Pràctica 1. Comandos de gestió processos en Linux

Linux, com se sap, és un sistema operatiu **multitasca i multiusuari**. Això vol dir que múltiples processos poden operar simultàniament sense interferir-se uns amb els altres. Cada procés té la "il·lusió" que és l'únic procés en el sistema i que té accés exclusiu a tots els serveis del sistema operatiu.

Programes i processos són entitats diferents. En un sistema operatiu multitasca, múltiples instàncies d'un programa poden executar-se simultàniament. Cada instància és un procés separat. Per exemple, si cinc usuaris des d'equips diferents, executen el mateix programa al mateix temps, hi hauria cinc instàncies del mateix programa, és a dir, cinc processos diferents.

Cada procés que s'inicia és referenciat amb un número d'identificació únic conegut com Process ID PID, que és sempre un enter positiu. Pràcticament tot el que s'està executant en el sistema en qualsevol moment és un procés, incloent el shell, l'ambient gràfic que pot tindre múltiples processos, etc. L'excepció a l'anterior és el kernel en si, el qual és un conjunt de rutines que residexen en memòria i als quals els processos a través de crides al sistema poden tindre accés.

Comandos

>ps

Mostra la llista de processos del sistema, i algunes de les seues característiques: hora d'inici, ús de memòria, estat d'execució, propietari i altres detalls. Sense opcions llesta els processos creats pel usuari actual i associats al terminal d'usuari.

Opcions:

-a Mostra els processos creats per qualsevol usuari i associats a un terminal.
-l Format llarg. Mostra la prioritat, el PID del procés pare entre altres informacions.
-u Format d'usuari. Inclou l'usuari propietari del procés i l'hora d'inici.
-U usr Llista els processos creats pel usuari “usr”
-x Mostra els processos que no estan associats a cap terminal de l'usuari. Útil per a veure els “dimonis” (programes residents) no iniciats des del terminal.

L'ordre ps proporciona una informació molt interessant sobre els processos que tenim en execució. Podem saber el pid del procés, quanta memòria ocupa, quanta CPU consumeix, quant temps d'execució porta, etc.

Realitza

1. Mostra en la teua terminal el comando ps amb cadascuna de les opcions i observa que retorna el comando (no fa falta entregar res).
2. Executa en la teua terminal >ps axu (no fa falta entregar res).
3. Defineix procés i programa.

>pstree

Aquest comando mostra la jerarquia dels processos mitjançant una estructura d'arbre.
Si s'especifica el PID d'un procés, l'arbre començarà des d'aqueix procés, en cas contrari l'arbre començarà pel procés init (PID=1) i mostrarà tots els processos del sistema. Si s'especifica un usuari vàlid es mostrarà la jerarquia de tots els processos de l'usuari.
Opcions:

-a Inclou en l'arbre de processos la línia de comandos que s'use per a iniciar el procés.
-c Deshabilita la unió de processos fills amb el mateix nom (repliques d'un mateix procés).
-G Usa els caràcters de línia per a dibuixar l'arbre. La representació de l'arbre és més clara, però no funciona en redirigir l'eixida.
-h Remarca la jerarquia del procés actual (normalment el terminal). No funciona en redirigir l'eixida.
-n Per defecte els processos amb mateix pare s'ordenen pel nom. Aquesta opció força a ordenar els processos pel seu PID.
-p Inclou el PID dels processos en l'arbre.

Realitza

4. Executa
   > pstree -AGu (no fa falta entregar res).

>top

El comando top ofereix una llista dels processos similar al comando ps, però l'eixida s'actualitza contínuament. És especialment útil quan és necessari observar l'estat d'un o més processos o comprovar els recursos que consumeixen.
Opcions

-i Ignora els processos inactius, llistant únicament els que utilitzen recursos del sistema.
-d Especifica el ritme d'actualització de la pantalla en segons. És possible especificar decimals

Realitza

5. Executa
   > top -i
   Observa com s'executen els processos (no fa falta entregar res).

>kill

El comando kill, que literalment vol dir matar, serveix no sols per a matar o acabar processos sinó principalment per a enviar senyals (signals) als processos. El senyal per defecte (quan no s'indica cap és acabar o matar el procés), i la sintaxi és kill PID, sent PID el número de ID del procés. Així per exemple, és possible enviar un senyal de STOP al procés i es detindrà la seua execució, després quan es vulga manar un senyal de continuar i el procés continuarà des d'on es va quedar.

Com actua kill?

- kill -9 11428 (acaba, mata un procés completament)
- kill -SIGKILL 11428 (el mateix que l'anterior)

Realitza

6. Executa
   > top -i
   Llança el procés firefox.
   Esbrina el seu pid.
   Acaba-lo amb kill
   Anota els comandos.

Execució en primer i segon pla

Un procés es llança en 1r pla simplement introduint el seu nom en el terminal i prement intro.

>firefox

Un procés executat en 2n pla, s'executa sense bloquejar el terminal des del qual es va llançar. Els programes es poden iniciar-se en 2n pla afegint el caràcter & al final del comando.
>firefox &

Quan un procés s'inicia en segon pla, es crea un treball (job), al qual se li assigna un nombre enter, començant per 1 i numerat seqüencialment.
Quan un procés s'executa en segon pla (background), l'única manera de comunicar-se amb el procés és mitjançant l'enviament de missatges (signals).

>nice

Permet canviar la prioritat d'un procés. Per defecte, tots els processos tenen una prioritat igual davant el CPU que és de 0. Amb nice és possible iniciar un programa (procés) amb la prioritat modificada, més alta o més baixa segons es requerisca. Les prioritats van de -20 (la més alta) a 19 la més baixa. Només root o el superusuario pot establir prioritats negatives que són més altes. Amb l'opció -l de ps és possible observar la columna NI que mostra aquest valor.

>sleep

Aquest comando espera la quantitat de segons passats com a paràmetre. Podem utilitzar-ho quan desitgem executar un procés després d'una quantitat de temps. Un exemple simple (però sense utilitat) seria llistar un directori dins de 60 segons.

Executa:

(sleep 60; ls /home/usuari) &

Per a tallar l'execució pressionar CTRL + c.

Realitza

7. Executa
   > *nice*
   Què retorna?
8. Executa
   >*nice -n 5 firefox* (inicia *firefox* amb una prioritat de 5, la qual cosa li dona menys temps de cpu).
   Mitjançant top comprova que té major prioritat.
9. Esbrina què realitzen les següents ordres:
   a. jobs
   b. nohup
10. Mitjançant quines instruccions puc llistar els fitxers del meu directori dins de 24 segons?
11. Esbrina el funcionament de cron i crontab
12. Com podria programar la còpia del meu directori personal a un altre determinat tots els dies a les 23.00?