

Laboratorio di sistemi operativi
A.A. 2010/2011
Gruppo 2
Gennaro Oliva
Processi

Processo

- L'istanza di un programma in esecuzione in ambiente UNIX viene detta processo
- Ogni programma può generare diversi processi:
 - ogni utente collegato al sistema esegue la sua shell
 - ogni utente esegue la sua istanza di vi
- Ogni processo ha
 - process identification number (PID) univoco
 - parent PID (PPID) ovvero il PID del processo che lo ha generato
- init è il primo processo avviato al boot del sistema dopo che il kernel è stato caricato e ha PID=1 e nessun PPID

Real ed effective user

- A ogni processo sono associati due utenti
 - Real user: utente che ha lanciato il processo
 - Effective user: utente che determina i diritti del processo
- quando il processo accede al filesystem, il kernel verifica i permessi relativi all'effective user
- Solitamente i due utenti coincidono, ma quando un file eseguibile ha il set-uid-bit impostato, il corrispondente processo avrà:
 - Real user: utente che ha lanciato il processo
 - Effective user: utente proprietario dell'eseguibile
- Questo metodo consente agli utenti di effettuare operazioni che normalmente non potrebbero fare e al programma di stabilire l'effettiva identità dell'esecutore

Il comando passwd

- Il comando passwd serve per modificare la nostra password d'accesso al sistema
- Chiede all'utente la vecchia password, poi una volta verificato che la password coincide, chiede una nuova password 2 volte e sostituisce il campo corrispondente nel file di password del sistema (normalmente /etc/shadow)
- L'utente che esegue passwd non ha il permesso di modificare il file di sistema, ma il programma passwd ha il set-uid-bit impostato

```
$ ls -la /usr/bin/passwd  
-rwsr-xr-x 1 root root 34740 Feb 16 02:11 /usr/bin/passwd
```

Il comando passwd

- Quando viene eseguito da root il comando passwd ci permette di cambiare la password di qualsiasi utente
- Non ci chiede mai la vecchia password (neanche quando cambiamo la password di root)
- Come fa il comando passwd ad accorgersi di essere stato eseguito da root, se qualsiasi che l'utente lo esegue ottiene i permessi di root?

Il comando passwd

- Quando viene eseguito da root il comando passwd ci permette di cambiare la password di qualsiasi utente
- Non ci chiede mai la vecchia password (neanche quando cambiamo la password di root)
- Come fa il comando passwd ad accorgersi di essere stato eseguito da root, se qualsiasi che l'utente lo esegue ottiene i permessi di root?
- **Confrontando Real ed Effective user**

ps process status

- Il comando ps (process status) visualizza i processi in esecuzione sul sistema
- Scegliamo i processi da visualizzare utilizzando le seguenti opzioni:
- ps visualizza i processi eseguiti dalla shell corrente
- ps -u potter visualizza i processi con effective user potter
- ps -e visualizza tutti i processi

ps process status

- Di default su ogni processo dell'elenco vengono visualizzate le seguenti informazioni: il PID, il terminale (TTY), il tempo cpu utilizzato dal processo (TIME) ed il comando (CMD)
- `ps -f` (full) visualizza anche UID (effective user), PPID e argomenti
- `ps -F` visualizza informazioni ancora più dettagliate
- `ps -o` consente di specificare l'elenco di informazioni da visualizzare
- Esempio:
- `ps -u potter -o pid,ppid,%cpu,%mem,arg`

Esercizi

- Elencare tutte le istanze di bash in esecuzione
- Elencare tutti gli utenti che hanno almeno un processo in esecuzione (senza ripetizioni)
- Elencare i processi che non sono collegati a un terminale

Esecuzione in background

- Normalmente, la shell esegue i comandi dell'utente in foreground ovvero restituisce il controllo del terminale soltanto a comando terminato
- Utilizzando il simbolo '&' subito dopo il comando il comando viene eseguito in background e la shell restituisce subito il controllo del terminale
- Se non rediretto in file il comando in background utilizza il terminale per lo standard output e lo standard error

- Esempio:

```
$ ls -R /usr > ~/elenco
```

```
[1] 17722
```

job number

PID

Elenca ricorsivamente il contenuto di /usr e di tutte le sottodirectory scrivendolo nel file ~/elenco

Controllo di processi

- `jobs` mostra i processi attualmente in background
- `fg` riporta in foreground l'ultimo processo lanciato in background
- `Ctrl+Z` sospende il processo attualmente in foreground
- `bg` manda in background l'ultimo processo sospeso
- `Ctrl+C` termina il processo attualmente in foreground