

Prova de Laboratori de COM

Crea un fitxer, amb el nom **"respostes.txt"** per escriure les respostes a preguntes que no siguin d'implementar codi. NOTA: Es recomana utilitzar l'interpret de comandes **"bash"**.

Exercici 1 (5,5 punts)

En aquest exercici, treballarem amb el codi font **"encriptador.c"** que s'encarrega de llegir un fitxer de text comprensible (text no xifrat) i retorna un text encriptat intel·ligible (text xifrat). Per exemple, si es passa un fitxer que conté el text no xifrat **"Hola!"**, retorna el text xifrat **"Ryvk+"**.

Abans de començar, crea una carpeta, en el teu directori \$HOME, anomenada **"Exercici1"** i copia a dins els arxius **"encriptador.c"** i **"key.c"** (fent servir el mètode que vulguis).

- 1) **(1,5 punts)** Inspecciona el codi **"encriptador.c"** i respon a les següents preguntes en el fitxer **"respostes.txt"**:
 - a) Quants arguments d'entrada ha de rebre i de quin tipus són?
 - b) El codi funcionaria igualment bé si **"c"** fos un integer? Podem tenir problemes a l'hora de sumar les variables **"key"** i **"c"** generant un valor no representable? Tant si és així com si no, explica per què i com es comporta el codi?
 - c) Si canviem el primer paràmetre de la crida a sistema **"write"** i posem **"2"** en lloc **"1"**, quina conseqüència obtindrem a l'executar la nova versió? Explica-ho breument.
- 2) **(1 punt)** Escriu en el full de respostes la/les línia/es de comandes que s'ha/n d'executar en una terminal per fer les següents accions (NOTA: *no busquem optimització en quant a reduir el nombre de línies de comandes, només realitzar el que es demana*):
 - a) Crea un directori anomenat **"v2"** dins del **"Exercici1"**. A continuació, copia el fitxer **"encriptador.c"** per a què tingui el nom **"encriptador_v2.c"**. Aquesta còpia l'hauràs de crear dins del directori **"Exercici1/v2"**. Si ja conté un arxiu amb el mateix nom, assegura't que se li demana al usuari que confirmi si vol sobreescriure el fitxer.
 - b) Moure l'arxiu **"key.c"** de **"Exercici1"** al directori **"Exercici1/v2"**.
- 3) **(1 punt)** Modifica la línia 13 del **"encriptador_v2.c"**:

key = 10;

per la següent:

key = getKey();

La funció **getKey()** està en el fitxer **"key.c"**. No s'ha proveït el fitxer **"key.h"**. Si creus que fa falta generar-lo, crea'l. Si creus que no fa falta, justifica per què. A més, apart de l'esmentat canvi en la línia 13, indica al full de respostes quins altres canvis has d'efectuar a **"encriptador_v2.c"** per a què compili i funcioni correctament (justificant-ho breument) i implementa'ls.

- 4) **(1 punt)** Repeteix el procés de l'apartat 2 (no cal que ho tornis a escriure al full de respostes), per crear el directori "Exercici1/v3" i copiar el fitxer original "encriptador.c" a dins d'aquest nou directori, anomenant-lo **"encriptador_v3.c"**. Modifica aquesta nova versió per incloure els següents canvis:
- a) el paràmetre "key" s'ha de rebre com a argument d'entrada.
 - b) el text xifrat s'ha de escriure en un fitxer de sortida que s'ha de rebre també com a argument d'entrada. Si aquest fitxer de sortida no existeix, crea'l només amb permisos de lectura i escriptura per tothom (creador, grup, i altres).
- 5) **(1 punt)** Crea un fitxer **"Makefile"** en el directori "Exercici1" que permeti generar tots els programes de l'enunciat alhora i també cadascun per separat. Afegeix una regla "clean" per esborrar tots els binaris i/o fitxers objecte, i deixar només els fitxers font. Els programes s'han de generar si, i només si, hi ha hagut canvis als fitxers font. Els noms dels executables són "encript", "encript_v2", i "encript_v3" corresponents a cadascuna de les versions descrites anteriorment. Els tres executables s'han de crear al directori "Exercici1".

Exercici 2 (4,5 punts)

Per fer aquest exercici farem servir els executables "proc1", "proc2" i "proc3" que trobaràs a la carpeta "Exercici2". També et facilitem el fitxer "launch.sh" que vam fer servir per executar diversos processos alhora i en unitats de processament (processing units) concrets. A continuació, detallem la definició del que fa "launch.sh" (extret dels enunciats de les pràctiques):

The "launch.sh" file is a special group of command lines (also known as shell script) that can be interpreted by the Bash Shell in your terminal. This file needs pairs of input parameters. The first parameter in each pair indicates the n-th processing unit that will run the program indicated by the second parameter in the pair. For example:

```
#> ./launch.sh 0 ./proc1 1 ./proc2
```

Hauràs de decidir si has de fer servir el "launch.sh" o no per resoldre els següents exercicis:

- 1) **(1 punt)** Indica en el full de respostes quina/es línia/es de comandes has d'executar per esbrinar el nombre d'unitats de processament i cores de l'ordinador. Explica breument com estan distribuïdes les unitats de processament entre els cores en l'ordinador que estàs utilitzant per fer aquesta prova.
- 2) **(1 punt)** Executa "proc1" i "proc2" en segon pla (background). Un immediatament després de l'altre, però seqüencialment. Dona llibertat al SO per assignar les unitats de processament. Indica, en el full de respostes, quines línies de comandes has d'executar per:
 - executar els processos tal i com s'indica.
 - mesurar el temps d'execució (des de l'inici fins que acabi) i el promig de l'ús de CPU durant aquest període.
- 3) **(1 punt)** Executa "proc1" i "proc2" en segon pla (background). Un immediatament després de l'altre, però concurrentment i forçant a que s'executin a la mateixa unitat de processament. Indica en el full de respostes les línies de comandes que has d'executar per fer-ho. Mostra els valors de temps i promig d'ús de CPU que has obtingut en aquest cas. A més, explica, raonant breument la teva resposta, quina conclusió obtens al comparar les mètriques de l'apartat anterior amb les mètriques d'aquest.

- 4) **(1,5 punts)** Indica en el full de respostes les línies de comandes per trobar i poder identificar les regions de memòria (és a dir, data, code, heap, stack) de “proc3” mentre està en execució. Indica els rangs d’adreces de memòria de cadascuna. A més, explica breument si el programa ha sigut compilat amb enllaçament dinàmic o estàtic i com ho has esbrinat?

Lliurament

Has de lliurar el fitxer “**respostes.txt**”, “**Makefile**” i els fitxers de codi que hakis editat. Ho pots fer amb la següent línia de comandes:

\$ tar czvf ProvaLab.tar.gz respostes.txt Makefile ...

NOTA: hem posat “...” per a què posis els noms dels fitxers de codi que hakis editat separats per espai