



Control de SO

27 de novembre de 2014

Enginyeria de Sistemes TIC

90 MINUTS

COGNOMS:

NOM:

GRUP de LAB:

Exercici 1. Considereu la següent ordre de la shell:

```
man -k 'good things'
```

Com us podeu imaginar, **man** és un programa escrit amb C. Indiqueu quin serà el valor d'**argv** i d'**argc** en el cas de l'ordre mostrada.

Exercici 2. Escriviu un shell script de nom **clean** tal que quan s'executa neteja el directori de treball. La neteja del directori consisteix a esborrar:

- Els fitxers objecte.
- Els fitxers executables sempre que existeixi un fitxer font amb el mateix nom.
- Els backups d'**emacs** sempre que existeixi el fitxer original corresponent.

Per exemple, si el directori de treball conté els fitxers:

```
a.c      b.c      c.c~  
a.o      b.c~  
a        m
```

i tant **a** com **m** són executables, després de la neteja els fitxers haurien de ser:

```
a.c      b.c      c.c~      m
```

Exercici 3. Considera el següent programa i assumeix que les crides al sistema sempre funcionen correctament (no hi ha errors). Raona quin és el seu comportament i determina: (a) quants processos es creen i (b) quants caràcters de cada tipus s'escriuen.

```
void main()
{
    ...
    pid1 = fork();
    printf("A");
    if (pid1) pid2 = fork();
    if (!pid2) {
        pid3 = fork();
        printf("B");
    }
    else {
        execvp(...);
        printf("C");
    }
    if (pid1 != pid3) {
        printf("D");
        if (pid1 && pid3) {
            printf("E");
            execvp(...);
        }
        pid4 = fork();
        printf("F");
    }
    else {
        printf("G");
        pid5 = fork();
    }
    if (pid2) {
        pid6 = fork();
        pid7 = fork();
        if (pid6) pid8 = fork();
        printf("H");
    }
    if (!pid1 && !pid2 && !pid3 && !pid4 && !pid5) printf("I");
    execvp(...);
    printf("J");
}
```



Exercici 4. És possible que existeixi comunicació entre dos processos sense que impliqui cap mena de sincronització? Respon aquesta pregunta justificant-la convenientment i raona quines estratègies d'implementació del sistema de comunicació podrien usar-se si s'escau.

Exercici 5. Suposa que un sistema operatiu té un sistema de comunicació entre processos de tipus *rendezvous*. Suposa també que les primitives d'aquest sistema són `send(pid, msg)` i `msg = receive()` entenent que els missatges són simples valors enters.

Dissenya un programa *A* que crea un procés *B* de forma que *B* escriu *k* vegades la lletra '@' essent *k* un valor que *A* envia a *B*. *A* i *B* només poden comunicar-se usant el mecanisme de rendezvous definit prèviament.