

# Taxas de leitura e escrita de processos em *bash*

Novembro/Dezembro de 2022

Relatório referente ao Trabalho Prático 1, desenvolvido no âmbito da disciplina de  
Sistemas Operativos / Turma P06  
Professor José Nuno Panelas Nunes Lau  
Professor António Guilherme Rocha Campos

# Índice

## **Introdução**

## **Passos da Resolução do Problema**

<b>Geração valores</b>	<b>6</b>
<b>Argumentos</b>	<b>8</b>
<b>Implementação dos filtros</b>	<b>12</b>

## **Testes**

<b>Testes com apenas 1/0 argumento(s)</b>	<b>18</b>
<b>Testes para 2 ou mais argumentos</b>	<b>20</b>

## **Conclusão**

# Introdução

Tal como o que está referente pelo título do trabalho, o objetivo do mesmo trata-se por realizar uma pequena descrição dos processos correntes do computador em questão, de maneira a obter um print de uma tabela que possua o nome de cada processo, o nome do utilizador que executa cada processo, o PID , que é único para cada processo e funciona como maneira de identificar os processos e por isso recebe a sigla de PID ( Process Identifier ) , o número total de *bytes* que um processo leu e escreveu desde que iniciou atividade no computador e as respetivas taxas I/O (*in and out*). No final da tabela deve existir a informação da data em que foi iniciado.

O trabalho deve ser realizado na linguagem de *bash script*, criando um programa executável em *Unix*.

Para além disso, o programa deve poder receber argumentos que especifiquem a ordem pela qual os processos aparecem no terminal ou até mesmo argumentos que filtram os processos de maneira a aparecer uma porção específica de todos os processos existentes no computador. No seguimento da afirmação anterior, passamos a citar todos os argumentos que devem ser recebidos pelo *script* e as mudanças que devem suceder no programa

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
bash	nlau	10174	24690	58597	3770687.30	14239.00	Sep 12 11:45
dropbox	sop0100	2636	918784	1356760	40022.70	31114.50	Sep 19 08:49
dice	sop0200	5036	2914000	1356	4022.70	5114.70	Sep 23 18:14

1

---

<sup>1</sup> *Print* colocada pelo professor de como devem ser apresentados os resultados

Argumentos de Entrada	Consequência no Terminal
<b>-c "re"</b>	Seleção de processos através do seu nome. A variável "re" deve conter uma expressão regular que permita filtrar os diferentes nomes e a conteúdo da variável deve ser verificado pelo programa
<b>-e "data"</b> <b>-s "data"</b>	Filtro de processos pela data de inicialização. O argumento -e funciona como uma data máxima, ou seja, aparecem todos os processos até à data inserida após o argumento em questão. No caso do argumento -s o funcionamento é similar, apenas com a diferença de que este se refere à data mínima, obtendo então todos os processos que se iniciaram naquela data ou após.
<b>-u "re"</b>	Este argumento em questão trata da filtragem pelo nome de utilizador, onde "re" é uma expressão regular, e tal como no argumento -c, deve ser posteriormente validada
<b>-m "int"</b> <b>-M "int"</b>	Idêntico ao que acontece no argumento -e e no -s, estes argumentos filtram os procedimentos por valor mínimo (-m) do PID e consequentemente por valor máximo (-M), tendo em conta que o sucessor de cada um destes argumentos é um número inteiro.
<b>-r</b> <b>-w</b>	Os dois argumentos -r e -w não recebem argumentos opcionais, tendo em conta que o uso destes 2 argumentos apenas atua sobre a maneira como os os valores aparecem no ecrã, neste caso, quando se usa o argumento -r é suposto fazer <i>print</i> da ordem reversa dos processos, já o -w atualiza a ordem da tabela pelos valores obtidos no que toca aos <i>bytes</i> escritos
<b>-p "int"</b>	Este argumento pega no valor recebido no argumento opcional e mostra apenas "int" número de processos, ou seja , se for passado o -p 2 no <i>rwstat.sh</i> , o terminal apenas dá <i>print</i> a dois processos, mais em específico, aparecerão apenas aqueles primeiros 2 processos que apareceriam em primeiro lugar caso o -o não existisse.

Para além de todos os resultados corretos, o programa deve também perceber quando os argumentos não estão de acordo com o pedido, apresentando mensagens de erro se tal se verificar. Dado que já apresentamos na totalidade aquilo em que consiste o problema, passamos aos passos de resolução.

# Passos da Resolução Do Problema

## Geração valores

Para chegarmos a uma solução para este projeto decidimos correr todos os processos através de um ciclo For e armazenar todos os valores do Read Char num array associativo chamado **allRchar** e os valores do Write Char em outro array chamado **allWchar**. Cada um desses arrays tem como índice o **PID** de cada processo. No ciclo restringimos os processos para só colocarmos no array aqueles em que temos permissão e descartamos aqueles que foram iniciados durante o período do sleep.

```

82
83 function getInitialValues(){
84     pid=$(ps -ef | grep 'p' | awk '{print $2}')
85
86     for processos in $pid ;do
87         if [ -d $processos ];then
88             cd ./processos
89             if [ -r ./io ];then
90
91                 rchar=$(cat /proc/$processos/io | grep rchar | grep -o -E '[0-9]+' )
92                 allRchar[$processos]=$rchar #allRchar guarda os valores iniciais de rchar
93
94                 wchar=$(cat /proc/$processos/io | grep wchar | grep -o -E '[0-9]+' )
95                 allWchar[$processos]=$wchar #allWchar guarda o valores iniciais de wchar
96
97             fi
98             cd ../
99         fi
100     done
101 }
102

```

Após a chamada desta função executamos a função sleep e invocamos outra função chamada getSecondValues(), onde aqui carregamos os valores do COMM (Nome do Processo) para uma variável chamada **name**, os valores de USER em **owner**, os caracteres lidos e escritos após a invocação da função sleep em **rchar2** e **wchar2**, respetivamente, e a data atual numa variável chamada **date**.

Com todos estes valores conseguimos calcular as taxas de escrita e de leitura, subtraindo os valores de **rchar2** e **wchar2** pelos armazenados anteriormente em **allRchar[PID]** e em **allWchar[PID]** e dividindo pelo tempo:

```
function getSecondValues()
for processos in $pid;do
if [ -d $processos ];then
cd $processos
if [ -r ./io ];then
name=$(grep -E "Name" ./status | awk '{print $2}')

owner=$(ps -p $processos -o user=)

rchar2=$(cat /proc/$processos/io | grep rchar | grep -o -E '[0-9]+' )

rateR=$((rchar2-allRchar[$processos]))
rateR=$((echo "scale=2; $rateR/$ultimo" | bc -l))      # calculo do rateR

wchar2=$(cat /proc/$processos/io | grep wchar | grep -o -E '[0-9]+' )

rateW=$((wchar2-allWchar[$processos]))
rateW=$((echo "scale=2; $rateW/$ultimo" | bc -l))      # calculo do rateW

date=$(ps -p $processos -o lstart=)
date=$(echo $date | awk '{print $2,$3,$4}')            #Retira o mês, o dia e Horas:Min:Seg ----> de date

month=$(echo $date | awk '{print $1}')
month=$(meses[$month])                                #Retira o mês em numero

day=$(echo $date | awk '{print $2}')
hour=$(echo $date | awk '{print $3}' | awk -F: '{print $1}')
minutes=$(echo $date | awk '{print $3}' | awk -F: '{print $2}')

AllArray[$processos]=$((printf "%-20s %-10s %-10s %-15s %-15s %-15s %-20s \n" "$name" "$owner" "$processos" "$rchar2" "$wchar2" "$rateR" "$rateW" "$date"))
fi
cd ../
done
printf '%s \n' "${AllArray[@]}"
```

Não tendo nenhuma filtragem de nome, utilizador ou qualquer outra, a nossa `getSecondValues()` pode ficar como está na figura em cima.

Após armazenarmos todos os valores que queremos em variáveis vamos guardá-las num array dinâmico chamado **AllArray**, que funciona da seguinte maneira: a chave do array associativo é o PID de cada processo enquanto o valor contém todos os valores que pretendemos que apareçam na tabela final, mas de maneira formatada.

## Argumentos

Para lidarmos com os argumentos que podem ser colocados como input usamos a função **getopts** e percorremos todos os argumentos na variável autointitulada. Como para filtrarmos os processos só podemos usar: -c, -s, -e, -u, -m, -M, -p (têm valores seguindo deles, daí os “:”) e -r, -w (não têm valores seguindo deles) só colocamos estes no while getopts.

Para armazenar os valores que seguem a opção criamos um array associativo, argOpt, que guarda a cada caractere, ‘c’, por exemplo o valor que se encontra seguindo dele, caso não exista na opção coloca 0.

Ex: input: ./rwstat.sh -c “Discord” -r 2 —> argOpt[‘c’] = “Discord”;  
argOpt[‘r’]=0;

```
while getopts "c:s:e:u:m:M:p:rw" argumentos; do
    if [[ "${OPTARG}" ]]; then
        argOpt[$argumentos]=${OPTARG}
    else
        argOpt[$argumentos]=0
    fi

    case $argumentos in
> c) ...
        ;;
> s) ...
        ;;
> e) ...
        ;;
> u) ...
        ;;
> m) ...
        ;;
> M) ...
        ;;
> p) ...
        ;;
> r) ...
        ;;
> w) ...
        ;;
        *)
            echo "COMANDO INVALIDO"
            exit
            ;;
    esac
done
```



Na situação de o “-c” ser um argumento de uma chamada válida da função `rwstat.sh` vamos entrar no caso:

```
c)
argsC=${argOpt['c']}

if [[ $argsC == -* ]]; then                                #o valor seguinte a -c tem que ser um inteiro, não pode ser outro filtro
    echo "Argumento -c não pode começar com -"
    exit
fi

if [[ $argsC =~ ^[0-9]+$ ]]; then                            #o valor depois de -c tem que ser uma string
    echo "Formato incorreto!!" -c [String nome_processo], último argumento [int TEMPO]"
    exit
fi

printf "nome do PID: %s\n" $argsC

;;
```

Neste momento vamos armazenar o valor que está anexado a “-c” e realizar duas validações. Se esse valor (`argsC`) tiver um “-” ou for um número acaba e escreve o erro no terminal. Esta situação repete se para quando o argumento for “-u”, mas o valor agora fica armazenado na variável `argsU`.

Nas situações em que o argumento é “-s” e “-e”, a validação é a idêntica entre eles:

```
s)
argsS=${argOpt['s']}

nrArgsS=$(echo $argsS | awk '{print NF}')                  #nr de argumentos do -s
mes=$(echo $argsS | awk '{print $1}')

if [ $nrArgsS -ne 3 ]; then                                #a data tem de ter 3 argumentos
    echo "Formato incorreto!!" -s [String mes] [int dia] [int hora: int minutos], último argumento [int TEMPO]"
    exit
fi

if [[ $mes =~ ^[0-9]+$ ]] || [[ ! ${meses[$mes]} ]]; then    #verifica se o mes está bem colocado
    echo "Nome do mês: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec"
    exit
fi

mes=${meses[$mes]}

dia=$(echo $argsS | awk '{print $2}')

if [[ $dia -lt 10 ]]; then
    dia="0$dia"
fi

if [[ ! $dia =~ ^[0-9]+$ ]]; then
    echo "Formato incorreto!!" -s [String mes] [int dia] [int hora: int minutos], último argumento [int TEMPO]"
    exit
fi

if [ $dia -lt 1 ] || [ $dia -gt 31 ]; then
    echo "O dia deve ser um nr entre 1 e 31"
    exit
fi

hora_Min=$(echo $argsS | awk '{print $3}')
hora=$(echo $hora_Min | awk -F: '{print $1}')
min=$(echo $hora_Min | awk -F: '{print $2}')

if [[ ${#hora_Min} -ne 5 ]] || [[ $hora_Min:2:1 != ":" ]]; then
    echo "Formato incorreto!!" -s [String mes] [int dia] [int hora: int minutos], último argumento [int TEMPO]"
    exit
fi

argsS=""
argsS=$(echo $mes $dia $hora $min | awk '{printf "%s%s%s%s", $1,$2,$3,$4}')

argsS=$(echo $argsS | awk '{print int($1)}')                #A variavel argsS é convertida para inteiro --> mesdiahoraMin
printf "Data minima (tudo junto): %d\n" $argsS

;;
```

Primeiramente, guardamos em variáveis o mês, o dia, a hora:minuto e o valor que os argumentos “-e” ou “-s” (em argsE e argsS , respetivamente). Seguidamente, verificamos se o mês é uma string, se o dia é um inteiro entre [1, 31] e se o argumento hora:min (onde a hora e o minuto têm que ter obrigatoriamente 2 dígitos) está no formato correto, e aí damos a data como validada.

Como forma de futuramente comparar as datas pensamos em concatenar o mês, o dia, a hora e os minutos num número inteiro (**MesDiaHoraMinutos**). Reparamos que ao concatenar a data que o primeiro minuto, do primeiro dia de janeiro de um ano qualquer é **01010001** é mais pequeno do que o minuto seguinte ou a hora seguinte, etc. Com isto concluímos que os números formados por esta combinação são crescentes até ao último minuto do último dia de Dezembro (**12312359**). Para converter o mês para número usamos o array associativo **meses**.

Assim, concluímos que esta poderia ser uma maneira eficaz de ordenar as datas dos processos.

Passando agora para os argumentos “-m”, “-n”, “-p”, como para todas as situações que consideramos corretas o argumento opcional que segue cada uma destas 3 opções é um número inteiro, mostraremos apenas uma validação do género, mas deixamos em nota que todos os argumentos acima mencionados são validados de maneira idêntica.

```
m)
    argsm=${argOpt['m']}

    allLastButm=$(echo $@ | awk '{for(i=1;i<NF;i++) printf "%s ",$i}') #remove o ultimo argumento

    arrm=$(echo $allLastButm | awk -F"-m" '{print $2}')

    for i in $arrm; do
        if [[ $i =~ ^- ]]; then
            break
        fi
        countm=$((countm+1))
    done

    if [ $countm -ne 1 ]; then
        echo "Formato incorreto!!      -m [int PID_minimo], último argumento [int TEMPO]"
        exit
    fi

    #check if argsm is a number
    if [[ $argsm =~ ^[0-9]+$ ]]; then
        printf "PID minimo: %d\n" $argsm
    else
        echo "Formato correto!!      -m [int PID_minimo], último argumento [int TEMPO]"
        exit
    fi
fi
;;
```

Como forma de validar a quantidade de argumentos que se encontram depois de cada um destes `-*` (tem que ser 1) criamos um array que contém os argumentos que seguem o `-m`, por exemplo. O ciclo seguinte tem como objetivo contar o número de valores que existem de um `-*` até outro `-*`. Esse valor (guardado na variável `countm` no código acima descrito) tem que ser obrigatoriamente 1.

Também temos de verificar se o valor que está depois de `-m`, `-p` ou `-M` é um inteiro, se estiver tudo nas conformidades guardamos esses valores em variáveis e programa segue.

Por último, os argumentos `-r` e `-w`, sem valores associados:

```
r)
valR=1
posR=$(echo $@ | awk '{for(i=1;i<=NF;i++) if($i=="-r") print $(i+1)}')      #argumento a seguir ao -r
penultimoR=$(echo $@ | awk '{print $(NF-1)}')                             #penultimo argumento

if [ ! $penultimoR == "-r" ]; then
    #0 Valor a seguir a -r tem que ser ou o tempo ou outro filtro
    if [ $posR == "-c" ] || [ $posR == "-s" ] || [ $posR == "-e" ] || [ $posR == "-u" ] || [ $posR == "-m" ] || [ $posR == "-M" ] || [ $posR == "-p" ] || [ $posR == "-w" ]; then
        echo
    else
        echo "Formato incorreto!! -r, último argumento [int TEMPO]"
        exit
    fi
fi
;;
```

Como estes argumentos não tem argumentos opcionais, o que vem a seguir deles é sempre um outro filtro (`-c`, ...) ou o tempo. Se tudo correr bem, vamos guardar numa variável (`valR` ou `valW`, dependendo do argumento) o valor 1 que futuramente nos vai dar informação para a forma como ordenar os processos.

## Implementação dos filtros

Depois de colocarmos os valores que queremos escrever no terminal em variáveis:

```
for processos in $pid ;do
  if [ -d $processos ];then
    cd ./processos
    if [ -r ./io ];then
      name=$(grep -E "Name" ./status | awk '{print $2}')

      owner=$(ps -p $processos -o user=)

      rchar2=$(cat /proc/$processos/io | grep rchar | grep -o -E '[0-9]+' )

      difR=$((rchar2-allRchar[$processos]))
      rateR=$(echo "scale=2; $difR/$ultimo" | bc -l) # calculo do rateR

      wchar2=$(cat /proc/$processos/io | grep wchar | grep -o -E '[0-9]+' )

      difW=$((wchar2-allWchar[$processos]))
      rateW=$(echo "scale=2; $difW/$ultimo" | bc -l) # calculo do rateW

      date=$(ps -p $processos -o lstart=)
      date=$(echo $date | awk '{print $2,$3,$4}') #Retira o mês, o dia e Horas:Min:Seg ---> de date

      month=$(echo $date | awk '{print $1}')
      month=${meses[$month]} #Retira o mês em numero

      day=$(echo $date | awk '{print $2}')
      if [[ $day -lt 10 ]]; then
        day="0$day"
      fi

      hour=$(echo $date | awk '{print $3}' | awk -F: '{print $1}')
      minutes=$(echo $date | awk '{print $3}' | awk -F: '{print $2}')

      #add month day hour and minutes but without summing them
      dateAtual=$(echo $month $day $hour $minutes | awk '{printf "%s%s%s%s", $1,$2,$3,$4}')
      #turn dateAtual into int
      dateAtual=$(echo $dateAtual | awk '{print $1}') #Data atual em int MesDiaHoraMin
    fi
  fi
done
```

Depois de todos os valores das variáveis argsU, argsC, argsm, argsM, argsS e argsE carregados ou vazios (de acordo com dos inputs do utilizador) podemos começar as validações dos processos.

Dependendo dos argumentos encontramos formas diferentes de fazer as filtrações, para “-c”, “-u”, “-m”, “-M”, “-s” e “-e” fazemos da seguinte maneira:

```
if [ -z "${argsU}" ]; then #verifica se existe algum filtro para o USER
testeU=$owner #se não existir, ou seja, se o array estiver vazio o valor testeU é igual ao USER do processo
else
testeU=$argsU #se existir, o valor testeU é a variável $argsU
fi

if [ -z "${argsC}" ]; then #verifica se existe algum filtro para o COMM
testeC=$name #se não existir, ou seja, se o array estiver vazio o valor de testeC é o nome do processo
else
testeC=$argsC #se existir, o valor de testeC é a variável $argsC
fi

if [ -z "${argsm}" ]; then #verifica se existe algum filtro para valores mínimos de PID
testem=$processos #se não existir, ou seja, se o array estiver vazio o valor de testem é o PID do processo
else
testem=$argsm #se existir, o valor de testem é a variável $argsm
fi

if [ -z "${argsM}" ]; then #verifica se existe algum filtro para valores máximos de PID
testeM=$processos #se não existir, ou seja, se o array estiver vazio o valor de testeM é o PID do processo
else
testeM=$argsM #se existir, o valor de testeM é a variável $argsM
fi

if [ -z "${argsS}" ]; then #verifica se existe algum filtro para data mínima
testeS=$dateAtual #se não existir, ou seja, se o array estiver vazio o valor de testeS é a data atual do processo
else
testeS=$argsS #se existir, o valor de testeS é a variável $argsS
fi

if [ -z "${argsE}" ]; then #verifica se existe algum filtro para data máxima
testeE=$dateAtual #se não existir, ou seja, se o array estiver vazio o valor de testeE é a data atual do processo
else
testeE=$argsE #se existir, o valor de testeE é a variável $argsE
fi

if [[ $owner == $testeU ]] && [[ $name == $testeC ]] && [[ $testem -le $processos ]] && [[ $testeM -ge $processos ]] && [[ $testeS -le $dateAtual ]] && [[ $testeE -ge $dateAtual ]]; then
AllArray[$processos]=$ (printf "%-20s %-10s %-10s %-15s %-15s %-15s %-15s %-20s \n" "$name" "$owner" "$processos" "$difR" "$difW" "$rateR" "$rateW" "$date")
fi
```

Nesta primeira sequência de condições, estas são utilizadas para verificar os filtros que estão a ser utilizados, os que não tiverem a variável teste que vão conter o valor Default. Por exemplo: um certo processo de {PID=285178; COMM=Discord; USER:tomas; READB=174178; WRITEB=40446; DATE=Dec 1 03:01:18; }:

- INPUT= ./rwstat -c “Discord” -u “tomas” -m 5 3

```
testeC = argsC = Discord;
testeU = argsU = tomas;
testem = argsm = 5;
testeM = PID = 285178;
testeS = dateAtual = 12010301
testeE = dateAtual = 12010301
```

Com todas as variáveis teste carregadas passamos para a confirmação do processo.

No exemplo anterior:

```
(tomas == testeU) && (Discord == testeC) && (testem <= 285178) &&
(testeM >= 285178) && (testeS <= 12010301) && (testeE >= 12010301)
```

Nesta situação a que referimos em cima, o processo respeitou todas as condições, sendo assim os valores do PID, COMM, ..., vão ser guardados em **AllArray**.

Para os restantes argumentos como “-p”, “-r” e “-w” apenas fazemos a verificação a seguir à dos filtros anteriores e depois do ciclo for que itera os processos:

```
if [ -z ${argsP} ]; then
    testeP=${#AllArray[@]}
else
    testeP=$argsP
    if [ $testeP -gt ${#AllArray[@]} ]; then
        echo "0 numero de processos que quer mostrar é maior que o numero de processos existentes"
        exit
    fi
fi

if [ $valR -eq 1 ]; then
    ordenRev="-n"
else
    ordenRev=""
fi

if [ $valW -eq 1 ]; then
    ordenW="-k5"
else
    ordenW="-k6"
fi

#print apenas os primeiros $testeP processos de ordem reversa ou não, ou consoante os WRITEB
printf '%s\n' "${AllArray[@]}" | sort $ordenRev $ordenW | head -n $testeP
```

A primeira condição é relativa ao argumento “-p”, se este for um dos inputs a variável testeP vai guardar o seu valor correspondente, se não, testeP vai guardar o número total de processos.

As duas condições que faltam descrever referem-se à ordenação <sup>[OBJ]</sup> das colunas, ou seja, a “-r” e “-w”, as variáveis de estado valR e valW só podem ter dois valores: 0 e 1. Quando valW for 1, vamos ordenar a tabela de acordo com a taxa de escrita, se for 0, é ordenada em ordem a taxa de leitura. Para valR, quando este valor for 1 e a ordenação é feita por ordem crescente e em ordem a taxa de leitura, se valW=0, ou em ordem à taxa de escrita, se valW=1. Quando valR=0 a situação é semelhante, mas de ordem decrescente.

Realizamos mais algumas validações para situações com número de argumentos, tipo argumentos, ...:

```
function ValArgs(){
    if [ $NrArgs -eq 0 ]; then
        echo "Não existem argumentos"
        exit 1
    fi
}

function valTempo(){
    if [[ $ultimo =~ ^[0-9]+$ ]] && [[ $ultimo -gt 0 ]]; then          #verifica se o ultimo argumento é um numero e se é maior que 0
        return 1
    else
        echo "O tempo tem que ser um inteiro e positivo"
        exit
    fi
}
```

A função ValArgs() valida o número de argumentos que tem que ser sempre pelo menos um e valTempo() valida o tempo do sleep, que tem que ser um número inteiro, positivo e o último argumento.

Além destas duas funções também criamos esta porção de código que verifica argumentos a mais que não estão a ser usados.

```
i=0
while [[ $i -lt $# ]]
do

    if [[ ${argumentos[$i]} != -* ]] && [[ $i -lt $#-1 ]]
    then
        echo "O Argumento de índice: $i e de valor: ${argumentos[$i]} não tem utilidade"
        countArg=$((countArg+1))
    fi
    if [[ ${argumentos[$i]} == "-w" ]] || [[ ${argumentos[$i]} == "-r" ]]
    then
        i=$((i+1))
    elif [[ ${argumentos[$i]} == -* ]]
    then
        i=$((i+2))          #se encontrarmos um argumento com "-" avançamos dois argumentos que são o do "-" e o seguinte
    else
        i=$((i+1))
    fi
done

if [[ $countArg -ne 0 ]]
then
    exit
fi
```

O ciclo while está a percorrer todos os argumentos e iterando de forma diferente dependendo do valor encontrado, por exemplo: ./rwstat.sh -r -w -c "a" "s" 2:

Neste caso os i's vão ser 0, 1, 2 e 4, cujos seus valores vão ser -r, -w, -c, "s" esta situação deve dar erro no argumento "s" que não tem utilidade nenhuma.

Quando encontramos um "-r", "-w", ou outro qualquer que não seja considerado um argumento em getops, que não tem argumento opcional

seguimos para o próximo argumento, se encontrarmos outro “-” , avançamos duas vezes , uma vez que eles têm obrigatoriamente 1 argumento opcional. Ao encontrar um valor que não desejamos, vamos sair do programa.

No início do programa declaramos as variáveis:

```
ultimo=${@: -1}           # guarda o ultimo argumento --> tempo
NrArgs=$#                 # guarda o numero total de argumentos
declare -a argumentos=("${@}") # guarda os argumentos numa array

declare -A argOpt=()

declare -A AllArray        #declarar um array associativo para guardar a informação de todos os processos

valR=0                    #variavel para distinguir quando alterar a ordem de AllArray
valW=0                    #variavel para distinguir quando alterar a ordem de AllArray apartir dos WRITEB

countM=0
countm=0
countP=0

countArg=0                #variavel que conta os argumentos sem significado

declare -A meses
meses=(["Jan"]=01 ["Feb"]=02 ["Mar"]=03 ["Apr"]=04 ["May"]=05 ["Jun"]=06 ["Jul"]=07 ["Aug"]=08 ["Sep"]=09 ["Oct"]=10 ["Nov"]=11 ["Dec"]=12)
```

Por último, vamos invocar todas as funções:

```
#!/usr/bin/perl
#//verificar argumentos sem significado//
i=0
while [[ $i -lt $# ]]
do
done

if [[ $countArg -ne 0 ]]
then
    exit
fi
#//verificar argumentos sem significado//

while getopts "c:s:e:u:m:M:p:r:w" argumentos; do
done

#main function
function main(){

    cd /proc/

    ValArgs          #validações
    valTempo

    printTable        #printa o cabeçalho

    getInitialValues

    sleep $ultimo
    printf "\n\n"

    getSecondValues
}

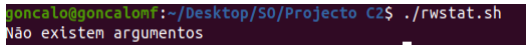
main
```



# Testes

Após a apresentação do código, passamos a mostrar o que o programa faz em todas as situações que consideramos possíveis de acontecer.

## Testes com apenas 1/0 argumento(s)

Na Teoria	Na Prática
Na teoria, quando se corre o programa na prompt do Unix sem qualquer tipo de argumento anexado o programa deveria dar erro, porque precisa de pelo menos um argumento onde se passa um inteiro que significa os segundos de diferença entre as leituras, usado para fazer a diferença nas taxas de escrita e leitura apresentadas.	 <p>O Programa corre e deparando-se com a falta de argumentos, dá <i>print</i> de uma mensagem de erro no terminal</p>

Já para apenas um argumento, temos várias opções para testar, apesar de ele só poder receber um inteiro com valor maior que 0.

Sendo assim, testamos o programa anexado por:

1. Um inteiro positivo
2. Um inteiro negativo
3. 0
4. Uma outra opção de argumento
5. Uma string (Array de Caracteres)
6. Um caractere

## Teste 1

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh 1
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
rwstat.sh	goncalo	119956	122069850	178489	116981296.00	57761.00	Dec 2 00:49:52
spotify	goncalo	15943	511492069	216519097	121337.00	7544.00	Dec 1 17:55:49
Xorg	goncalo	1780	156549377	1151594710	6816.00	88300.00	Dec 1 16:48:12
pulseaudio	goncalo	1689	101417339	104227147	5663.00	5823.00	Dec 1 16:48:12
brave	goncalo	92555	41181	10	3036.00	0	Dec 2 00:40:53
brave	goncalo	92525	538095	296937	1523.00	0	Dec 2 00:40:52
brave	goncalo	45868	3128901	190799	1522.00	0	Dec 1 21:38:26
brave	goncalo	18438	99964700	4893	1522.00	0	Dec 1 18:47:55
brave	goncalo	2579	1128648110	1127625338	995.00	835.00	Dec 1 16:48:19
code	goncalo	38913	100240843	15744751	683.00	683.00	Dec 1 20:17:45
brave	goncalo	2615	2640376780	2641331456	118.00	609.00	Dec 1 16:48:19
spotify	goncalo	16034	792714	17673915	51.00	718.00	Dec 1 17:55:49
brave	goncalo	3994	889507212	8711585	44.00	4.00	Dec 1 16:49:13
brave	goncalo	2614	4851929	84116102	44.00	711.00	Dec 1 16:48:19
code	goncalo	38961	156756063	7571906	29.00	29.00	Dec 1 20:17:46
code	goncalo	39422	22857351	251331	13.00	13.00	Dec 1 20:18:32
spotify	goncalo	16103	505467901	1963404	11.00	11.00	Dec 1 17:55:49
spotify	goncalo	16046	141211213	52693635	10.00	7274.00	Dec 1 17:55:49
code	goncalo	39100	81858133	2176269	8.00	8.00	Dec 1 20:17:48
brave	goncalo	54291	11278867	1709848	5.00	5.00	Dec 1 23:03:25
brave	goncalo	2763	2466781	301626	2.00	2.00	Dec 1 16:48:20
xdg-permission-	goncalo	2040	22828	4475	0	0	Dec 1 16:48:13
xdg-document-po	goncalo	2279	79649	71962	0	0	Dec 1 16:48:13
xdg-desktop-por	goncalo	2413	9101016	126670	0	0	Dec 1 16:48:15
xdg-desktop-por	goncalo	2409	97731	105122	0	0	Dec 1 16:48:15
update-notifier	goncalo	3983	132146629	14462786	0	0	Dec 1 16:49:13
spotify	goncalo	16040	92220	36512	0	0	Dec 1 17:55:49
spotify	goncalo	16015	103973	36	0	0	Dec 1 17:55:49
spotify	goncalo	16014	103971	20	0	0	Dec 1 17:55:49
snap-store	goncalo	2248	19441996	23970329	0	0	Dec 1 16:48:13
script-fu	goncalo	54733	390946	38821	0	0	Dec 1 23:05:33
rwstat.sh	goncalo	62170	31451214	134437	0	0	Dec 1 23:51:29
ps	goncalo	63731	219201	0	0	0	Dec 1 23:51:31
nautilus	goncalo	54379	30692951	71574801	0	0	Dec 1 23:04:17
ibus-portal	goncalo	2022	62491	66569	0	0	Dec 1 16:48:13
ibus-engine-sim	goncalo	2229	66255	62552	0	0	Dec 1 16:48:13
ibus-daemon	goncalo	2013	150546	161585	0	0	Dec 1 16:48:13
gvfs-mtp-volume	goncalo	1765	49292	19094	0	0	Dec 1 16:48:12
gvfs-gphoto2-vo	goncalo	1739	1302079	19398	0	0	Dec 1 16:48:12
gvfsd-trash	goncalo	2102	238160	30376	0	0	Dec 1 16:48:13
gvfsd-recent	goncalo	40779	9269418	15256	0	0	Dec 1 20:32:01
gvfsd-network	goncalo	4586	35557	12113	0	0	Dec 1 16:58:04
gvfsd-http	goncalo	15567	291179	364301	0	0	Dec 1 17:45:44
gvfsd-dnssd	goncalo	4600	44875	22072	0	0	Dec 1 16:58:05
gsd-usb-protect	goncalo	2140	38474	23475	0	0	Dec 1 16:48:13
gsd-screensaver	goncalo	2130	33132	20372	0	0	Dec 1 16:48:13
gsd-print-notif	goncalo	2124	65381	19440	0	0	Dec 1 16:48:13
gsd-printer	goncalo	2244	80399	5640	0	0	Dec 1 16:48:13
gsd-power	goncalo	2121	456684	66910	0	0	Dec 1 16:48:13
gsd-housekeepin	goncalo	2113	9144398	21473	0	0	Dec 1 16:48:13
gnome-calendar	goncalo	14239	394031	34998	0	0	Dec 1 17:26:02
gimp	goncalo	54641	82211832	5470540487	0	0	Dec 1 23:05:32

## Teste 2

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -1
./rwstat.sh: illegal option -- 1
COMANDO INVALIDO
```

## Teste 3

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh 0
0 tempo tem que ser um inteiro e positivo
```

## Teste 4

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -x
./rwstat.sh: illegal option -- x
COMANDO INVALIDO
```

## Teste 5

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh randomString  
0 tempo tem que ser um inteiro e positivo
```

## Teste 6

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh c  
0 tempo tem que ser um inteiro e positivo
```

Assim conseguimos concluir que tal como o previsto, todos os programas terminaram vista a um erro a não ser aquele onde o argumento passado era um número inteiro positivo. Todos os testes que deram erro apresentaram o mesmo erro, menos os testes realizados a outro qualquer argumento e ao número inteiro negativo, que tiveram problemas ao entrar na função *getopts* devido ao caractere especial "-". Esse caractere garante que ele entra na função, e como não existe nenhum "-x" ou "-1" predefinidos nas opções da função, ele vai para o default, onde dá *print* de "COMANDO INVÁLIDO"

## Testes para 2 ou mais argumentos

Para testar todos os argumentos da função e verificar se todos funcionam de maneira correta vamos correr os mesmos testes para todos os diferentes argumentos, de maneira a manter a coerência dos testes. Para tal, escolhemos os seguintes testes:

Teste para -c:

1. Testar para quando o -c não tem argumento;
2. Testar -c para quando o input não está no formato certo, -c tem que ser seguido por uma string e só então o tempo;
3. Testar -c sem a variável do tempo;
4. Testar -c no formato correto com a string="c\*";
5. Testar da mesma maneira que o teste 4 mas com -r (reverse);
6. Testar -c com outro filtro no formato certo;
7. Testar -c com mais argumentos seguintes que o necessário.

1.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -c
./rwstat.sh: option requires an argument -- c
COMANDO INVALIDO
```

2.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -c 2
Formato incorreto!! -c [String nome_processo], último argumento [int TEMPO]
```

3.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -c "c*"
nome do PID: c*
O tempo tem que ser um inteiro e positivo
```

4.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -c "c*" 2
nome do PID: c*
COMM      USER      PID      READB      WRITEB      RATER      RATEW      DATE
code       goncalo    38913    102509571  18378840    248.00     248.00     Dec 1 20:17:45
code       goncalo    39422    23137033   320013      6.50       0          Dec 1 20:18:32
code       goncalo    39100    90133344   2429761     4.00       4.00       Dec 1 20:17:48
code       goncalo    38961    207515620  7865332     4.00       10.50      Dec 1 20:17:46
code       goncalo    39638    405655     1010        0          0          Dec 1 20:21:13
code       goncalo    39423    770650     6025        0          0          Dec 1 20:18:32
code       goncalo    39129    1230352    182539      0          15.50      Dec 1 20:17:48
code       goncalo    39076    40221      805102      0          0          Dec 1 20:17:47
code       goncalo    38949    2065319    606486      0          0          Dec 1 20:17:45
code       goncalo    38916    97894      4           0          0          Dec 1 20:17:45
code       goncalo    38915    404509     217598      0          0          Dec 1 20:17:45
chrome_crashpad goncalo    2583     7700       0           0          0          Dec 1 16:48:19
chrome_crashpad goncalo    2581     7832       72          0          0          Dec 1 16:48:19
```

5.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -r -c "c*" 2
nome do PID: c*
COMM      USER      PID      READB      WRITEB      RATER      RATEW      DATE
chrome_crashpad goncalo    2581     7832       72          0          0          Dec 1 16:48:19
chrome_crashpad goncalo    2583     7700       0           0          0          Dec 1 16:48:19
code       goncalo    38915    404509     217598      0          0          Dec 1 20:17:45
code       goncalo    38916    97894      4           0          0          Dec 1 20:17:45
code       goncalo    38949    2086939    613691      0          0          Dec 1 20:17:45
code       goncalo    39129    1230352    184213      0          15.50      Dec 1 20:17:48
code       goncalo    39423    770922     6451        0          0          Dec 1 20:18:32
code       goncalo    39638    405655     1010        0          0          Dec 1 20:21:13
code       goncalo    39100    90564438   2437932     4.00       4.00       Dec 1 20:17:48
code       goncalo    39076    40425      810786      5.00       5.00       Dec 1 20:17:47
code       goncalo    39422    23158215   321013      6.50       6.50       Dec 1 20:18:32
code       goncalo    38961    210331395  7887474     14.50      14.50      Dec 1 20:17:46
code       goncalo    38913    103074077  18407424    422.50     422.50     Dec 1 20:17:45
```

6.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -e "Dec 2 02:00" -c "c*" 2
data maxima (tudo junto): 12020200
nome do PID: c*
COMM          USER      PID        READB      WRITEB      RATER      RATEW      DATE
code          goncalo    38913      103087852  18466886    202.00     203.50     Dec 1 20:17:45
code          goncalo    38961      210332208  7888287     10.50      4.00       Dec 1 20:17:46
code          goncalo    39638      405655     1010        0          0          Dec 1 20:21:13
code          goncalo    39423      770922     6451        0          0          Dec 1 20:18:32
code          goncalo    39422      23158579   321377      0          6.50      Dec 1 20:18:32
code          goncalo    39129      1230352    185081      0          0          Dec 1 20:17:48
code          goncalo    39100      90989363   2438302     0          0          Dec 1 20:17:48
code          goncalo    39076      40425      810786      0          0          Dec 1 20:17:47
code          goncalo    38949      2117290    627948      0          0          Dec 1 20:17:45
code          goncalo    38916      97894      4           0          0          Dec 1 20:17:45
code          goncalo    38915      404509     217598      0          0          Dec 1 20:17:45
chrome_crashpad goncalo    2583      7700       0           0          0          Dec 1 16:48:19
chrome_crashpad goncalo    2581      7832       72          0          0          Dec 1 16:48:19
```

7.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -c "c*" "tomas" 2
0 Argumento de índice: 2 e de valor: tomas não tem utilidade
```

Teste para -s e -e:

8. Testar para -s com o formato certo;
9. Testar para quanto o -e e o -s não têm nenhum argumento ou os argumentos não estão no formato certo;
10. Testar para -s e -e em simultâneo;
11. Testar para -e com e sem outro filtro;

8.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -s "Dec 2 12:00" 2
Data minima (tudo junto): 12021200
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
rwstat.sh	goncalo	37083	96544153	49278	48272076.50	24639.00	Dec 2 16:48:59
spotify	goncalo	4133	123406	10245	61703.00	5122.50	Dec 2 16:13:09
brave	goncalo	2452	38352	396	19176.00	198.00	Dec 2 16:10:57
pulseaudio	goncalo	1647	9270	9758	4635.00	4879.00	Dec 2 16:10:42
brave	goncalo	37034	4585	22	2292.50	11.00	Dec 2 16:48:30
brave	goncalo	2488	4355	27	2177.50	13.50	Dec 2 16:10:57
brave	goncalo	37036	1516	0	758.00	0	Dec 2 16:48:30
code	goncalo	16294	552	555	276.00	277.50	Dec 2 16:28:37
brave	goncalo	3105	517	29	258.50	14.50	Dec 2 16:11:10
Xorg	goncalo	1737	224	14728	112.00	7364.00	Dec 2 16:10:42
brave	goncalo	2487	69	69	34.50	34.50	Dec 2 16:10:57
code	goncalo	16343	29	29	14.50	14.50	Dec 2 16:28:37
code	goncalo	16484	13	13	6.50	6.50	Dec 2 16:28:40
spotify	goncalo	4266	9	549	4.50	274.50	Dec 2 16:13:12
code	goncalo	16483	8	8	4.00	4.00	Dec 2 16:28:40
xdg-permission-	goncalo	1997	0	0	0	0	Dec 2 16:10:44
xdg-document-po	goncalo	2236	0	0	0	0	Dec 2 16:10:44
xdg-desktop-por	goncalo	2370	0	0	0	0	Dec 2 16:10:45
xdg-desktop-por	goncalo	2366	0	0	0	0	Dec 2 16:10:45
update-notifier	goncalo	3588	0	0	0	0	Dec 2 16:11:44
spotify	goncalo	4294	0	0	0	0	Dec 2 16:13:12
spotify	goncalo	4261	0	0	0	0	Dec 2 16:13:12
spotify	goncalo	4236	0	0	0	0	Dec 2 16:13:10
spotify	goncalo	4221	0	0	0	0	Dec 2 16:13:10
spotify	goncalo	4220	0	0	0	0	Dec 2 16:13:10
snap-store	goncalo	2194	0	0	0	0	Dec 2 16:10:44
script-fu	goncalo	4968	0	0	0	0	Dec 2 16:17:20
nautilus	goncalo	16898	0	0	0	0	Dec 2 16:34:49
ibus-portal	goncalo	1980	0	0	0	0	Dec 2 16:10:43
ibus-engine-sim	goncalo	2179	0	0	0	0	Dec 2 16:10:44
ibus-daemon	goncalo	1970	0	0	0	0	Dec 2 16:10:43
gvfs-mtp-volume	goncalo	1722	0	0	0	0	Dec 2 16:10:42
gvfs-gphoto2-vo	goncalo	1698	0	0	0	0	Dec 2 16:10:42
gvfsd-trash	goncalo	2059	0	0	0	0	Dec 2 16:10:44
gvfsd-network	goncalo	4710	0	0	0	0	Dec 2 16:17:07
gvfsd-dnssd	goncalo	4724	0	0	0	0	Dec 2 16:17:08
gsd-usb-protect	goncalo	2105	0	0	0	0	Dec 2 16:10:44
gsd-screensaver	goncalo	2092	0	0	0	0	Dec 2 16:10:44
gsd-print-notif	goncalo	2085	0	0	0	0	Dec 2 16:10:44
gsd-printer	goncalo	2206	0	0	0	0	Dec 2 16:10:44
gsd-power	goncalo	2079	0	0	0	0	Dec 2 16:10:44
gsd-housekeepin	goncalo	2073	0	0	0	0	Dec 2 16:10:44
gnome-calendar	goncalo	4746	0	0	0	0	Dec 2 16:17:17
gimp	goncalo	4870	0	0	0	0	Dec 2 16:17:18
gdm-x-session	goncalo	1735	0	0	0	0	Dec 2 16:10:42
dbus-daemon	goncalo	1878	0	0	0	0	Dec 2 16:10:43
dbus-daemon	goncalo	1653	0	0	0	0	Dec 2 16:10:42
code	goncalo	16566	0	0	0	0	Dec 2 16:28:40
code	goncalo	16513	0	31	0	15.50	Dec 2 16:28:40
code	goncalo	16458	0	0	0	0	Dec 2 16:28:39
code	goncalo	16330	0	7200	0	3600.00	Dec 2 16:28:37

9.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -e
./rwstat.sh: option requires an argument -- e
COMANDO INVALIDO
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -s
./rwstat.sh: option requires an argument -- s
COMANDO INVALIDO
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -s 2
Formato incorreto!! -s [String mes] [int dia] [int hora: int minutos], último argumento [int TEMPO]
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -e 2
Formato incorreto!! -e [String mes] [int dia] [int hora: int minutos], último argumento [int TEMPO]
goncalo@goncalomf:~/Desktop/S0/Projecto C2$
```

10.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -e "Dec 2 01:00" -r 2
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
Data máxima (tudo junto): 12020100							

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -s "Dec 2 12:00" -r 2
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
Data mínima (tudo junto): 12021200							
at-spi2-registr	goncalo	1993	0	0	0	0	Dec 2 16:10:43
at-spi-bus-laun	goncalo	1873	0	0	0	0	Dec 2 16:10:43
bash	goncalo	4551	0	0	0	0	Dec 2 16:15:17
brave	goncalo	2462	0	0	0	0	Dec 2 16:10:57
brave	goncalo	2463	0	0	0	0	Dec 2 16:10:57
brave	goncalo	2465	0	0	0	0	Dec 2 16:10:57
brave	goncalo	2555	0	0	0	0	Dec 2 16:11:00
brave	goncalo	2565	0	0	0	0	Dec 2 16:11:00
brave	goncalo	2646	0	0	0	0	Dec 2 16:11:00
brave	goncalo	2735	0	0	0	0	Dec 2 16:11:00
brave	goncalo	2908	0	0	0	0	Dec 2 16:11:02
brave	goncalo	3007	1	1	.50	.50	Dec 2 16:11:05
brave	goncalo	3111	0	0	0	0	Dec 2 16:11:10
brave	goncalo	3202	0	0	0	0	Dec 2 16:11:11
brave	goncalo	3215	0	0	0	0	Dec 2 16:11:11
brave	goncalo	3229	0	0	0	0	Dec 2 16:11:11
brave	goncalo	3267	0	0	0	0	Dec 2 16:11:12
brave	goncalo	3319	0	0	0	0	Dec 2 16:11:14
brave	goncalo	3336	0	0	0	0	Dec 2 16:11:14
brave	goncalo	37036	0	0	0	0	Dec 2 16:48:30
chrome_crashpad	goncalo	2454	0	0	0	0	Dec 2 16:10:57
chrome_crashpad	goncalo	2456	0	0	0	0	Dec 2 16:10:57
code	goncalo	16296	0	0	0	0	Dec 2 16:28:37
code	goncalo	16397	0	0	0	0	Dec 2 16:28:37
code	goncalo	16330	0	0	0	0	Dec 2 16:28:37
code	goncalo	16458	0	0	0	0	Dec 2 16:28:39
code	goncalo	16513	0	31	0	15.50	Dec 2 16:28:40
code	goncalo	16566	0	0	0	0	Dec 2 16:28:40
dbus-daemon	goncalo	1653	0	0	0	0	Dec 2 16:10:42
dbus-daemon	goncalo	1878	0	0	0	0	Dec 2 16:10:43
gdm-x-session	goncalo	1735	0	0	0	0	Dec 2 16:10:42
gimp	goncalo	4870	0	0	0	0	Dec 2 16:17:18
gnome-calendar	goncalo	4746	0	0	0	0	Dec 2 16:17:17
gsd-housekeepin	goncalo	2073	0	0	0	0	Dec 2 16:10:44
gsd-power	goncalo	2079	0	0	0	0	Dec 2 16:10:44
gsd-printer	goncalo	2206	0	0	0	0	Dec 2 16:10:44
gsd-print-notif	goncalo	2085	0	0	0	0	Dec 2 16:10:44
gsd-screensaver	goncalo	2092	0	0	0	0	Dec 2 16:10:44
gsd-usb-protect	goncalo	2105	0	0	0	0	Dec 2 16:10:44
gvfsd-dnssd	goncalo	4724	0	0	0	0	Dec 2 16:17:08
gvfsd-network	goncalo	4710	0	0	0	0	Dec 2 16:17:07
gvfsd-trash	goncalo	2059	0	0	0	0	Dec 2 16:10:44
gvfs-gphoto2-vo	goncalo	1698	0	0	0	0	Dec 2 16:10:42
gvfs-mtp-volume	goncalo	1722	0	0	0	0	Dec 2 16:10:42
ibus-daemon	goncalo	1970	0	0	0	0	Dec 2 16:10:43
ibus-engine-sim	goncalo	2179	0	0	0	0	Dec 2 16:10:44

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -s "Dec 2 12:00" -e "Dec 2 17:30" 2
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
Data mínima (tudo junto): 12021200							
Data máxima (tudo junto): 12021730							
rwstat.sh	goncalo	73780	96635932	48722	48317966.00	24361.00	Dec 2 17:15:18
brave	goncalo	2488	4905581	4896601	2452790.50	2448300.50	Dec 2 16:10:57
gsd-housekeepin	goncalo	2073	18715	0	9357.50	0	Dec 2 16:10:44
spotify	goncalo	4133	9330	2579	4665.00	1289.50	Dec 2 16:13:09
brave	goncalo	2452	4418	306	2209.00	153.00	Dec 2 16:10:57
brave	goncalo	3007	4233	17	2116.50	8.50	Dec 2 16:11:05
pulseaudio	goncalo	1647	2902	3350	1451.00	1675.00	Dec 2 16:10:42
brave	goncalo	3079	2663	1951	1331.50	975.50	Dec 2 16:11:08
brave	goncalo	3098	618	618	309.00	309.00	Dec 2 16:11:10
Xorg	goncalo	1737	160	9432	80.00	4716.00	Dec 2 16:10:42
code	goncalo	16343	29	29	14.50	14.50	Dec 2 16:28:37
code	goncalo	16294	18	18	9.00	9.00	Dec 2 16:28:37
code	goncalo	16484	13	13	6.50	6.50	Dec 2 16:28:40
spotify	goncalo	4266	9	549	4.50	274.50	Dec 2 16:13:12
code	goncalo	16483	8	8	4.00	4.00	Dec 2 16:28:40
xdg-permission-	goncalo	1997	0	0	0	0	Dec 2 16:10:44
xdg-document-po	goncalo	2236	0	0	0	0	Dec 2 16:10:44
xdg-desktop-por	goncalo	2370	0	0	0	0	Dec 2 16:10:45
xdg-desktop-por	goncalo	2366	0	0	0	0	Dec 2 16:10:45
update-notifier	goncalo	3588	0	0	0	0	Dec 2 16:11:44
spotify	goncalo	4294	0	0	0	0	Dec 2 16:13:12
spotify	goncalo	4261	0	0	0	0	Dec 2 16:13:12
spotify	goncalo	4236	0	0	0	0	Dec 2 16:13:10
spotify	goncalo	4221	0	0	0	0	Dec 2 16:13:10
spotify	goncalo	4228	0	0	0	0	Dec 2 16:13:10
snap-store	goncalo	2194	0	0	0	0	Dec 2 16:10:44
script-fu	goncalo	4968	0	0	0	0	Dec 2 16:17:20
nautilus	goncalo	49595	0	0	0	0	Dec 2 17:01:39
ibus-portal	goncalo	1980	0	0	0	0	Dec 2 16:10:43
ibus-engine-sim	goncalo	2179	0	0	0	0	Dec 2 16:10:44
ibus-daemon	goncalo	1970	0	0	0	0	Dec 2 16:10:43
gvfs-mtp-volume	goncalo	1722	0	0	0	0	Dec 2 16:10:42
gvfs-gphoto2-vo	goncalo	1698	0	0	0	0	Dec 2 16:10:42
gvfsd-trash	goncalo	2059	0	0	0	0	Dec 2 16:10:44
gvfsd-network	goncalo	4710	0	0	0	0	Dec 2 16:17:07
gvfsd-dnssd	goncalo	4724	0	0	0	0	Dec 2 16:17:08
gsd-usb-protect	goncalo	2105	0	0	0	0	Dec 2 16:10:44
gsd-screensaver	goncalo	2092	0	0	0	0	Dec 2 16:10:44
gsd-print-notif	goncalo	2085	0	0	0	0	Dec 2 16:10:44
gsd-printer	goncalo	2206	0	0	0	0	Dec 2 16:10:44
gsd-power	goncalo	2079	0	0	0	0	Dec 2 16:10:44
gnome-calendar	goncalo	4746	0	0	0	0	Dec 2 16:17:17
gimp	goncalo	4870	0	0	0	0	Dec 2 16:17:18
gdm-x-session	goncalo	1735	0	0	0	0	Dec 2 16:10:42
dbus-daemon	goncalo	1878	0	0	0	0	Dec 2 16:10:43
dbus-daemon	goncalo	1653	0	0	0	0	Dec 2 16:10:42
code	goncalo	16566	0	0	0	0	Dec 2 16:28:40
code	goncalo	16513	0	31	0	15.50	Dec 2 16:28:40



11.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -c "c*" -e "Dec 2 17:30" 2
nome do PID: c*
Data máxima (tudo junto): 12021730
COMM      USER      PID      READB      WRITEB      RATER      RATEW      DATE
code      goncalo    16294    1068      1068      534.00     534.00     Dec 2 16:28:37
code      goncalo    16343     29        29        14.50      14.50      Dec 2 16:28:37
code      goncalo    16484     13        13         6.50       6.50      Dec 2 16:28:40
code      goncalo    16483      8         8          4.00       4.00      Dec 2 16:28:40
code      goncalo    16566      0         0           0           0      Dec 2 16:28:40
code      goncalo    16513      0        31         0        15.50      Dec 2 16:28:40
code      goncalo    16458      0         0           0           0      Dec 2 16:28:39
code      goncalo    16330      0         0           0           0      Dec 2 16:28:37
code      goncalo    16297      0         0           0           0      Dec 2 16:28:37
code      goncalo    16296      0         0           0           0      Dec 2 16:28:37
chrome_crashpad goncalo    2456      0         0           0           0      Dec 2 16:10:57
chrome_crashpad goncalo    2454      0         0           0           0      Dec 2 16:10:57
```

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -e "Dec 2 19:00" 2
Data máxima (tudo junto): 12021900
COMM      USER      PID      READB      WRITEB      RATER      RATEW      DATE
rwstat.sh goncalo    40994    96543288    49273    48271644.00  24636.50   Dec 2 16:49:14
spotify   goncalo    4133     30373     10830    15186.50    5415.00    Dec 2 16:13:09
pulseaudio goncalo    1647     9322     9891     4661.00    4945.50    Dec 2 16:10:42
brave     goncalo    37034    4584      21      2292.00     10.50      Dec 2 16:48:30
brave     goncalo    3229     1513      0       756.50      0          Dec 2 16:11:11
brave     goncalo    2452     1234     160031    617.00     80015.50   Dec 2 16:10:57
code      goncalo    16294    1206     1206     603.00      603.00     Dec 2 16:28:37
brave     goncalo    3105     268      20      134.00      10.00      Dec 2 16:11:10
Xorg      goncalo    1737     265     13449    132.50     6724.50    Dec 2 16:10:42
brave     goncalo    3098     72       8        36.00       4.00      Dec 2 16:11:10
brave     goncalo    2488     66       18       33.00       9.00      Dec 2 16:10:57
brave     goncalo    2487     38       33       19.00      16.50      Dec 2 16:10:57
code      goncalo    16343     29       29       14.50      14.50      Dec 2 16:28:37
spotify   goncalo    4266     15       15       7.50       7.50      Dec 2 16:13:12
code      goncalo    16484     13       13       6.50       6.50      Dec 2 16:28:40
code      goncalo    16483      8        8        4.00       4.00      Dec 2 16:28:40
xdg-permission- goncalo    1997      0         0           0           0      Dec 2 16:10:44
xdg-document-po goncalo    2236      0         0           0           0      Dec 2 16:10:44
xdg-desktop-por goncalo    2370      0         0           0           0      Dec 2 16:10:45
xdg-desktop-por goncalo    2366      0         0           0           0      Dec 2 16:10:45
update-notifier goncalo    3588      0         0           0           0      Dec 2 16:11:44
spotify   goncalo    4294      0         0           0           0      Dec 2 16:13:12
spotify   goncalo    4261      0         0           0           0      Dec 2 16:13:12
spotify   goncalo    4236      0         0           0           0      Dec 2 16:13:10
spotify   goncalo    4221      0         0           0           0      Dec 2 16:13:10
spotify   goncalo    4220      0         0           0           0      Dec 2 16:13:10
snap-store goncalo    2194      0         0           0           0      Dec 2 16:10:44
script-fu goncalo    4968      0         0           0           0      Dec 2 16:17:20
nautilus  goncalo    16898     0         0           0           0      Dec 2 16:34:49
ibus-portal goncalo    1980      0         0           0           0      Dec 2 16:10:43
ibus-engine-sim goncalo    2179      0         0           0           0      Dec 2 16:10:44
ibus-daemon goncalo    1970      0         0           0           0      Dec 2 16:10:43
gvfs-mtp-volume goncalo    1722      0         0           0           0      Dec 2 16:10:42
gvfs-gphoto2-vo goncalo    1698      0         0           0           0      Dec 2 16:10:42
gvfsd-trash goncalo    2059      0         0           0           0      Dec 2 16:10:44
gvfsd-network goncalo    4710      0         0           0           0      Dec 2 16:17:07
gvfsd-dnssd goncalo    4724      0         0           0           0      Dec 2 16:17:08
gsd-usb-protect goncalo    2105      0         0           0           0      Dec 2 16:10:44
gsd-screensaver goncalo    2092      0         0           0           0      Dec 2 16:10:44
gsd-print-notif goncalo    2085      0         0           0           0      Dec 2 16:10:44
gsd-printer   goncalo    2206      0         0           0           0      Dec 2 16:10:44
gsd-power     goncalo    2079      0         0           0           0      Dec 2 16:10:44
gsd-housekeepin goncalo    2073      0         0           0           0      Dec 2 16:10:44
gnome-calendar goncalo    4746      0         0           0           0      Dec 2 16:17:17
gimp         goncalo    4870      0         0           0           0      Dec 2 16:17:18
gdm-x-session goncalo    1735      0         0           0           0      Dec 2 16:10:42
dbus-daemon  goncalo    1878      0         0           0           0      Dec 2 16:10:43
dbus-daemon  goncalo    1653      0         0           0           0      Dec 2 16:10:42
code      goncalo    16566      0         0           0           0      Dec 2 16:28:40
code      goncalo    16513      0        31         0        15.50      Dec 2 16:28:40
code      goncalo    16458      0         0           0           0      Dec 2 16:28:39
```



Teste para -u:

12. Testar para -u sem argumentos, sem o tempo certo, sem o formato certo, e com tudo nas conformidades;
13. Testar para -u com outros filtros no formato certo (-w);

12.

```

goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -u
./rwstat.sh: option requires an argument -- u
COMANDO INVALIDO
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -u 2
formato incorreto!!
-u [String nome.utilizador], último argumento [int TEMPO]
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -u "g*"
USER: g*
O tempo tem que ser um inteiro e positivo
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -u "g*" 2
USER: g*

```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
rwstat.sh	goncalo	82406	96571850	49396	48285925.00	24698.00	Dec 2 17:27:47
pulseaudio	goncalo	1647	3229	3701	1614.50	1850.50	Dec 2 16:10:42
brave	goncalo	3079	2700	1985	1350.00	992.50	Dec 2 16:11:08
code	goncalo	16294	1068	1068	534.00	534.00	Dec 2 16:28:37
brave	goncalo	2452	760	856	380.00	428.00	Dec 2 16:10:57
brave	goncalo	3098	623	623	311.50	311.50	Dec 2 16:11:10
brave	goncalo	3105	287	7	143.50	3.50	Dec 2 16:11:10
korg	goncalo	1737	265	10473	132.50	5236.50	Dec 2 16:10:42
spotify	goncalo	4133	98	1384	49.00	692.00	Dec 2 16:13:09
brave	goncalo	2487	48	48	24.00	24.00	Dec 2 16:10:57
brave	goncalo	2488	38	6	19.00	3.00	Dec 2 16:10:57
code	goncalo	16343	29	29	14.50	14.50	Dec 2 16:28:37
code	goncalo	16484	13	13	6.50	6.50	Dec 2 16:28:40
code	goncalo	16483	8	8	4.00	4.00	Dec 2 16:28:40
spotify	goncalo	4266	5	16929	2.50	8464.50	Dec 2 16:13:12
brave	goncalo	3007	2	2	1.00	1.00	Dec 2 16:11:05
xdg-permission-	goncalo	1997	0	0	0	0	Dec 2 16:10:44
xdg-document-po	goncalo	2236	0	0	0	0	Dec 2 16:10:44
xdg-desktop-por	goncalo	2370	0	0	0	0	Dec 2 16:10:45
xdg-desktop-por	goncalo	2366	0	0	0	0	Dec 2 16:10:45
update-notifier	goncalo	3588	0	0	0	0	Dec 2 16:11:44
spotify	goncalo	4294	0	0	0	0	Dec 2 16:13:12
spotify	goncalo	4261	0	0	0	0	Dec 2 16:13:12
spotify	goncalo	4236	0	0	0	0	Dec 2 16:13:10
spotify	goncalo	4221	0	0	0	0	Dec 2 16:13:10
spotify	goncalo	4220	0	0	0	0	Dec 2 16:13:10
snap-store	goncalo	2194	0	0	0	0	Dec 2 16:10:44
script-fu	goncalo	4968	0	0	0	0	Dec 2 16:17:20
nautilus	goncalo	49595	0	0	0	0	Dec 2 17:01:39
ibus-portal	goncalo	1980	0	0	0	0	Dec 2 16:10:43
ibus-engine-stm	goncalo	2179	0	0	0	0	Dec 2 16:10:44
ibus-daemon	goncalo	1970	0	0	0	0	Dec 2 16:10:43
gvfs-mtp-volume	goncalo	1722	0	0	0	0	Dec 2 16:10:42
gvfs-gphoto2-vo	goncalo	1698	0	0	0	0	Dec 2 16:10:42
gvfsd-trash	goncalo	2059	0	0	0	0	Dec 2 16:10:44
gvfsd-network	goncalo	4710	0	0	0	0	Dec 2 16:17:07
gvfsd-dnssd	goncalo	4724	0	0	0	0	Dec 2 16:17:08
jsd-usb-protect	goncalo	2105	0	0	0	0	Dec 2 16:10:44
jsd-screensaver	goncalo	2092	0	0	0	0	Dec 2 16:10:44
jsd-print-notif	goncalo	2085	0	0	0	0	Dec 2 16:10:44
jsd-printer	goncalo	2206	0	0	0	0	Dec 2 16:10:44
jsd-power	goncalo	2079	0	0	0	0	Dec 2 16:10:44
jsd-housekeepin	goncalo	2073	0	0	0	0	Dec 2 16:10:44

13.

```

goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -r -u "x*" 2
USER: x*
COMM          USER          PID          READB          WRITEB          RATER          RATEW          DATE
^[[A^[[A^[[B^Z
[i]+ Stopped          ./rwstat.sh -r -u "x*" 2
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -w -u "g*" 2
USER: g*
COMM          USER          PID          READB          WRITEB          RATER          RATEW          DATE
brave          goncalo          2488          4785680          4769919          2392840.00          2384959.50          Dec 2 16:10:57
rwstat.sh      goncalo          92980          100812153          50654          50406076.50          25327.00          Dec 2 17:32:05
Xorg           goncalo          1737          265          10001          132.50          5000.50          Dec 2 16:10:42
pulseaudio     goncalo          1647          3075          3547          1537.50          1773.50          Dec 2 16:10:42
brave          goncalo          3079          2616          1907          1308.00          953.50          Dec 2 16:11:08
brave          goncalo          3098          666          602          333.00          301.00          Dec 2 16:11:10
spotify        goncalo          4133          52          469          26.00          234.50          Dec 2 16:13:09
code           goncalo          16294          354          354          177.00          177.00          Dec 2 16:28:37
brave          goncalo          2452          335          265          167.50          132.50          Dec 2 16:10:57
code           goncalo          16513          0          31          0          15.50          Dec 2 16:28:40
code           goncalo          16343          29          29          14.50          14.50          Dec 2 16:28:37
brave          goncalo          3007          4193          17          2096.50          8.50          Dec 2 16:11:05
code           goncalo          16484          13          13          6.50          6.50          Dec 2 16:28:40
code           goncalo          16483          0          8          4.00          4.00          Dec 2 16:28:40
brave          goncalo          3105          44          4          22.00          2.00          Dec 2 16:11:10
xdg-permission- goncalo          1997          0          0          0          0          Dec 2 16:10:44
xdg-document-po goncalo          2236          0          0          0          0          Dec 2 16:10:44
xdg-desktop-por goncalo          2370          0          0          0          0          Dec 2 16:10:45
xdg-desktop-por goncalo          2366          0          0          0          0          Dec 2 16:10:45
update-notifier goncalo          3588          0          0          0          0          Dec 2 16:11:44
spotify        goncalo          4294          0          0          0          0          Dec 2 16:13:12
spotify        goncalo          4266          0          0          0          0          Dec 2 16:13:12
spotify        goncalo          4261          0          0          0          0          Dec 2 16:13:12
spotify        goncalo          4236          0          0          0          0          Dec 2 16:13:10
spotify        goncalo          4221          0          0          0          0          Dec 2 16:13:10
spotify        goncalo          4220          0          0          0          0          Dec 2 16:13:10
snap-store     goncalo          2194          0          0          0          0          Dec 2 16:10:44
script-fu      goncalo          4968          0          0          0          0          Dec 2 16:17:20
rwstat.sh      goncalo          90523          0          0          0          0          Dec 2 17:31:48
ps             goncalo          92964          0          0          0          0          Dec 2 17:31:52
nautilus       goncalo          49595          0          0          0          0          Dec 2 17:01:39
ibus-portal    goncalo          1980          0          0          0          0          Dec 2 16:10:43
ibus-engine-sin goncalo          2179          0          0          0          0          Dec 2 16:10:44
ibus-daemon    goncalo          1970          0          0          0          0          Dec 2 16:10:43
gvfs-mtp-volume goncalo          1722          0          0          0          0          Dec 2 16:10:42
gvfs-gphoto2-vo goncalo          1698          0          0          0          0          Dec 2 16:10:42
gvfsd-trash    goncalo          2059          0          0          0          0          Dec 2 16:10:44

```

Teste para -m e -M:

14. Testar para -m e -M sem o argumento seguinte, com tipo diferente do pedido e por último um teste com -m com o formato certo
15. Testar para -M com o formato certo;

14.

```

goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -m 2
Formato incorreto!! -m [int PID_minimo], último argumento [int TEMPO]
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -M 2
Formato incorreto!! -M [int PID_maximo], último argumento [int TEMPO]
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -m "sport" 2
Formato correto!! -m [int PID_minimo], último argumento [int TEMPO]
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -M "sport" 2
Formato incorreto!! -M [int PID_maximo], último argumento [int TEMPO]
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -m 4 2
PID minimo: 4

```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
rwstat.sh	goncalo	102969	92869020	48051	46434510.00	24025.50	Dec 2 21:04:12
Xorg	goncalo	1737	233	2377	116.50	1188.50	Dec 2 16:10:42
brave	goncalo	2488	32	0	16.00	0	Dec 2 16:10:57
code	goncalo	16343	29	29	14.50	14.50	Dec 2 16:28:37
spotify	goncalo	4133	20	20	10.00	10.00	Dec 2 16:13:09
brave	goncalo	2452	17	113	8.50	56.50	Dec 2 16:10:57
code	goncalo	16484	13	13	6.50	6.50	Dec 2 16:28:40
code	goncalo	16483	8	8	4.00	4.00	Dec 2 16:28:40
spotify	goncalo	4236	5	0	2.50	0	Dec 2 16:13:10
xdg-permission-	goncalo	1997	0	0	0	0	Dec 2 16:10:44
xdg-document-po	goncalo	2236	0	0	0	0	Dec 2 16:10:44
xdg-desktop-por	goncalo	2370	0	0	0	0	Dec 2 16:10:45
xdg-desktop-por	goncalo	2366	0	0	0	0	Dec 2 16:10:45
update-notifier	goncalo	3588	0	0	0	0	Dec 2 16:11:44
spotify	goncalo	4294	0	0	0	0	Dec 2 16:13:12
spotify	goncalo	4266	0	0	0	0	Dec 2 16:13:12
spotify	goncalo	4261	0	0	0	0	Dec 2 16:13:12
spotify	goncalo	4221	0	0	0	0	Dec 2 16:13:10
spotify	goncalo	4220	0	0	0	0	Dec 2 16:13:10
snap-store	goncalo	2194	0	0	0	0	Dec 2 16:10:44
script-fu	goncalo	4968	0	0	0	0	Dec 2 16:17:20
pulseaudio	goncalo	1647	0	0	0	0	Dec 2 16:10:42
nautilus	goncalo	49595	0	0	0	0	Dec 2 17:01:39
ibus-portal	goncalo	1980	0	0	0	0	Dec 2 16:10:43
ibus-engine-slm	goncalo	2179	0	0	0	0	Dec 2 16:10:44
ibus-daemon	goncalo	1970	0	0	0	0	Dec 2 16:10:43
gvfs-mtp-volume	goncalo	1722	0	0	0	0	Dec 2 16:10:42
gvfs-gphoto2-vo	goncalo	1698	0	0	0	0	Dec 2 16:10:42
gvfsd-trash	goncalo	2059	0	0	0	0	Dec 2 16:10:44
gvfsd-network	goncalo	4710	0	0	0	0	Dec 2 16:17:07
gvfsd-dnssd	goncalo	4724	0	0	0	0	Dec 2 16:17:08
gsd-usb-protect	goncalo	2105	0	0	0	0	Dec 2 16:10:44
gsd-screensaver	goncalo	2092	0	0	0	0	Dec 2 16:10:44
gsd-print-notif	goncalo	2085	0	0	0	0	Dec 2 16:10:44
gsd-printer	goncalo	2206	0	0	0	0	Dec 2 16:10:44
gsd-power	goncalo	2079	0	0	0	0	Dec 2 16:10:44
gsd-housekeepin	goncalo	2073	0	0	0	0	Dec 2 16:10:44
gnome-calendar	goncalo	4746	0	0	0	0	Dec 2 16:17:17
gimp	goncalo	4870	0	0	0	0	Dec 2 16:17:18
gdm-x-session	goncalo	1735	0	0	0	0	Dec 2 16:10:42
dbus-daemon	goncalo	1878	0	0	0	0	Dec 2 16:10:43
dbus-daemon	goncalo	1653	0	0	0	0	Dec 2 16:10:42
code	goncalo	16566	0	0	0	0	Dec 2 16:28:40

15.

```

goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -M 1892 2
PID maximo: 1892
COMM      USER      PID      READB     WRITEB     RATER     RATEW     DATE
Xorg      goncalo    1737      192       2680       96.00      1340.00    Dec 2 16:10:42
pulseaudio goncalo    1647      0          0          0          0          Dec 2 16:10:42
gvfs-mtp-volume goncalo    1722      0          0          0          0          Dec 2 16:10:42
gvfs-gphoto2-vo goncalo    1698      0          0          0          0          Dec 2 16:10:42
gdm-x-session goncalo    1735      0          0          0          0          Dec 2 16:10:42
dbus-daemon goncalo    1878      0          0          0          0          Dec 2 16:10:43
dbus-daemon goncalo    1653      0          0          0          0          Dec 2 16:10:42
at-spi-bus-laun goncalo    1873      0          0          0          0          Dec 2 16:10:43
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -M 1892 -m 1700 2
PID maximo: 1892
PID minimo: 1700
COMM      USER      PID      READB     WRITEB     RATER     RATEW     DATE
Xorg      goncalo    1737      425       5357       212.50     2678.50    Dec 2 16:10:42
gvfs-mtp-volume goncalo    1722      0          0          0          0          Dec 2 16:10:42
gdm-x-session goncalo    1735      0          0          0          0          Dec 2 16:10:42
dbus-daemon goncalo    1878      0          0          0          0          Dec 2 16:10:42
at-spi-bus-laun goncalo    1873      0          0          0          0          Dec 2 16:10:43
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -M 1892 -m 1700 -r 2
PID maximo: 1892
PID minimo: 1700
COMM      USER      PID      READB     WRITEB     RATER     RATEW     DATE
at-spi-bus-laun goncalo    1873      0          0          0          0          Dec 2 16:10:43
dbus-daemon goncalo    1878      0          0          0          0          Dec 2 16:10:43
gdm-x-session goncalo    1735      0          0          0          0          Dec 2 16:10:42
gvfs-mtp-volume goncalo    1722      0          0          0          0          Dec 2 16:10:42
Xorg      goncalo    1737      352       5172       176.00     2586.00    Dec 2 16:10:42
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -M 1892 -m 1700 -w 2
PID maximo: 1892
PID minimo: 1700
COMM      USER      PID      READB     WRITEB     RATER     RATEW     DATE
Xorg      goncalo    1737      11286     21058     5643.00     10529.00    Dec 2 16:10:42
gvfs-mtp-volume goncalo    1722      0          0          0          0          Dec 2 16:10:42
gdm-x-session goncalo    1735      0          0          0          0          Dec 2 16:10:42
dbus-daemon goncalo    1878      0          0          0          0          Dec 2 16:10:43
at-spi-bus-laun goncalo    1873      0          0          0          0          Dec 2 16:10:43
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ 

```

Teste para -p:

16. Testar para -p sem argumentos, sem o argumento seguinte, com o argumento seguinte no formato errado e para -p no formato certo;
17. Testar para -p com outros filtros no formato certo;
18. Testar input com alguns valores sem utilidade;

16.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -p
./rwstat.sh: option requires an argument -- p
COMANDO INVALIDO
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -p 2
Formato incorreto!! -p [int Nr_PIDs], último argumento [int TEMPO]
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -p "randomstring"
Formato incorreto!! -p [int Nr_PIDs], último argumento [int TEMPO]
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -p r
Formato incorreto!! -p [int Nr_PIDs], último argumento [int TEMPO]
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -p 2 2
nr PIDs: 2
COMM          USER          PID          READB        WRITEB        RATER        RATEW        DATE
rwstat.sh     goncalo        122677       92893629     48130        46446814.50   24065.00     Dec 2 21:22:41
Xorg          goncalo        1737        15195       27932        7597.50       13966.00     Dec 2 16:10:42
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -r -p 2 2
nr PIDs: 2
COMM          USER          PID          READB        WRITEB        RATER        RATEW        DATE
at-spi2-registr goncalo        1993         0           0           0           0           Dec 2 16:10:43
at-spi-bus-laun goncalo        1873         0           0           0           0           Dec 2 16:10:43
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -r -p 0 2
nr PIDs: 0
COMM          USER          PID          READB        WRITEB        RATER        RATEW        DATE
```

17.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -r -p 12 -c "d*" 2
nr PIDs: 12
nome do PID: d*
COMM          USER          PID          READB        WRITEB        RATER        RATEW        DATE
0 numero de processos que quer mostrar é maior que o numero de processos existentes
```

18.

```
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -w -r -p 2 -c "d*" -e "Dec 2 20:20" 2
nr PIDs: 2
nome do PID: d*
data maxima (tudo junto): 12022020
COMM          USER          PID          READB        WRITEB        RATER        RATEW        DATE
dbus-daemon   goncalo        1653         0           0           0           0           Dec 2 16:10:44
dbus-daemon   goncalo        1878         0           0           0           0           Dec 2 16:10:45
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -w -r -p 2 -c "d*" -e "Dec 2 20:20" "sdadsa" 2
Argumento de índice: 8 e de valor: sdadsa não tem utilidade
goncalo@goncalomf:~/Desktop/S0/Projecto C2$ ./rwstat.sh -w -r -p 2 3 -c "d*" -e "Dec 2 20:20" 2
Argumento de índice: 4 e de valor: 3 não tem utilidade
```

# Conclusão

Depois de termos explicado o código feito e testado de acordo com o que era pedido no enunciado, podemos concluir que concluímos o trabalho de maneira minimamente consistente e coerente. Este trabalho serviu para consolidar os nossos conhecimentos na linguagem de bash e também serviu para aprendermos mais sobre todos os processos que correm no nosso computador diariamente. Aprendemos também a ir buscar dados relativos a cada um desses processos e por final, consideramos que este trabalho aumentou o nosso interesse sobre a matéria devido a todo o desafio que se apresentou durante os dias de resolução.

Sendo assim, podemos terminar este trabalho concluindo que todo ele apresentou uma aventura para nós, que recendo-a de maneira saudável e desafiadora, terminados de maneira satisfatória e que sem dúvida ajudou a consolidar todos os nossos conhecimentos acerca de bash e mais importante de tudo, sobre a disciplina de sistemas operativos.