

UF2.NF1.Processos

- 1.- Visualitza en el cmd tots els processos i els seus fills
- 2.- Visualitza en el cmd tots els processos del teu usuari
- 3.- Cerca la comanda per trobar els processos zombies i matar-los a tots.
- 4.- Visualitza en el cmd tots els processos que estan bloquejats.
- 5.- Crea un mètode en Java que mostri per pantalla els fitxers i directoris de l'escriptori. Utilitza la llibreria `java.lang.Process`
- 6.- Crea un mètode en Java que rebi una comanda per teclat, l'executi i mostri els resultats. Fes la prova amb 3 comandes. Utilitza la llibreria `java.lang.Process`
- 7.- Utilitza la classe `ProcessBuilder` per crear dos processos: un que faci ping al 127.0.0.1 i un altre que obri el notepad. S'han d'executar de manera concurrent (a la vegada).
- 8.- Programa l'algorisme de l'enunciat que se't presenta a continuació. En la pròxima pràctica, farem el mateix amb fils per veure'n la diferència en temps i recursos.

El problema del cavall és un antic problema matemàtic en el qual es demana que, tenint una quadrícula de $N \times N$ caselles i un cavall d'escacs col·locat en una posició qualsevol (x, y) , el cavall passi per totes les caselles una sola vegada.

Has de realitzar l'algorisme perquè el cavall comenci en totes les posicions del tauler.

Mostra la posició inicial en què parteix el cavall i els moviments finals del cavall en el tauler. A l'exemple, el cavall parteix de la posició inicial (0,0) (Número 1 al tauler). El primer moviment vàlid el fa a (1,2) (Número 2 al tauler), el segon a (0,4), etc.

Exemple output:

Posició de inici del cavall: (0 , 0);

Posició final tauler:

| | | | | | | | |
|--|----|----|----|----|----|----|--|
| | 1 | 8 | 31 | 28 | 15 | 20 | |
| | 6 | 27 | 16 | 21 | 32 | 29 | |
| | 9 | 2 | 7 | 30 | 19 | 14 | |
| | 26 | 5 | 12 | 17 | 22 | 33 | |
| | 3 | 10 | 35 | 24 | 13 | 18 | |
| | 36 | 25 | 4 | 11 | 34 | 23 | |

Posició de inici del cavall: (1 , 0);

Posició final tauler:

| | | | | | | | |
|--|----|----|----|----|----|----|--|
| | 18 | 1 | 30 | 11 | 20 | 23 | |
| | 31 | 6 | 19 | 22 | 29 | 12 | |
| | 2 | 17 | 10 | 13 | 24 | 21 | |
| | 7 | 32 | 5 | 26 | 35 | 28 | |
| | 16 | 3 | 34 | 9 | 14 | 25 | |
| | 33 | 8 | 15 | 4 | 27 | 36 | |

etc..

Guarda't el temps que tarda el programa en executar totes les posicions inicials del cavall.