

```

/* Un padre genera argv[1] figli, ognuno dei
quali esegue un ciclo infinito.
Il padre dorme per argv[2] secondi, dopodiche'
lancia un SIGUSR1 ai figli.
Il padre deve visualizzare le iterazioni compiute
da ogni figlio in argv2 secondi.
*/

```

```

#include <stdio.h>
#include <fcntl.h>
#include <signal.h>
#define mode 0777

```

```

int fd, cont=0;

```

```

void trap(signo)
int signo;
{
    char S[10];
    sprintf(S,"%d", cont);
    write(fd, S, 10);
    exit(0);
}

```

```

main(argc, argv)
int argc;
char **argv;
{
    int Numero_figli,Tempo,i, pid, ris;
    int stato, count;
    char buf[10];

    if (argc != 3) {
        printf("Errore: %s arg1 arg2\n",argv[0]);
        exit(1);
    }

```

```

    Numero_figli=atoi(argv[1]);
    Tempo=atoi(argv[2]);

```

```

    signal(SIGUSR1, trap);

```

```

    fd=creat("tmp", mode);

```

```

    for (i=0; i<Numero_figli; i++) {
        pid=fork();
        if (pid==0) {
            for(; ;cont++);
            exit(1); /* Inutile */
        }
    }
    close(fd);

```

```

    sleep(Tempo);
    signal(SIGUSR1, SIG_IGN);
    kill(0, SIGUSR1); /*segnale SIGUSR1 al gruppo*/

```

```

    for (i=0; i<Numero_figli; i++) wait(&stato);

```

```

    fd=open("tmp", O_RDONLY);
    for(i=0, count=0; i<Numero_figli; i++){
        read(fd, buf, 10);
        ris=atoi(buf);
        printf("Ris figlio %d: %d \n", ++count, ris);
    }
    close(fd);
    exit(0);
}

```