

I Processi Demone

Un processo **demone** è un processo che **non** è associato a un terminale di controllo e fornisce un servizio:

- ad intervalli di tempo prefissati
- in risposta ad un evento

Hanno in generale un tempo di vita molto lungo, spesso coincidente col tempo di funzionamento del sistema

Esempio: demone di stampa, inetd, telnetd, sendmail, cron...

ps -axj, caso BSD:

PPID	PID	PGID	SID	TT	TPGID	UID	COMMAND
0	0	0	0	?	-1	0	swapper
0	1	0	0	?	-1	0	/sbin/init
0	2	0	0	?	-1	0	pagedaemon
1	55	55	55	?	-1	0	portmap
1	60	41	41	?	-1	0	keyserv
1	78	74	74	?	-1	0	(biode)
1	89	89	89	?	-1	0	syslogd
1	97	97	97	?	-1	0	/usr/lib/sendmail
1	103	103	103	?	-1	0	(nfsd)
1	104	104	104	?	-1	0	rpc.mountd
1	115	115	115	?	-1	0	rpc.statd
1	117	117	117	?	-1	0	rpc.lockd
1	131	131	131	?	-1	0	cron
1	136	136	136	?	-1	0	inetd
1	3016	3016	3016	?	-1	0	/usr/lib/lpd
136	5147	5147	5147	?	-1	0	in.rlogind
136	6909	6909	6909	?	-1	0	rpc.rusersd
136	7455	7455	7455	?	-1	0	in.telnetd
136	8859	8859	8859	?	-1	65	in.fingerd
1	7033	7033	7033	co	7033	0	console(getty)
7341	7799	7799	7341	p2	7799	230	ps-axj

I Processi Demone

Un processo demone può essere messo in **esecuzione**:

- al boot del sistema (file /etc/rc eseguito da processo init)
- dal demone cron, che consulta lo script
/usr/lib/crontab di sistema
crontab di utente (solo System V)
- dal comando at
- dal terminale utente

Il progetto di un demone deve portare a un funzionamento corretto qualunque sia il modo scelto per la messa in esecuzione del demone, per cui bisogna considerare l'interazione del processo demone con:

- i segnali
- il terminale
- il gruppo di processi

Progetto di un demone

- esecuzione di una **fork** con terminazione del padre, per:
 - mettere in background il figlio restituendo il controllo alla shell (considerare il caso in cui il demone venga lanciato da uno script /etc/rc)
 - rendere il figlio non leader di un gruppo (prerequisito per setsid())
- esecuzione di **setsid()** per **sganciare il processo dal terminale di controllo e staccarsi dal gruppo del processo padre** (il processo diventa leader di una nuova sessione non collegata a nessun terminale)
- **chdir("/")** per cambiare la directory ed andare in /, per non bloccare un eventuale filesystem montato, oppure **chdir("/var/spool")** nel caso di demoni che lavorino in specifiche directory del filesystem
- file mode creation mask a 0 (**umask(0)**)
- chiusura di tutti i file descriptor
- Installare un **gestore** del SIGTERM per una terminazione graduale e controllata delle sue operazioni (termine processi: SIGTERM e dopo alcuni secondi SIGKILL)
- non lasciare figli zombie:
 - raccogliere lo stato di terminazione oppure
 - generare figli segnalando al kernel di non essere interessati allo stato di ritorno dei figli stessi

Processi e terminale di controllo

```
.....
pid=fork();
if(pid>0){ /* padre */

    .....
    exit(0);
} else /* figlio */
```

ps -xj quando il padre e il figlio sono vivi (caso BSD)

PPID	PID	PGID	SID	TPGID	UID	COMMAND
9371	9372	9372	9372	9539	230	-tcsh (tcsh)
9372	9539	9539	9372	9539	230	provaback
9539	9540	9539	9372	9539	230	provaback

ps -xj dopo il termine del padre

PPID	PID	PGID	SID	TPGID	UID	COMMAND
9371	9372	9372	9372	9372	230	-tcsh (tcsh)
	1	9540	9539	9372	9372	230 provaback

Progetto di un demone

```
#include <stdio.h>
#include <sys/types.h>

int daemon_init(void)
{
    pid_t pid;
    int i;

    if ( (pid=fork()) <0) return (-1);

    if(pid>0)
        exit(0);
    else {
        /* codice eseguito dal figlio */

        setsid();

        chdir("/");

        umask(0);

        for(i=0;i<_NFILE;i++) close(i);
        /* _NFILE è una costante che indica
           il max num di fd per processo */

        return(0);
    }
}
```

Progetto di un demone

Nel progetto di un demone bisogna considerare le differenze dovute alle diverse versioni Unix.

Per esempio, in **BSD** bisognerebbe eseguire

```
#ifdef SIGTTOU
    signal(SIGTTOU, SIG_IGN);
#endif

#ifdef SIGTTIN
    signal(SIGTTIN, SIG_IGN);
#endif
```

al posto della setsid():

- setpgrp(0, getpid()) per creare un nuovo gruppo di cui il demone è leader e
- aprire il file /dev/tty e inviargli un TIOCNTTY per scollegarsi dal terminale

```
ioctl(fd, TIOCNTTY, (char *) 0)
```

Strutture dati del kernel

