   hugo                2998              0              0              0              0      Dec 2 15:31
gsd-wacom                       hugo                2997              0              0              0              0      Dec 2 15:31
gsd-sound                       hugo                2989              0              0              0              0      Dec 2 15:31
gsd-smartcard                   hugo                2986              0              0              0              0      Dec 2 15:31
gsd-sharing                     hugo                2982              0              0              0              0      Dec 2 15:31
```

Figura 47 - Filtragem : Apenas data máxima

```
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh -s "Dec 02 15:50" -e "Dec 02 15:40" 1
Data máxima inválida! A data máxima deve ser maior que a data mínima, ou não existir!
 ./rwstat01.sh opções:
      -c : filtra por expressão regular
      -s : data mínima para início do processo
      -e : data máxima para início do processo
      -u : seleção dos processos através do user name
      -m -M : gama de pids
      -p : número de processos a visualizar
      -r : ordenação da tabela pela ordem inversa
      -w : ordenação da tabela por valores escritos
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$
```

Figura 46 - ERRO: Período temporal inválido

```
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$ ./rwstat01.sh -m 30000000 -M 200000 1
O argumento -M deve ser maior que o argumento -m!
 ./rwstat01.sh opções:
      -c : filtra por expressão regular
      -s : data mínima para início do processo
      -e : data máxima para início do processo
      -u : seleção dos processos através do user name
      -m -M : gama de pids
      -p : número de processos a visualizar
      -r : ordenação da tabela pela ordem inversa
      -w : ordenação da tabela por valores escritos
hugo@pop-os:~/Desktop/lei/SO/Projeto1/SO_TP01$
```

Figura 48 - ERRO: Gama de pids inválida

4 - Conclusão

Concluindo, rwstat.sh permite visualizar todas as estatísticas dos processos, pedidas pelo enunciado, que estão em execução no computador do utilizador. Mais do que apenas listar, o programa também permite filtrar e ordenar consoante as opções passadas no terminal no momento de execução do script.

Este trabalho, serviu para clarificar os nossos conhecimentos em diferentes aspetos da linguagem bash. Primeiramente, aprendemos a usar arrays associativos que, de maneira muito eficiente guardam as informações através de keys, permitindo que o seu acesso seja facilitado. Aprendemos a manipular e explorar diretorias e ficheiros bem como as suas permissões.

Ao longo do projeto, o mesmo foi submetido a vários testes e guardado num repositório de GitHub para que pudéssemos estar sempre a par de erros e ultrapassá-los com mais facilidade. Acreditamos, assim, que conseguimos ir criando um programa sólido e fortificado através deste método.

Neste ponto da realização do trabalho/relatório podemos afirmar que conseguimos alcançar todos os objetivos que o guião propunha, sendo desta forma muito satisfatória a realização do projeto.

5 - Bibliografia

Na realização deste trabalho foram consultados os slides teóricos bem como os guiões práticos disponibilizados no e-learning.

Para além destes, foram visitados alguns sites entre 26/11/2022 e 02/12/2022, tais como:

- https://app.diagrams.net/
- https://stackoverflow.com/
- https://linuxhint.com/
- https://www.cyberciti.biz/
- https://devhints.io/bash