Sistemi Embedded e Internet-of-Things

Appunti lezione
26/11/2020

Architetture di controllo per sistemi embedded con sistema operativo

Come vengono i programmi con un sistema operativo che ci consente di utilizzare dei thread, come riscriveremo tutto quello studiato fino ad oggi ?

Superloop architettura di controllo che ha un loop che viene eseguito di continuo

Tutto ciò che veniva modellato come task nelle architetture senza sistema operativo, viene modellato come thread, quindi a questo livello diventano delle entità che devono eseguire un compito e possono essere implementate come una macchina sincrona o asincrona o super loop o event loop (architettura pervasiva e generica)

Agente è l'entità che esegue il task

Nell'esempio del prof ho un semplice Blinking Led dove emula un' accensione e spegnimento di un led nello nostro sistema, il professore utilizza un super loop controller che incapsula i metodi principali per emulare quelli utilizzati da Arduino, rispettivamente:

- Setup() Dove abbiamo le dichiarazioni delle variabili e costruttore.
- Loop() Dove abbiamo la logica del programma che vogliamo eseguire

Una volta inseriti questi due metodi astratti e protetti, li va a richiamare dentro i programmi veri e propri del blinker o del button led.
Per il momento nella parte di setup richiamata non inserisce nulla, e nella parte che si chiamava run sostituisce alcune cose tra cui la firma del metodo richiamandolo loop come dichiarato nel super loop controller.

Il prof ha creato un package con tutti i dispositivi utilizzati per gli esempi, e dentro ogni interfaccia ha inserito delle funzioni base per ognuna, esempio nel button ha inserito la funzione booleana isPressed().

Oltre alle interfacce ha inserito anche delle classi che emulano l'accensione o lo spegnimento di un dispositivo come per esempio il led.

Ci sono anche delle classi che non sono emulate, ma sono delle classi che utilizzano delle librerie che possono controllare e pilotano tutti i dispositivi e non sono emulazioni in standard output (seriale eclipse o visual studio code).

Come implemento una macchina a stati finiti con queste nuove cose aggiunte come i thread ?

A differenza delle funzioni utilizzate in Arduino per simulare un attesa come per esempio delay che sono funzioni che generano del busy waiting andando a utilizzare la CPU, in sistemi con thread andiamo a usare delle funzioni bloccanti come waitfor che rilasciano la risorsa in oggetto concedendone l'utilizzo ad altre richieste di altri thread.

Nicholas Ciarafoni

Sistemi Embedded e Internet-of-Things

Appunti lezione
26/11/2020

In una singola applicazione io posso avere più thread e utilizzare la CPU per mettere in attesa una funzione di un thread sarebbe uno spreco, quindi posso decidere di fermare per il tempo richiesto dalla funzione del thread 1 l'uso della CPU, e permettere al thread 2 di utilizzarla nel lasso di tempo in cui thread 1 con priorità ha finito il tempo di attesa o delay, senza così creare del busy waiting, e ottimizzare di fatto l'utilizzo delle risorse importanti come per esempio la CPU di questo caso.

Come è fatta una macchia stati finiti sincrona ?

SyncSFMController, è un semplice modo di implementare dei task in un modo che richiami le macchine a stati finiti sincrone dove abbiamo un discorso di periodo, ogni tot di tempo il thread procede richiamando un tick.

Comportamento periodico che a seconda del tempo deciso in un delta, abbiamo un richiamo al tick.

Esempio del button led , ho due stati per la mia macchina a stati finiti, ed a ogni colpo di clock della mia macchina a stati finiti, vado a controllare in che stato sono, se lo stato è acceso lo spengo, se lo stato è spento lo accendo.

Sostanzialmente non devo farmi lo scheduler, ma richiamo il tick con un periodo che dipende dal periodo che mi ha specificato l'utente.
Nell'esempio specifico vado a richiamare tick,con un tempo di attesa che tiene conto del fatto che la computazione dell'ultimo tick ha richiesto del tempo, quindi nel periodo si fa una sleep di un certo numero di millisecondi dipende dal tempo che tick vecchio ha impiegato per essere eseguito.

Overrun se io vedo che è trascorso più tempo rispetto al periodo che ho impostato, non solo che aspetto 0 millisecondi, anzi sono in ritardo, quindi sono in una situazione d' eccezione.

Come funziona un architettura EventLoop ?
Viene contenuto in un unico flusso di controllo, che non fa altro di rimanere in attesa nella sua coda di eventi, e interagisce con il mondo esterno solamente con gli eventi, e può percepire solamente gli stessi.
Aspetta di vedere il prossimo evento, si blocca con blocking queue, struttura a coda, concorrentemente, che blocca una risorsa.
Non appena si sblocca fa una chiamata a funzione.
Non è ancora un sistema asincrono.

Observable
classe base nel pattern observer, che è una sorgente di eventi, dove mi registro come osservatore di eventi e posso notificare un evento, per ogni osservatore io notifico ogni evento in esame.

Nicholas Ciarafoni

Appunti lezione
26/11/2020

Tutte le volte che viene premuto un pulsante, dal frame, viene cambiato lo stato dalla GUI che simula il pulsante, e poi viene notificato l'evento del button.
La notifica cosa fa ? Non fa altro che inserire questo evento nella coda degli eventi, poi chiama proccess event, per capire cosa deve fare con quel'evento specifico.

Quando ho un sistema complesso posso dividerlo in vari thread ognuno dei quali ha un event loop a parte e ognuno di questi può comunicare con altri task.
Mandando un messaggio o accodando dei messaggi.
Paradigma ad attori, sistemi che lavorano e scambiano dei messaggi e reagiscono a questi messaggi ricevuti.

Quando si lavora con event loop, negli handler che reagiscono come eventi non mi posso bloccare perché se mi blocco li, bloccherei il flusso di controllo che reagisce agli eventi, non posso bloccarmi e tutto è asincrono.

Nicholas Ciarafoni