Examen Final de Sistemes Operatius (SIOP) EPSEVG

Sistemes Operatius (SIOP) EPSEVG

12-Jan-2022

1 Exercici pràctic

1. **Volem escriure un programa que representi un petit Betlem, seguint els següents passos:**

   (a) **Primer, programeu un programa que li direu *jesus* que faci un echo, llegeixi de la stdin i escrigui a la stdout. Feu dos versions una que ho faci caràcter a caràcter i una altre que faci servir la tècnica de buffering.**

   (b) **Segon pas, programeu un programa que li direu *rei* que escrigui per la stdout un caràcter per segon de l'argument que rebi per la línia de comandes. A continuació es mostra un exemple de quin és el comportament esperat del programa *rei*:**

   ```
   #./rei mirra
   mirra
   (es mostra per pantalla un caràcter cada segon: m (1 segon) i (1 segon) ...
   ```

   (c) **Tercer pas programeu un programa que li direu *betlem* que crei un proces *jesus* i tres processos *rei*. El programa ha de llegir d'un fitxer que rebrà com argument per la línia de comandes (i.e. *regals.txt*) la llista de regals i repartir un regal per rei. Cada regal ocupa una línia del fitxer i pots assumir que el fitxer nomes tindrà tres línies Els processos rei han d'enviar cadascú el seu regal per una pipe que llegira el proces jesus. El proces jesus escriurà els regals que rebi de la pipe per la seva stdout. En aquest programa has de fer servir els programes del primer i segon pas tal i com estan sense modificar-los. A continuació es mostra un exemple del contingut del fitxer *regals.txt* i com s'invocaria el programa *betlem*:**

   ```
   # cat regals.txt
   or
   incens
   mirra
   # ./betlem regals.txt
   ```

   La següent figura 1 mostra un esquema dels processos implicats en el programa betlem, el seu parentesc i com es comunicarien.

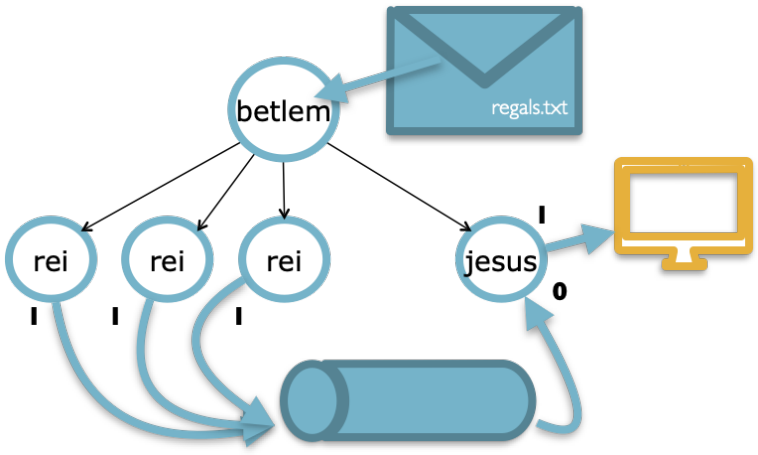

Figura 1: Esquema de processos del programa betlem

Es demana:

(a) **Programa el programa jesus tal i com es demana, recorda fer control d'errors de les crides al sistema.**

(b) **Programa el programa rei tal i com es demana, recorda fer control d'errors..**

(c) **Programa el programa betlem, tal i com es demana i es mostra a la figura 1, recorda fer control d'errors i esperar l'acabament de tots els processos fills que creis.**

(d) **Quina serà la sortida del programa? Serà sempre la mateixa?**

(e) **Modifica el programa *betlem* de manera que la sortida per pantalla sigui la mateixa que el contingut del fitxer de regals però en ordre invers . Raona si els canvis que has fet han afectat la concurrència del programa betlem i de quina manera. Exemple:**

```
# ./cat regals.txt
or
incens
mirra
# ./betlem regals.txt
mirra
incens
or
```

Puntuació exercici 1: 5 punts en total. Es valorara que el programa funcioni però recorda que tens temps limitat, et recomano que primer resolguis tots els apartats de l'exercici amb pseudocodi i després el facis funcionar. Això es per si se t'acaba el temps que puguis entregar alguna cosa.

1. *Aquest apartat (programa* jesus*) fa referencia a la gestió de l'entrada sortida. Cal tenir en compte els errors que es poden produir (0,75 punts).*

2. *Aquest apartat (programa* rei*) fa referencia a la gestió de signals i gestió de l'entrada sortida. (0,75 punts).*

3. *Aquest apartat (programa* betlem*) fa referencia a la gestió de processos, gestió de l'entrada sortida i comunicació entre processos. Cal recollir l'error de totes les crides a sistema i esperar l'acabament dels fills. 2 punts*

4. *Aquest apartat valora la comprensió del funcionament de les crides al sistema. (0,5 punts)*

5. *Aquest apartat (programa* betlem *modificat) valora la comprensió de les crides a sistema que hem estudiat.(1 punt).*

2 Exercicis teòrics

2. Suposeu un sistema de fitxers (basat en inodes) que té els següents inodes i blocs de dades que es mostren a la figura 2

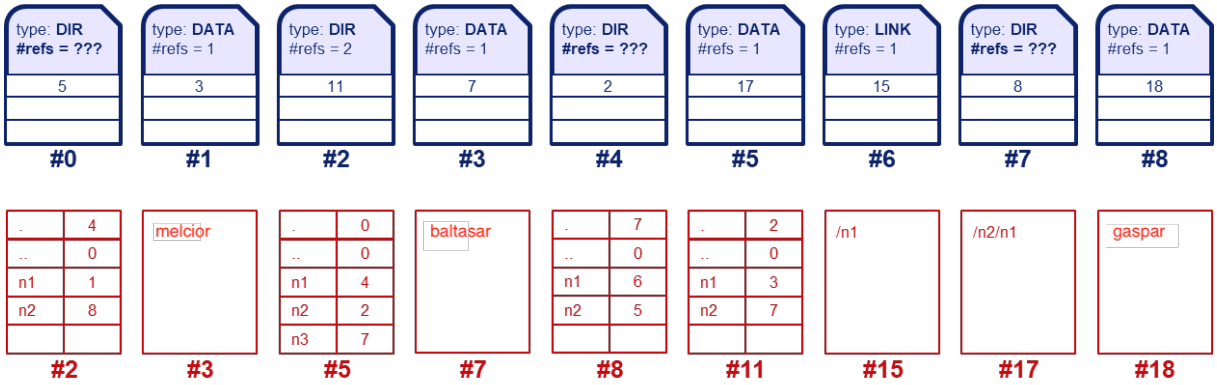

Figura 2: Esquema d'inodes i blocs de dades

(a) **Dibuixa el graf de directoris que es pot deduir d'aquest esquema de inodes i blocs de dades. Indica quin és l'inode arrel. Per fer el graf pots fer servir el format que hem fet a classe on indiquem el nom a l'aresta i el numero d'inode al node del graf. Els nodes del graf tenen diferent forma dependent del tipus: els fitxers que son soft-link els representem amb un triangle, els directoris un quadrat i els fitxers ordinaris un cercle. Indica també quins fitxers són hard-link. En el cas dels soft-link indica mitjançant fletxes discontinues el path on apunten. Una foto d'un dibuix fet a mà es suficient (per si realitzes la prova online).**

(b) **Quan valdrà el camp `#ref` per als inodes 0,4 i 7?**

(c) **Que sortira per pantalla si executem la comanda: `cat /n2/n2/n2`?**

(d) **Que sortira per pantalla si executem la comanda: `cat /n2/n2/n1/n2`?**

(e) **Indica el número d'accessos a disc que es generaria si executem la crida a sistema `open("/n3/n1/n2",O_RDONLY)` Pots assumir que el superbloc està a memòria però el sistema no disposa de cap cache. Assumeix que els inodes ocupen 1 bloc de disc.**

(f) **Ara suposa que el sistema té caches (buffer cache i dentry). Indica el numero d'accessos a disc al executar la mateixa crida. Suposa que la crides per primera vegada.**

Puntuació exercici 2: 3 punts en total. Aquesta pregunta fa referencia a gestió de fitxers. 0,5 per apartat.

3. Contesta raonadament les següents preguntes:

(a) **Explica la diferencia entre memòria lògica i memòria física Podria passar en algun cas que la memòria lògica coincideixi amb la memòria física? Raona la resposta.**

(b) **Si tinc un processador Intel de 64 bits de quina mida es el meu espai d'adreces lògic? i el físic?**

(c) **Que significa que el sistema operatiu s'executa en mode privilegiat? Si executes un codi que mostra per la pantalla un *Hello world* ho faràs en mode usuari, com pot ser que puguis escriure coses per pantalla (un recurs compartit)?**

(d) **Explica que es el Dirty bit i en quin contexte es fa servir.**

Puntuació exercici 3: 2 punts en total, fa referencia a gestió de la memòria (0,5 per apartat)