

1º Teste Prático – Enunciado A

2018.10.31

Prova com Consulta

Duração: 120 minutos

Nome Completo: _____

N.º de Estudante: _____ Regime: [] Diurno [] Pós-laboral

IMPORTANTE

É expressamente proibido o recurso à Internet durante a prova. Qualquer utilização não autorizada da Internet leva à anulação da prova e ao reportar da situação às autoridades competentes. O mesmo sucede com outros tipos de tentativa de fraude.

• Antes de iniciar a prova:

- Execute os seguintes comandos:

```
cd; mkdir -p ~/Prova01/R_NUMERO/
```

(em que **R** deve ser substituído pela letra **D** se for do regime diurno e **N** se for aluno do regime pós-laboral e **NUMERO** deve ser substituído pelo seu número ESTG);

- Para garantir que o seu diretório de trabalho seja o correto, faça:

```
cd ~/Prova01/R_NUMERO/
```

• Após ter terminado a prova:

- Deverá proceder à criação de um arquivo TAR, fazendo uso do seguinte comando:

```
cd ~/Prova01/R_NUMERO/; tar cvf Prova01_YYYYMMDD_R_NUMERO.tar *
```

(em que YYYYMMDD corresponde à data corrente (e.g., 20181031) e R_NUMERO obedece ao formato acima indicado);

- Verifique que o arquivo “.tar” que criou não está vazio, através da execução de:

```
tar tvf Prova01_YYYYMMDD_R_NUMERO.tar
```

- Entregue o arquivo “.tar” através da plataforma moodle, no espaço reservado para o efeito. Em caso de dúvidas, pergunte ao professor;
- Informe o professor para este validar a receção dos seus ficheiros.

Pergunta 1 [10 valores]

(Escreva as suas respostas a esta pergunta no diretório “~/Prova01/R_NUMERO/Pergunta1”. Deve indicar o seu nome completo e número de estudante IPLeiria no ficheiro **README.txt** a ser criado no diretório)

NOTA 1: não é permitida a chamada a comandos externos através da função **system** ou de outra com funcionalidade similar.

NOTA 2: a solução deve ser implementada com recurso aos ficheiros existentes no arquivo **EmptyProject-Templatev3.05.zip**.

NOTA 3: código entregue que **não compile** leva à atribuição da classificação de **0 (zero) valores** à resposta.

Recorrendo à linguagem C, elabore a aplicação **catch_all_signals**. A aplicação deve capturar os sinais que lhe são enviados, consoante a configuração que é indicada através dos parâmetros da linha de comandos. Sempre que recebe um sinal para o qual está configurada a captura, a aplicação deve imprimir uma mensagem no canal de saída padrão e prosseguir a execução. A única exceção prende-se com o sinal SIGINT que quando recebido deve sempre levar ao término da aplicação.

A aplicação deve ainda assinalar os *signals* que, por imposição do sistema, não consegue capturar. Sempre que referenciar um *signal*, a aplicação **catch_all_signals** deve apresentar simultaneamente o nome do *signal* e o respetivo identificador numérico. O nome do *signal* pode ser obtido através da função `strsignal` (man `strsignal`).

Os parâmetros da linha de comandos determinam a gama de sinais a serem processados pela aplicação **catch_all_signals** e devem ser processados através do utilitário `gengetopt`. Os parâmetros são:

--min <int>: valor inteiro do **menor** sinal que deve ser capturado pela aplicação `catch_all_signals`, sendo que o valor nunca poderá ser inferior a 1. O parâmetro é opcional. Caso não seja especificado deve ser assumido o valor 1.

--max <int>: valor inteiro do **maior** sinal que deve ser capturado pela aplicação `catch_all_signals`, sendo que o valor nunca poderá ser superior a 64. O parâmetro é opcional. Caso não seja especificado deve ser assumido o valor 64.

Considere os seguintes exemplos que ilustram o funcionamento da aplicação.

Exemplo 1

```
./catch_all_signals
[INFO] Processing signal from [MIN:1, MAX:64]
[WARNING] Can't install handler for signal 'Killed' (9)
[WARNING] Can't install handler for signal 'Stopped (signal)' (19)
[WARNING] Can't install handler for signal 'Unknown signal 32' (32)
[WARNING] Can't install handler for signal 'Unknown signal 33' (33)
[INFO] Terminate: kill -s SIGKILL 8777 | kill -s SIGINT 8777
Waiting for a signal
(processo recebe sinal 2 / SIGINT)
```

```
[PID:8777] Got signal 'INTERUPT' (2) from process '0'
Got SIGINT -- terminating
```

Exemplo 2

```
./catch_all_signals --min 3 --max 20
[INFO] Processing signal from [MIN:3, MAX:20]
[WARNING] Can't install handler for signal 'Killed' (9)
[WARNING] Can't install handler for signal 'Stopped (signal)' (19)
[INFO] Terminate: kill -s SIGKILL 8792 | kill -s SIGINT 8792
Waiting for a signal
```

(*processo recebe sinal 4 / SIGILL*)

```
[PID:8792] Got signal 'Illegal instruction' (4) from process '8726'
Waiting for a signal
```

(*processo recebe sinal 5 / TRACE/BREAKPOINT TRAP*)

```
[PID:8792] Got signal 'Trace/breakpoint trap' (5) from process '8726'
Waiting for a signal
```

Exemplo 3

```
./catch_all_signals --min -2 --max 20
[ERROR] min signal must be >= 1 (-2 given)
```

Exemplo 4

```
./catch_all_signals --min 30 --max 20
[ERROR] min signal must be <= max (20)
```

Pergunta 2 [10 valores]

(Escreva as suas respostas a esta pergunta no diretório "~/Prova01/R_NUMERO/Pergunta2". Deve indicar o seu nome completo e número de estudante IPLeiria no ficheiro **README.txt** a ser criado no diretório)

NOTA 1: não é permitida a chamada a comandos externos através da função **system** ou de outra com funcionalidade similar.

NOTA 2: a solução deve ser implementada com recurso aos ficheiros existentes no arquivo **EmptyProject-Templatev3.05.zip**.

NOTA 3: código entregue que **não compile** leva à atribuição da classificação de **0 (zero) valores** à resposta.

A empresa **Containers4All** decidiu simular o seu sistema de distribuição de contentores que é feito através de camiões porta-contentores. Concretamente, a empresa pretende simular o tempo de entrega de N contentores por M camiões. N e M são parâmetros da simulação passados através da linha de comando. Em cada viagem, um camião apenas transporta um contentor, tendo que percorrer uma distância de X kms, sempre múltipla de 100 e que é gerada aleatoriamente. Esta distância X varia entre 100 e 1000 kms, sendo que para efeitos de simulação, uma distância de 100 kms corresponde a uma espera de 1 segundo e 1000 kms a 10 segundos. Note-se que a distância a percorrer já considera o percurso de ida e volta e que eventuais paragens dos camiões já estão contabilizadas no tempo de percurso. Em termos de implementação, a aplicação deve simular cada camião através de uma *thread*.

A simulação deve terminar quando todos os contentores tiverem sido entregues. No final da simulação, deve ser apresentado:

- Número de contentores entregues;
- Número de quilómetros percorrido por cada camião;
- Número de contentores entregue por cada camião;
- Média de quilómetros percorrido por camião;
- Total de quilómetros percorridos.

Assim, pretende-se que seja desenvolvida, com recurso à linguagem C, a aplicação **containers4All**. A aplicação deve processar os parâmetros da linha de comando através do utilitário **gengetopt**, implementando as seguintes opções:

--trucks/-t <int>: número de camiões de transporte. Parâmetro obrigatório, cujo valor tem que ser sempre maior que 0 (zero).

--containers/-c <int>: número de contentores que têm que ser transportados. Parâmetro obrigatório, cujo valor tem que ser maior que 0 (zero).

Considere os exemplos da página seguinte que ilustram o funcionamento da aplicação.

(continua na página seguinte)

Exemplo 1	Exemplo 2
<pre>./containers4all -t 2 -c 10 [INFO] TruckID '2' starting service [ID:2] Got container 001 (9 left): 500kms [INFO] TruckID '1' starting service [ID:1] Got container 002 (8 left): 500kms [ID:2] Got container 003 (7 left): 400kms [ID:1] Got container 004 (6 left): 400kms [ID:2] Got container 005 (5 left): 500kms [ID:1] Got container 006 (4 left): 200kms [ID:1] Got container 007 (3 left): 100kms [ID:1] Got container 008 (2 left): 200kms [ID:2] Got container 009 (1 left): 100kms [ID:1] Got container 010 (0 left): 400kms ----- Truck 1: 6 container(s); 1800 kms ----- Truck 2: 4 container(s); 1500 kms ----- Total kms: 3300 Avg kms per truck: 1650.0</pre>	<pre>./containers4all -t 3 -c 2 [INFO] TruckID '3' starting service [ID:3] Got container 001 (1 left): 200kms [INFO] TruckID '2' starting service [ID:2] Got container 002 (0 left): 200kms [INFO] TruckID '1' starting service ----- Truck 1: 0 container(s); 0 kms ----- Truck 2: 1 container(s); 200 kms ----- Truck 3: 1 container(s); 200 kms ----- Total kms: 400 Avg kms per truck: 133.3</pre>

Exemplo 3	Exemplo 4
<pre>./containers4all -t 0 -c 20 [ERROR] Invalid number of trucks (0)</pre>	<pre>./containers4all -t 3 -c -3 [ERROR] Invalid number of containers (-3)</pre>