**Esercitazione di laboratorio TPSIT**

classe 4 G – a.s. 2016/2017

**CAMPANILE**

Scrivere un’applicazione in linguaggio C/Linux (campanile.c) che consenta ad un processo padre di “far suonare” tre campane rappresentate da tre processi figli ciascuno in grado di “suonare” cioè visualizzare su schermo, una propria tonalità (DIN, DON oppure DAN).

Si supponga che la sequenza delle tonalità da suonare sia :

**DIN DIN DON ….**

ripetuta all’infinito sino alla pressione della combinazione di tasti CTRL-C che termina l’esecuzione dell’applicazione multi-process.

/\*

\* Semplice scheduler che alterna le attività

\* svolte dai processi figli.

\* I tempi di esecuzione di quest'ultimi deve essere minore

\* del tempo di "time-slice" (1sec.) impostato nel padre.

\* Il lavoro dei figli viene organizzato nel metodo di gestione signal.

\* Al termine dell'attività (5 iterazioni) il padre uccide i figli

\* per non lasciare processi in esecuzione nel sistema.

\*

\* Versione ++

\* Avviato il lavoro di un figlio il padre si mette in attesa indefinita

\* Al termine del suo lavoro il filgio sblocca il padre dandogli la possibilità

\* di avviare l'altro figlio.

\*/

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <signal.h>

void GestoreFiglio1(int);

void GestoreFiglio2(int);

int main(void)

{

int pid1, pid2,i;

if (( pid1=fork() )>0)

{

if (( pid2=fork() )>0) // Processo padre

{

sleep(2); // Attende 2 secondi

for (i=0; i<5; i++) // Itera attività di scheduling

{

kill ( pid1, SIGUSR1 ); // Avvia lavoro figlio 1

sleep(1);

kill ( pid2, SIGUSR2 ); // Avvia lavoro figlio 2

sleep(1);

}

kill ( pid1, SIGKILL ); // Termina l'esecuzione dei figli

kill ( pid2, SIGKILL );

exit(0);

}

else

{ // Processo figlio 2

signal (SIGUSR2, GestoreFiglio2); // Registrazione gestore

for(;;)

pause();

exit(0);

}

}

else

{ // Processo figlio 1

signal (SIGUSR1, GestoreFiglio1); // Registrazione gestore

for(;;)

pause();

exit(0);

}

}

// Gestore del segnale SIGUSR1

void GestoreFiglio1(int ts)

{

printf("DIN int \n");

}

// Gestore del segnale SIGUSR2

void GestoreFiglio2(int ts)

{

printf("DON \n");

}

