

1º Teste Prático – Enunciado A

2017.11.11 / 09h30'

Prova com Consulta

Duração: 110 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., 20171111) 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: a solução deve ser implementada com recurso aos ficheiros existentes no arquivo **EmptyProject-Templatev3.02.zip**.

NOTA 2: código entregue que **não compile** leva à atribuição da classificação de **0 (zero) valores** à resposta.

Recorrendo à linguagem C, elabore a aplicação **exec_on_signal**. A aplicação deve estar preparada para receber o parâmetro **--execute <comando> / -e <comando>**, em que *comando* corresponde a um comando Linux sem opções adicionais. Por exemplo, é permitido especificar **--execute date**, mas não **--execute date +%Y%m%d**. Sempre que a aplicação recebe o sinal SIGUSR1, a aplicação deve mostrar uma mensagem apropriada na saída padrão e seguidamente executar o comando que foi especificado através do parâmetro **--execute/-e**. Caso a execução do comando falhe, a aplicação **exec_on_signal** deve terminar.

Quando recebe o sinal SIGINT, a aplicação deve terminar, indicando através de uma mensagem para a saída padrão que recebeu o sinal SIGINT.

Considere os seguintes exemplos de execução. As linhas sublinhadas correspondem à saída padrão produzida pela execução do commando:

<pre>\$/exec_on_signal -e xpto [PID:7851]: command to execute is 'xpto' [INFO] kill -s SIGUSR1 7851 (to execute command) [INFO] kill -s SIGINT 7851 (to terminate) [INFO] SIGUSR1 received. Executing command 'xpto' xpto: No such file or directory [ERROR] Failed execution: exiting!</pre>	<pre>\$/exec_on_signal --execute date [PID:7823]: command to execute is 'date' [INFO] kill -s SIGUSR1 7823 (to execute command) [INFO] kill -s SIGINT 7823 (to terminate) [INFO] SIGUSR1 received. Executing command 'date' Sun Nov 5 01:09:19 WET 2017 [WAITING for signal] [INFO] SIGUSR1 received. Executing command 'date' Sun Nov 5 01:09:21 WET 2017 [WAITING for signal] [INFO] SIGINT received. Exiting!</pre>
---	--

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 IPEiria 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-Templatev3.02.zip**.

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

Duas pessoas têm a missão de trocar o pneu furado de um carro o mais rápido possível e, para isso, foram distribuídas as tarefas da seguinte forma:

Pessoa	Tarefa
1	Soltar (desapertar ligeiramente) parafusos
2	Levantar carro (usando o 'macaco')
1	Retirar parafusos
1	Retirar o pneu
2	Colocar o pneu suplente
2	Apontar parafusos (posicionar e apertar de forma ligeira)
1	Baixar o carro
2	Apertar parafusos

Tenha em consideração as seguintes regras:

- As tarefas têm que ser executadas de forma sequencial;
- Cada tarefa deve demorar N segundos cada. O valor de N é especificado através da opção `--duracaoTarefa / -d <valor>`. A aplicação deve validar que o valor indicado é positivo.

Recorrendo à linguagem C, e fazendo uso de threads, elabore a aplicação troca-pneu que simule a ação das duas pessoas no desempenho das suas tarefas.

Tenha em consideração os seguintes exemplos de execução da aplicação (P1?=Pessoa1; P2=Pessoa2):

<pre>./troca-pneu P1 - Soltei parafusos P2 - Levantei o carro P1 - Retirei parafusos P1 - Retirei o pneu P2 - Coloquei pneu suplente P2 - Apontei os parafusos P1 - Baixei o carro P2 - Apertei parafusos</pre>	<pre>./troca-pneu --duracaoTarefa 1 [INFO] Cada tarefa tem duracao 1s P1 - Soltei parafusos P2 - Levantei o carro P1 - Retirei parafusos P1 - Retirei o pneu P2 - Coloquei pneu suplente P2 - Apontei os parafusos P1 - Baixei o carro P2 - Apertei parafusos</pre>
---	---