L:EIC / SO2122
Trabalho Prático

Q1. Escreva um programa `phrases` que dado um ficheiro de texto na linha de comando imprima o número de frases no texto e as escreva de seguida, uma por linha no terminal. Utilize o mais possível as funções da "Standard C Library" para manipulação de "strings" e ficheiros ("buffered I/O"). Note que as frases podem terminar com um ponto final mas também com pontos de exclamação ou de interrogação.

```
$ phrases
usage: phrases [-l] file
$ cat quote.txt
Such is the nature of evil. Out there in the vast ignorance of the
world it festers and spreads. A shadow that grows in the dark. A
sleepless malice as black as the oncoming wall of night. So it ever
was. So will it always be. In time all foul things come forth.
-- Thranduil, Elven King
$ phrases quote.txt
8
$ phrases -l quote.txt
[1] Such is the nature of evil.
[2] Out there in the vast ignorance of the world it festers and spreads.
[3] A shadow that grows in the dark.
[4] A sleepless malice as black as the oncoming wall of night.
[5] So it ever was.
[6] So will it always be.
[7] In time all foul things come forth.
[8] -- Thranduil, Elven King
```

Q2. Escreva um programa `addmx` que para duas matrizes de números inteiros positivos $n \times m$ fornecidas na linha de comando calcule uma matriz que $n \times m$ que é a sua soma. O programa deverá lançar m processos filho. Cada processo P_i, com $0 \leq i \leq m-1$, é responsável por calcular a coluna i da matriz resultado. As matrizes dadas como input e a matriz resultado devem ser colocadas num segmento de memória partilhado pelos processos para que todos tenham acesso às ditas (sugestão: veja como usar a função `mmap` da clib). A matriz resultado é assim calculada "em paralelo" pelos vários processos filho. O processo pai deve esperar pelo fim de todos os processos filho após o que escreve para o "stdout" a matriz resultado com o mesmo formato do ficheiro original. Um exemplo:

```
$ addmx
$ usage: addmx file1 file2
$ cat matrix1.txt
4x3
5 4 4
7 1 7
8 5 4
2 4 4
$ cat matrix2.txt
4x3
2 3 3
2 5 1
1 1 9
6 2 2
$ addmx matrix1.txt matrix2.txt
4x3
7 7 7
9 6 8
9 6 13
8 6 6
```

Q3. Escreva um programa `cypher` que crie duas "pipes" entre um processo (pai) e um seu processo filho. Uma das "pipes" permite enviar informação do processo pai para o processo filho. A outra permite enviar informação no sentido contrário. O processo pai recebe uma ou mais strings do "stdin"e envia-a para o processo filho através de uma das "pipes". O processo filho recebe-a e substitui todas as palavras de acordo com a informação no ficheiro `cypher.txt`. Depois de processada envia-a de volta, através da outra "pipe", para o processo pai, que a escreve no "stdout". Nota: quando aplicado duas vezes o programa deve produzir o texto original, pois a composição de duas substituições deriva cada palavra original.

```
$ cat cypher.txt
evil good
dark light
black white
night day
$
$ cat quote.txt
Such is the nature of evil. Out there in the vast ignorance of the
world it festers and spreads. A shadow that grows in the dark. A
sleepless malice as black as the oncoming wall of night. So it ever
was. So will it always be. In time all foul things come forth.
-- Thranduil, Elven King
$
$ cypher < quote.txt > quote_cyphered.txt
$ cat hobbit_cyphered.txt
Such is the nature of good. Out there in the vast ignorance of the
world it festers and spreads. A shadow that grows in the light. A
sleepless malice as white as the oncoming wall of day. So it ever
was. So will it always be. In time all foul things come forth.
-- Thranduil, Elven King
$
$ cypher < quote_cyphered.txt
Such is the nature of evil. Out there in the vast ignorance of the
world it festers and spreads. A shadow that grows in the dark. A
sleepless malice as black as the oncoming wall of night. So it ever
was. So will it always be. In time all foul things come forth.
-- Thranduil, Elven King
```

Considerações Gerais. Para garantir que os resultados que obtiveram podem ser reproduzidos pelos docentes, devem fazer os testes finais de todos os programas na máquina `gnomo.fe.up.pt`.

Quando o trabalho estiver terminado, cada grupo deve enviar por e-mail, para o docente da turma prática respectiva, um arquivo `.zip` com uma pasta - `tp` - contendo três subpastas - `q1`, `q2` e `q3` - cada uma contendo o código fonte C para o problema respectivo bem como um ficheiro `makefile` com regras que permitam compilar o programa e limpar ficheiros temporários e binários. A pasta `tp` deve incluir também um ficheiro de texto com os nomes completos e os números mecanográficos dos elementos do grupo.

Para além do envio do código fonte com a resolução dos problemas, como descrito acima, cada grupo deverá fazer uma demonstração do trabalho prático para o docente da turma prática respectiva em data a combinar. *Nesta apresentação é obrigatória a presença de todos os membros do grupo.*

Bom trabalho!

A equipa da unidade curricular "Sistemas Operativos"