

Sistemes Operatius

Lab 1. Creació de Time Thieves (TT)

1. Gestió d'arguments

Es vol crear un TT que rebi dos valor, M i N, i comprovi que aquests son nombres enters més grans que 0. Altrament, escriurà un missatge informatiu i finalitzarà amb un -1.

A més, necessitem un replicant que rebi un valor, N, i la seva missió sigui comprovar que aquest es un nombre més gran que 0. Altrament, finalitzarà amb un -1.

- a) Identifica l'enunciat amb els continguts de l'assignatura

- TT

El programa que una vegada en execució serà un procés únic que anirà creant i executant els replicants.

- replicant

Serà un nou procés creat pel programa TT, sent ell mateix un programa fill i TT el programa pare, es creara amb la instrucció fork()

- M, N

Els paràmetres entrants pel l'entrada del teclat

- missatge informatiu

Frase que sortirà per pantalla com si es treuen dades per pantalla, amb les instruccions printf o write

- finalitzar amb -1

Instrucció que torni un -1 per acabar el procés, per exemple instrucció exit (0,-1)

- b) Crea un TT

- c) Crea un replicant

- d) Realitza les proves que creguis convenient per verificar el seu funcionament

2. Gestió de processos

Els TT han de tenir la capacitat de clonació i mutació canviant, així, la seva forma d'actuar per a ser com un replicant.

- a) Identifica l'enunciat amb els continguts de l'assignatura

Per poder clonar i mutar alguns processos, es fa servir la crida execlp() per poder mutar el programa fill comparat amb el programa pare, sino entraria en un bucle infinit de creació de programes fills.

- b) Fes que el TT crei M replicants que rebran un valor N cadascú.

c) Realitza les proves que creguis convenient per verificar el seu funcionament (diferents valors de M i N,...)

3. Gestió de la finalització

Els replicants finalitzen amb un 0 si la missió de comprovació del valor N, que els hi ha passat el TT, ha estat un èxit. La missió es considera un èxit quan N és més gran o igual a 0. Si la missió ha fracassat, el replicant ha de finalitzar amb un -1.

Per altra banda, el TT ha d'esperar que els seus replicants acabin la seva missió, recollint l'estat de finalització de cada replicant. En cas de que algun replicant no hi hagi tingut èxit en la seva missió, el TT escriurà un missatge informatiu.

- a) Realitza un pseudocodi que expliqui clarament, en termes de l'assignatura l'enunciat.

- b) Modifica el comportament del replicant

- c) Modifica el comportament del TT

4. entrenant als replicants amb dummies

Els replicants han de tenir la capacitat de robar temps en blocs de N unitats i emmagatzemar-los en contenidors creat a la seva memòria.

a) Identifica l'enunciat amb els continguts de l'assignatura

Els replicants son els programes fill que es creen a través del programa pare que anirà mirant i esperant a cada replicant que es crei que la seva missió sigui un èxit, on N sigui un valor més gran o igual a 0, sino la missió serà un fracàs

b) Modifica el replicant per a que tingui un contenidor a memòria «suficientment gran». Comprova que N es menor o igual que la mida del contenidor. En cas contrari el replicant finalitzarà amb un -1.

Per tal de entrenar als replicant abans d'atacar al Capitol, la resistència ha creat als dummies. El primer dummy que utilitzarem no fa res, simplement serveix per a que el replicant es familiaritzi amb ell.

c) identifica l'enunciat amb els continguts de l'assignatura

Els dummies seran l'entrada de valors per poder provar el programa, en el capitol seran diferents valors del temps que s'ha de robar

d) Modifica el replicant per a que utilitzi al dummy de la següent forma:

```
(...) // comprovacions inicials
dummy_ini( buffer, N );
while (!final)
    final = dummy_cpt( buffer, N );
dummy_end();
```

→ tenir en el directori dummy.h i libdummy.a

```
gcc myprog.c -o myprog -L. -ldummy
```

```
gcc programaTT.c -o programaTT -L. -ldummy
```

```
gcc replicant.c -o replicant -L. -ldummy
```