

ATTENTI AL LUPO !!!

5th July 2004

Progetto per il corso di Sistemi Operativi Real Time a.a. 2003-2004

Gruppo 1: Laura Azzolini, Alessandro Fogliani, Alessandro Pellegrini

INTRODUZIONE Il progetto da noi sviluppato prevede l'implementazione di più task che lavorano in maniera concorrente in ambiente S.Ha.R.K..

L'idea nasce dall'esempio fornito dal programma fly, che crea ad ogni pressione di un tasto delle "mosche" che fluttuano casualmente all'interno di una scatola bidimensionale, senza però interagire fra loro. Il nostro progetto era orientato a dare un certo comportamento a tali mosche, in particolare potevamo scegliere o di creare un mondo virtuale del tutto automatico o permettere all'utente di interagire direttamente con i processi in esecuzione in maniera tale da modificare il loro comportamento. Secondo noi non era trascurabile l'interazione con l'utente, in particolare, in questa maniera ci sembrava di poter meglio comprendere l'ambiente Real Time. Da qui nasce l'idea di avere un task che funge da inseguitore di oggetti che si muovono casualmente (come le mosche) e di un ulteriore task comandato dall'utente che influenzi il movimento dell'inseguitore. Ecco come hanno preso vita il "Lupo", la "Pecora" ed il "Cacciatore".

ARCHITETTURA La parte Real Time dell'architettura, prevede la condivisione di più strutture dati per il controllo della posizione dei task in esecuzione, oltre alla condivisione della grafica. In particolare usiamo dei semafori per governare l'accesso alle strutture condivise. I dati condivisi sono stati realizzati con vettori di strutture, così che ogni pecora, ogni lupo ed il cacciatore possiedono posizioni cartesiane, un flag che segna se è stato ucciso o no ed un semaforo privato. Abbiamo limitato l'utilizzo della mutua esclusione solo alla singola struttura del vettore stesso, così da non bloccare le altre strutture nelle fasi di lettura/scrittura. Con l'utilizzo di tali vettori è stato possibile gestire la lettura/scrittura dei dati propri di ogni task e l'accesso alle strutture dati.

Pecora La pecora è un task che nasce premendo il tasto "p" da tastiera e si muove nello schermo in maniera casuale rimanendo però all'interno di un perimetro prestabilito.

Alla creazione, ad ogni task pecora viene assegnata la prima posizione libera nel vettore di strutture, alla quale accede in mutua esclusione con gli altri task pecora presenti e con i task lupo. E' possibile creare pecore fino a che il vettore non si riempie (max quattro pecore). Nella struttura che gli è stata assegnata, il task aggiorna le proprie coordinate ad ogni spostamento. L'interazione con gli altri task viene realizzata tramite il flag di vita o morte; in particolare quando un lupo ha raggiunto la pecora, attiva il flag e la pecora, notando l'attivazione muore.

Lupo Il lupo è un task che nasce premendo il tasto "l" da tastiera e si muove nello schermo, all'interno del perimetro prestabilito, scappando dal task cacciatore o inseguendo il task pecora più vicino. Se non sono presenti né il cacciatore né le pecore, il lupo rimane nella sua tana.

Alla creazione, ad ogni task lupo viene assegnata la prima posizione libera nel vettore di strutture, alla quale accede in mutua esclusione con gli altri task lupo presenti e con il task cacciatore. A seconda della posizione assunta nel vettore, il lupo si muoverà a velocità crescente. E' possibile creare lupi fino a che il vettore non si riempie (max tre lupi). Nella struttura che gli è stata assegnata, il task aggiorna le proprie coordinate ad ogni spostamento. L'interazione con gli altri task viene realizzata tramite il controllo delle posizioni e il relativo calcolo della distanza dal cacciatore e dalle pecore, leggendo nelle rispettive strutture dati. Abbiamo dato priorità alla fuga dal cacciatore se la distanza tra esso e il lupo è minore di un certo valore. Se questa condizione non si verifica, il lupo insegue la pecora più vicina. Quando il lupo raggiunge la pecora, attiva il relativo flag di vita/morte. Se il cacciatore raggiunge il lupo, gli attiva il flag di vita o morte e il lupo, notando l'attivazione, muore.

Cacciatore Il cacciatore è un unico task che nasce premendo il tasto "c" da tastiera e si muove nello schermo, all'interno del perimetro prestabilito, controllato dall'utente tramite il mouse. Lo scopo del cacciatore è quello di raggiungere i lupi ed ucciderli prima che questi ammazzino tutte le pecore.

Quando il cacciatore raggiunge il lupo, attiva il relativo flag di vita/morte.

Main All'interno del main abbiamo inserito le inizializzazioni del mouse per il task cacciatore, del semaforo di mutua esclusione per l'accesso a video, delle strutture contenute nei vettori (semafori privati e posizione) e della grafica (creazione del perimetro per il movimento dei task e visualizzazione delle istruzioni). Abbiamo definito il modello di task da utilizzare e premendo i tasti corrispondenti a pecora, lupo e cacciatore, abbiamo creato i vari task. Premendo invece il tasto "Esc" si esce dal programma.

Altre funzioni

- Draw_obj: funzione chiamata dai singoli task per disegnare i lupi e le pecore accedendo a video in mutua esclusione .

- `my_mouse_handler`: funzione richiamata dal task cacciatore per gestire lo spostamento del cursore del mouse a video e per l'aggiornamento della posizione del cacciatore.
- `bye_bye`: funzione che termina il programma chiudendo la libreria grafica e quella del mouse. Stampa inoltre a video la scritta "Bye bye!".

CONCLUSIONI Dopo aver verificato la correttezza ed il funzionamento del programma abbiamo provato a cambiare i parametri del periodo e del `wcet` di ogni task. In particolare abbiamo notato che diminuendo il `wcet` sotto il valore di 30 us, quando tutti i task erano stati creati, il programma terminava con un errore di `wcet-violation`. Per avere un buon margine di sicurezza sulle possibili configurazioni, abbiamo scelto un valore di `wcet` pari a 100 us. Per quanto riguarda il periodo lo abbiamo impostato a 40000 us perchè diminuendolo i task si muovevano troppo velocemente per avere un programma giocabile. Sotto i 1000 us anche la visibilità dei vari task era ridotta al minimo.

E' possibile creare un numero di task (pecore + lupi + cacciatore) anche più alto di quello massimo impostato nel programma, a condizione che la somma di tutti i `wcet` sia minore del periodo specificato, con un certo margine.