

Trabalho 1

Monitorização de I/O de processos em **bash**

Guião

O objectivo do trabalho é o desenvolvimento de um *script* em **bash** para monitorizar a quantidade de I/O que os processos estão a efetuar. Esta ferramenta permite visualizar o número total de bytes de I/O que um processo leu/escreveu e também a taxa de leitura/escrita correspondente aos últimos *s* segundos para uma selecção de processos (o valor de *s* é passado como parametro). Permite também comparar os dados obtidos em períodos distintos.

O script **ioproc.sh** permite a visualização do número de total de bytes de I/O (linhas **rchar** e **wchar** de **/proc/[pid]/io**) e da taxa de leitura/escrita (em bytes por segundo) dos processos seleccionados nos últimos *s* segundos (calculadas a partir de 2 leituras de **/proc/[pid]/io** com intervalo de *s* segundos). Este script tem um parâmetro obrigatório que é o número de segundos que serão usados para calcular as taxas de I/O. A selecção dos processos a visualizar pode ser realizada através de uma expressão regular que é verificada com o comando (tal como aparece em **/proc/[pid]/comm**) associado (opção **-c**), ou através da definição de um período temporal para o início do processo. A especificação do período temporal faz-se através da data mínima (opção **-s**) e data máxima (opção **-e**) para o início do processo. A selecção dos processos pode ainda ser realizada através do nome do utilizador (opção **-u**). A visualização está formatada como uma tabela, com um cabeçalho, aparecendo os processos por ordem inversa da taxa de leitura. O número de processos a visualizar é controlado pela opção **-p**. Existem opções para alterar a ordenação da tabela (**-r** – *reverse*, **-w** – *sort on write values* e **-t** – *sort on total values*). Seguem-se exemplos do que pode aparecer na consola durante a execução deste *script*:

```
$ ./ioproc.sh 10
```

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

```
$ ./ioproc.sh -c "d.*" 10
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
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

```
$ ./ioproc.sh -u sop0100 10
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
dropbox	sop0100	2636	918784	1356760	40022.70	31114.50	Sep 19 08:49

```
$ ./ioproc.sh -s "Sep 10 10:00" -e "Sep 20 18:00" 10
```

COMM	USER	PID	READB	WRITEB	RATER	RATEW	DATE
bash	nlau	10174	15324690	58597	3770687.30	14239.00	Sep 12 11:45
dropbox	sop0100	2636	918784	1356760	40022.70	31114.50	Sep 19 08:49

```

$ ./ioproc.sh -t -c "d.*" 10
COMM      USER      PID      READB      WRITEB      RATER      RATEW      DATE
dice      sop0200    5036     2914000     1356       4022.70     5114.70 Sep 23 18:14
dropbox   sop0100    2636     918784     1356760    40022.70    31114.50 Sep 19 08:49

$ ./ioproc.sh -w -c "d.*" 10
COMM      USER      PID      READB      WRITEB      RATER      RATEW      DATE
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

$ ./ioproc.sh -t -w -r -c "d.*" 10
COMM      USER      PID      READB      WRITEB      RATER      RATEW      DATE
dice      sop0200    5036     2914000     1356       4022.70     5114.70 Sep 23 18:14
dropbox   sop0100    2636     918784     1356760    40022.70    31114.50 Sep 19 08:49

$ ./ioproc.sh -t -w -r -c "d.*" -p 1 10
COMM      USER      PID      READB      WRITEB      RATER      RATEW      DATE
dice      sop0200    5036     2914000     1356       4022.70     5114.70 Sep 23 18:14

```

A estrutura da linha de comando dos *scripts* deve ser sempre validada, garantindo assim que os parâmetros que foram usados estão de acordo com o esperado.

A execução do trabalho poderá ser suportada através de um repositório GIT ou SVN a criar na plataforma **code.ua.pt**.

O trabalho será realizado em grupos de 2 alunos. Durante a execução do trabalho deve ser respeitado um exigente código de ética que impede o plágio, sob qualquer forma, bem como a execução do trabalho por elementos externos ao grupo ou a partilha de código entre grupos distintos.

A entrega do trabalho será realizada através do **elearning.ua.pt** e deverá incluir o código fonte da solução encontrada e um relatório que descreve qual a abordagem usada para resolver o problema e os testes realizados para validar a solução.

Dicas: alguns comandos que poderão ser úteis para este trabalho são **awk**, **bc**, **cat**, **cut**, **date**, **getopts**, **grep**, **head**, **ls**, **printf**, **sleep**, **sort**

Data de entrega do trabalho: 20 de novembro de 2017