

Entrega Tema 5: Gestió de Memòria

1. Escriu un programa en C que faci servir la crida malloc per demanar memòria dinàmica. No t'oblidis d'alliberar la memòria demanada quan has acabat de fer-la servir. El programa rebrà el nom d'un fitxer de text per la línia de comandes. Has d'esbrinar la mida del fitxer i demanar memòria per llegir tot el fitxer en memòria. Un cop en memòria mostra el text per pantalla encipatat girant cada dos caràcters (veure exemple). El programa ha de minimitzar els accessos a disc.

prompt\$ more in.txt

You're seventeen and tunnel-vision drunk,

prompt\$./problema in.txt

oY'uers veneetnea dnt nuen-livisnod urkn.

```
int main(int argc, char *argv[])
{
    int fitxer, lectura=0;
    char *mall;
    int mida=0;
    if (argc!=2) exit(-1); //error
    fitxer = open(argv[1], O_RDONLY);
    if (fitxer<0) { //error
        perror("S'ha produït un error al obrir el fitxer\n");
    }

    mida = lseek(fitxer, 0, SEEK_END);
    mall= malloc(mida);

    lseek(fitxer, 0, SEEK_SET);
    lectura=read(fitxer, mall, mida);
    if (lectura < 0) { //error
        perror("Error al read\n");
    } else {
        for (int i=0; i<lectura; ++i) {
            if (write(1, &mall[i], sizeof(char))<0) //error
                perror("Error al write\n");
        }
    }
    close(fitxer);
}
```

Normes per corregir l'exercici 1:

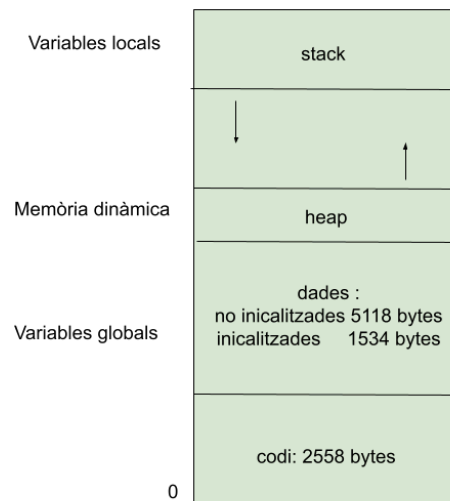
2 punts en total. Restar 0,25 per cada error/perror

→ **NOTA:** 1,75 punts (errors a l'últim for amb un write de menys)

2. Considerem que un programa es divideix en quatre àrees: codi, pila, dades inicialitzades i dades no inicialitzades. Volem poder escriure i llegir sobre la zona de dades i a la pila però volem que la zona de codi no es pugui modificar (be sigui voluntàriament o per error del programa). Suposant que l'espai lògic del processador es lineal de mida 16MBytes i el sistema de memòria que fem servir es paginació sota demanda amb pàgines de 2Kbytes, contesteu les preguntes tenint en compte les mides de les diferents zones del codi que es detallen a la següent taula:

—Zona —	Mida—
— Codi —	2558 bytes
— Pila —	6KBytes
— Dades inicialitzades—	1534 bytes
— Dades no inicialitzades—	5118bytes

- a) Dibuixa l'espai lògic del procés que es crea per executar el programa anterior. També cal tenir en compte que volem que tan la pila com les dades no inicialitzades puguin créixer de manera dinàmica, es a dir, a mida que el programa s'executa.



b) Indica quin tipus de fragmentació interna i externa es produeix, i perquè. El codi no podia compartir les pàgines ja que no es pot saber si aquest té el permís de lectura, llavors no es crea una fragmentació interna de la memòria ja que el codi només ocupa més d'una pàgina amb 2558bytes, ja que es una paginació molt petita.

c) Respon a les dues preguntes anteriors de nou, suposant un sistema d'assignació de memòria segmentat. L'espai si hi hagués una memòria segmentada, l'espai quedaria segmentat i les mides de les particions tindrà una mida variable, quedarien espais de memòria lliure disponible però que no es podria aprofitar perquè la mida de la memòria seria massa petit pel possible codi.

d) Quants bits necessito per direccionar l'espai d'adreces del procés? I com es reparteixen, es a dir quants bits indiquen l'entrada a la taula de pàgines i quants bits el offset dins la pagina? Quina mida te la taula de pagines si cada entrada te 16 bits??

Es necessiten 24 bits, ja que l'espai lògic des de 16Mbytes, si la pagina es de 2kbytes es necessiten 11 bits per indicar els bits offset d'aquesta, llavors queden 13 bits per indicar la pàgina, i aplicada a la taula de les pàgines quedaria amb 2^{14} entrades amb 16bits cadascuna.

Normes per corregir l'exercici 2:

3 punts en total. 0,75 per apartat correcte.

→ **NOTA:** 2,25 punts

3. Un sistema Pentium basat en assignació dinàmica de memòria implementa taula de pàgines multinivell amb 2 nivells: El primer nivell es el directori i el segon nivell la taula de pàgines seccionada. L'espai lògic del processador es de 4GB i la mida de la memòria principal es de 1GB. El sistema utilitza els 12 bits de menor pes de la adreça lògica per seleccionar el contingut d'una pagina i els 10 bits de mes pes per seleccionar les entrades del directori de taula de pàgines. Contesta lessegüents preguntes:

a) Quina mida te una pagina (en KB)? Suposa que posició de la pagina es de 1 Byte. Si la pàgina es de 1 Byte i la mida de la pagina es de 4KB es la mida de la pàgina, els bits de la pàgina serán 12 bits per indicar el offset.

b) Quants marcs (frames) te la memòria principal (en K frames)?
La memòria principal es de 1GB i els marcs son de la mateixa mida sobre les pàgines de 2KB, hi hauran 256Kmarcs.

c) Quantes taules de pagina (o seccions de taula de pagina) pot tenir el sistema (en K taules)?
El nombre de paraules variara depèn de les entrades al directori de 1Ktaules.

d) Quantes entrades te cada Taula de pagina (o secció de TP) (en K entrades)?
La suma dels 10 bits per indicar el directorio els 12 bits per indicar l'offset, quedarien 10 bits per indicar l'entrada, tenint així una adreça completa de 32 bits.

e) Quina mida te el directori si cada entrada te 16 bits (en KB)?
Amb l'entrada de 16bits i una 1k Entrades mediria 2KB com a mida del directori.

Normes per corregir l'exercici 3:

3.75 punts en total. 0,75 per apartat correcte.

→ **NOTA:** 3,75 punts

4. Explica

a) que es una condició de carrera (race condition) i posa un exemple de com es pot produir i en quin context.
Aquesta condicio només es pot produir si hi ha un recurs compartit i aquest es fa servir per diferents processos, aquest poden produir diferents resultats depenen del seu ordre d'execució i podria donar resultats que no tinguessin relació amb el procés. Es un problema que passa quan hi ha una memòria compartida i fan servir un recurs compartit per la seva execució.

b) Explica també quins mecanismes ofereix el SO per evitar les condicions de carrera.
El SO ofereix mecanismes poder crear un accés en exclusió mútua per exemple semàfor, garanteix que si ja hi ha algun procés a la accedint a la regió compartida, els altres procesos

que intenten accedir o que estiguin en execució es quedin parats a l'espera fins que la regió compartida hagi acabat d'executar-se o d'utilitzar-se.

Normes per corregir l'exercici 4:

1.25 punts en total, (a) 0,75 (b) 0,5

→ **NOTA:** 1.25 punts