

RELATÓRIO - SISTEMAS OPERATIVOS

Monitorização de I/O de processos em bash



universidade
de aveiro

José Gonçalves – NMec 84967
Rui Coelho – NMec 86182

Índice

Abordagem à monitorização de I/O de processos em bash	2
Testes de validação	4

Abordagem à monitorização de I/O de processos em bash

Neste trabalho foi proposto criar um script que permita monitorização de I/O de processos em bash que estão a correr no sistema.

Para resolver este problema optamos por criar um diretório temporário em que armazenamos os dados dos processos estruturados como a figura 1.

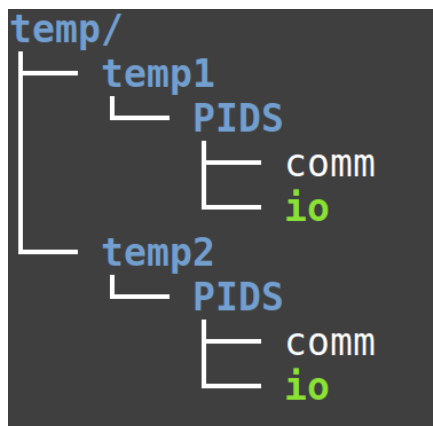


Figura 1 - Esquema representativo da estruturação do diretório temporário

Caso não consigamos ler o ficheiro I/O por falta de permissões de acesso, este não será copiado, apenas será copiado o comm para um posterior tratamento de dados. Assim, copiando os ficheiros para um diretório temporário teremos uma leitura mais estável. No final, de modo a que o utilizador não perca espaço no seu disco apagamos o diretório temp criado para esta monitorização. Caso o utilizador pare o script com CTRL+C os ficheiros também serão apagados, essa situação foi precavida com o comando trap que se encontra no código.

A cópia dos ficheiros retira algumas informações sobre estes processos como por exemplo quem era o "dono" desse processo e, portanto, temos de guardar essa informação em algum sítio antes de esta ser extraída, designamos o ficheiro comm para isso, deste modo, o ficheiro comm quando é extraído não é exatamente igual ao que se encontra no processo, tem todas as informações que se encontravam neste mais o "dono" do processo e, para além disso, terá também a data de quando o processo começou uma vez que esta, quando copiada para um diretório temporário, será modificada no diretório temporário apesar de no original se encontrar exatamente igual.

Para a apresentação dos dados na tabela requerida fazemos a abertura do ficheiro I/O, se ele não existir apresentamos -1 nos parâmetros da tabela, se ele existir, abrimos, retiramos os seus valores de forma a estes serem apresentados na tabela. Na tabela encontram-se dois valores que não são possíveis de extrair do I/O uma vez que não constam lá, são estes o RATER e RATEW. Nos casos referidos anteriormente me que não aparecem os valores estes terão de ser calculados por nós, o cálculo destes é feito através das duas leituras, isto é, vamos ao temp1 (subdiretório de temp/ no qual se encontram os ficheiros da primeira leitura de processos) e pegamos nos valores do rchar e wchar e vamos ao temp2 (subdiretório de temp/ no qual se encontram os ficheiros da segunda leitura de processos) e pegamos também nos valores do rchar e wchar.

Após isso fazemos os cálculos:

$$RATER = \frac{rchar1 - rchar2}{tempo}$$

Sendo rchar1 o rchar extraído do I/O do temp1/ e o rchar2 extraído do I/O do temp2/. O tempo é o tempo de sleep que é definido pelo utilizador.

$$RATEW = \frac{wchar1 - wchar2}{tempo}$$

Sendo wchar1 o wchar extraído do I/O do temp1/ e o wchar2 extraído do I/O do temp2/. O tempo é o tempo de sleep que é definido pelo utilizador (último argumento a ser introduzido). Sendo que esta divisão foi precavida devido a ser uma divisão de inteiros, para isso usamos o comando bc que permite fazer uma floating-point division.

Para facilitar a seleção de argumentos e verificação desta optamos pela utilização do comando getopt o qual permite selecionar as opções a ser usadas, no caso, -c, -s, -e, -u, -p, -r, -w, -t e, nos caso destas terem mais argumentos associados é possível obtê-los de forma fácil, facilitando assim o tratamento de dados.

Apesar do getopt fazer parte de verificação é necessário fazer mais algumas, isto é, assegurar que o último argumento é um número, assegurar que este último argumento é número positivo uma vez que é o tempo e, como sabemos, o tempo nunca pode ser negativo.

Nos parâmetros das opções usadas também são necessárias algumas verificações, no caso de ser -p o valor deste não pode ser negativo, em termos das verificações da data a data máxima tem de ser maior ou igual que a data mínima.

Para além do que era pedido para implementar decidimos colocar um menu de ajuda que pode ser obtido introduzindo -help, -help, -h, quando introduzido pelos utilizadores um destes comandos o menu de ajuda é impresso no terminal, este menu de ajuda é também impresso quando não são introduzidos argumentos e é só invocado ./ioproc.sh.

Testes de validação

Para validar o trabalho efetuado fizemos várias verificações entre as quais as que eram pedidas no projeto como se pode ver pelas figuras. (Por impossibilidade de usar os computadores da universidade os prints foram feitos através dos nossos computadores, contudo, foi testado nos computadores da universidade e correu consoante os conformes)

```
rui-coelho-laptop@rui-coelho-laptop ~/Desktop/Projeto_S0 $ ./ioproc.sh 10
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
ioproc.sh	rui-coelho-laptop	21955	0	327680	1001627	3871	Nov 21 2:35
sublime text	rui-coelho-laptop	14779	4096	4214784	50760	5561	Nov 21 1:49
cinnamon	rui-coelho-laptop	2542	368881664	766636032	1213	383	Nov 20 22:19
dbus-daemon	rui-coelho-laptop	2262	0	0	48	0	Nov 20 22:19

Figura 2 - Primeiro teste de validação

```
rui-coelho-laptop@rui-coelho-laptop ~/Desktop/Projeto_S0 $ ./ioproc.sh -c "d.*" 10
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
dconf-service	rui-coelho-laptop	2383	4096	3047424	0	0	Nov 20 22:19
dbus-launch	rui-coelho-laptop	2231	0	0	0	0	Nov 20 22:19
dbus-daemon	rui-coelho-laptop	2262	0	0	0	0	Nov 20 22:19
dbus-daemon	rui-coelho-laptop	2232	0	0	0	0	Nov 20 22:19
dnsmasq	root	1366	-1	-1	-1	-1	Nov 20 22:19
dhclient	root	10329	-1	-1	-1	-1	Nov 20 23:33
devfreq_wq	root	45	-1	-1	-1	-1	Nov 20 22:19
deferwq	root	128	-1	-1	-1	-1	Nov 20 22:19
dbus-launch	root	23223	-1	-1	-1	-1	Nov 20 23:14
dbus-daemon	root	878	-1	-1	-1	-1	Nov 20 22:19
dbus-daemon	root	23225	-1	-1	-1	-1	Nov 20 23:14

Figura 3 - Segundo teste de validação

```
rui-coelho-laptop@rui-coelho-laptop ~/Desktop/Projeto_S0 $ ./ioproc.sh -u "root" 10
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
Xorg	root	1152	-1	-1	-1	-1	Nov 20 22:19
writeback	root	35	-1	-1	-1	-1	Nov 20 22:19
wpa_supplicant	root	1215	-1	-1	-1	-1	Nov 20 22:19
winbindd	root	1779	-1	-1	-1	-1	Nov 20 22:19
winbindd	root	1778	-1	-1	-1	-1	Nov 20 22:19
winbindd	root	1774	-1	-1	-1	-1	Nov 20 22:19
winbindd	root	1736	-1	-1	-1	-1	Nov 20 22:19
watchdogd	root	46	-1	-1	-1	-1	Nov 20 22:19
watchdog/3	root	26	-1	-1	-1	-1	Nov 20 22:19

Figura 4 - Terceiro teste de validação

```
rui-coelho-laptop@rui-coelho-laptop ~/Desktop/Projeto_S0 $ ./ioproc.sh -s "Nov 20 23:00" -e "Nov 21 3:00" 10
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
ioproc.sh	rui-coelho-laptop	9950	0	1228800	984295	3795	Nov 21 2:57
nemo	rui-coelho-laptop	977	7081472	1282048	0	0	Nov 20 23:18
gnome-terminal	rui-coelho-laptop	24396	0	0	0	379	Nov 21 2:52
cp	rui-coelho-laptop	thread-self	0	0	0	0	Nov 21 2:57
cp	rui-coelho-laptop	self	0	0	0	0	Nov 21 2:57
bash	rui-coelho-laptop	24400	0	13070336	0	0	Nov 21 2:52

Figura 5 - Quarto teste de validação

```
rui-coelho-laptop@rui-coelho-laptop ~/Desktop/Projeto_S0 $ ./ioproc.sh -t -c "d.*" 10
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
dconf-service	rui-coelho-laptop	2383	4096	3047424	0	0	Nov 20 22:19
dbus-launch	rui-coelho-laptop	2231	0	0	0	0	Nov 20 22:19
dbus-daemon	rui-coelho-laptop	2262	0	0	0	0	Nov 20 22:19
dbus-daemon	rui-coelho-laptop	2232	0	0	0	0	Nov 20 22:19
dnsmasq	root	1366	-1	-1	-1	-1	Nov 20 22:19
dhclient	root	10329	-1	-1	-1	-1	Nov 20 23:33
devfreq_wq	root	45	-1	-1	-1	-1	Nov 20 22:19
deferwq	root	128	-1	-1	-1	-1	Nov 20 22:19
dbus-launch	root	23223	-1	-1	-1	-1	Nov 20 23:14
dbus-daemon	root	878	-1	-1	-1	-1	Nov 20 22:19
dbus-daemon	root	23225	-1	-1	-1	-1	Nov 20 23:14

Figura 6 - Quinto teste de validação

```
rui-coelho-laptop@rui-coelho-laptop ~/Desktop/Projeto_S0 $ ./ioproc.sh -w -c "d.*" 10
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
dconf-service	rui-coelho-laptop	2383	4096	3047424	0	0	Nov 20 22:19
dbus-launch	rui-coelho-laptop	2231	0	0	0	0	Nov 20 22:19
dbus-daemon	rui-coelho-laptop	2262	0	0	0	0	Nov 20 22:19
dbus-daemon	rui-coelho-laptop	2232	0	0	0	0	Nov 20 22:19
dnsmasq	root	1366	-1	-1	-1	-1	Nov 20 22:19
dhclient	root	10329	-1	-1	-1	-1	Nov 20 23:33
devfreq_wq	root	45	-1	-1	-1	-1	Nov 20 22:19
deferwq	root	128	-1	-1	-1	-1	Nov 20 22:19
dbus-launch	root	23223	-1	-1	-1	-1	Nov 20 23:14
dbus-daemon	root	878	-1	-1	-1	-1	Nov 20 22:19
dbus-daemon	root	23225	-1	-1	-1	-1	Nov 20 23:14

Figura 7 - Sexto teste de validação

```
rui-coelho-laptop@rui-coelho-laptop ~/Desktop/Projeto_S0 $ ./ioproc.sh -t -w -r -c "d.*" 10
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
dbus-daemon	root	23225	-1	-1	-1	-1	Nov 20 23:14
dbus-daemon	root	878	-1	-1	-1	-1	Nov 20 22:19
dbus-launch	root	23223	-1	-1	-1	-1	Nov 20 23:14
deferwq	root	128	-1	-1	-1	-1	Nov 20 22:19
devfreq_wq	root	45	-1	-1	-1	-1	Nov 20 22:19
dhclient	root	10329	-1	-1	-1	-1	Nov 20 23:33
dnsmasq	root	1366	-1	-1	-1	-1	Nov 20 22:19
dbus-daemon	rui-coelho-laptop	2232	0	0	0	0	Nov 20 22:19
dbus-daemon	rui-coelho-laptop	2262	0	0	0	0	Nov 20 22:19
dbus-launch	rui-coelho-laptop	2231	0	0	0	0	Nov 20 22:19
dconf-service	rui-coelho-laptop	2383	4096	3047424	0	0	Nov 20 22:19

Figura 8 - Sétimo teste de validação

```
rui-coelho-laptop@rui-coelho-laptop ~/Desktop/Projeto_S0 $ ./ioproc.sh -t -w -r -c "d.*" -p 1 10
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
dconf-service	rui-coelho-laptop	2383	4096	3047424	0	0	Nov 20 22:19

Figura 9 - Oitavo teste de validação

Para além dos testes que vêm no enunciado, fizemos mais alguns testes que validam a introdução de argumentos anteriormente referida.

```
rui-coelho-laptop@rui-coelho-laptop ~/Desktop/Project_S0 $ ./ioproc.sh -s "ola" 1
Data inválida
```

Figura 10 - Validação da data introduzida

```
rui-coelho-laptop@rui-coelho-laptop ~/Desktop/Project_S0 $ ./ioproc.sh -s "ola"
Último argumento tem de ser inteiro positivo
```

Figura 11 - Validação do último argumento para inteiro positivo

```
rui-coelho-laptop@rui-coelho-laptop ~/Desktop/Project_S0 $ ./ioproc.sh -s "Nov 21 23:00" -e "Nov 20 23:00" 1
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
------	------	-----	-------	--------	-------	-------	------

Figura 12 - Quando a data mínima é superior à data máxima nada é impresso

```
rui-coelho-laptop@rui-coelho-laptop ~/Desktop/Project_S0 $ ./ioproc.sh -p -1 10
Tem de ser um inteiro positivo ou zero
```

Figura 13 - Validação do argumento de -p (inteiro positivo)

```
rui-coelho-laptop@rui-coelho-laptop ~/Desktop/Project_S0 $ ./ioproc.sh
-c Expressão regular
-s Data mínima
-e Data máxima
-u Seleção de processos através do nome do utilizador
-p Número de processos
-r Reverso
-w Sort and write values
-t Sort on total values
0 último argumento deverá ser o tempo
```

Figura 14- Mostra o menu de ajuda quando não são introduzidos argumentos