

Esempio

Dai tre concetti : evento attività processo
derivano diversi approcci alla simulazione

Simulazione per **sequenziamento di eventi**

Simulazione per **scansione di attività**

Simulazione per **interazione fra processi**

Simulazione per interazione fra processi

Comportamento del sistema : insieme di processi
(programmi in esecuzione quasi parallela)

Entità processo

approccio naturale alla simulazione

S2.1

Esempio: interazione fra processi

Sistema di elaborazione uniprocessore con periferiche

- un processo per ogni job nel sistema,
- un processo per la CPU, uno per ogni periferica

Ogni processo viene eseguito ciclicamente

esempio: consideriamo il processo CPU :

centro di servizio; assumiamo che i job in coda per ottenere

l'elaborazione (il servizio) siano gestiti secondo la disciplina FIFO

il processo ripete ciclicamente fino

all'evento di **fine simulazione** i seguenti passi

- 1- Esamina la coda, se è vuota vai al passo 5
- 2- Preleva un utente dalla coda
- 3- Esegui il servizio* per il job selezionato
- 4- Vai al passo 1
- 5- Attendi l'arrivo del prossimo utente;
al suo arrivo vai al passo 1

* il servizio è realizzato valutandone il ritardo introdotto

S2.2

Esempio: interazione fra processi

Il passo 4 non viene eseguito finché il clock non è avanzato
di un tempo pari al tempo di servizio richiesto
sospensione temporanea del processo

Il passo 4 viene eseguito **dopo** il ritardo corrispondente al
servizio

il passo 4 è un punto di **riattivazione** del processo

Il passo 5 differisce dal passo 4 :

la durata del tempo di sospensione non è prevedibile

la riattivazione avviene **solo** tramite un **altro** processo

Il processo è allora in uno stato **passivo** (bloccato)

Un processo può essere

- attivo
- sospeso
- passivo

S2.3

Esempio: interazione fra processi

Nella simulazione per interazione fra processi una
routine gestisce i cambiamenti di stato dei processi.

Una routine salva lo stato del processo se questo viene

sospeso o bloccato.

Due liste di processi:

- { processi sospesi } (oggetti sospesi)
- { processi bloccati } (oggetti passivi)

La routine di controllo riattiva un processo al verificarsi
della condizione di riattivazione.

Un oggetto di tipo servente può essere

attivo : sta eseguendo le azioni specificate nella definizione

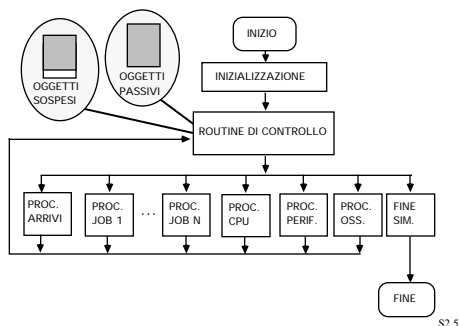
sospeso : è caratterizzato da un tempo di riattivazione

passivo : non è attivo e il tempo di riattivazione dipende da
un altro oggetto

terminato : ha terminato tutte le azioni

S2.4

Schema 1 - esempio



Esempio: interazione fra processi

Un processo cambia stato da sospeso ad attivo

tramite primitive del linguaggio.

Ad esempio la primitiva **HOLD(T)** in Simula la cui
esecuzione provoca una **sospensione** del processo per un
tempo T, preceduta dal salvataggio dello stato del processo.

Lista dei processi sospesi: insieme dei processi in attesa
del termine del periodo di sospensione
autodeterminato, ordinati cronologicamente.

Il primo processo è un oggetto **attivo**, corrispondente
al processo attualmente in esecuzione.

Lista degli oggetti passivi: insieme dei processi in attesa
di essere riattivati da altri processi al verificarsi
delle condizioni di riattivazione.

es.: processo CPU bloccato per coda vuota;

condizione di riattivazione: arrivo di un utente, generato
dal processo arrivi.

S2.6

Esempio: per eventi

Simulazione per sequenziamento di eventi

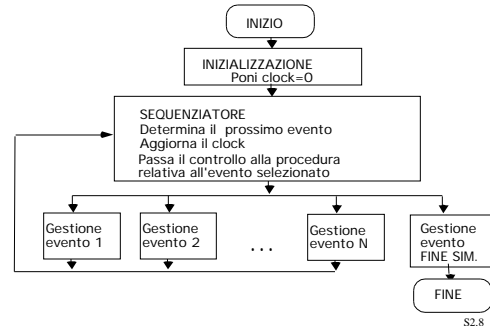
- La simulazione orientata ad eventi si basa su un insieme di procedure (sottoprogrammi), ognuna associata ad un evento, che realizzano le trasformazioni di stato relative.
- Lo schema del programma di controllo è simile al caso di simulazione orientata a processi.
- Il programma termina quando viene raggiunto il tempo di **fine simulazione**, stabilito con un criterio opportuno.

Entità procedura (segmento di programma)

- Il numero e la struttura delle procedure dipende dagli eventi considerati nel modello e dall'approccio seguito.

S2.7

Schema 2 - esempio



S2.8

Esempio: per attività

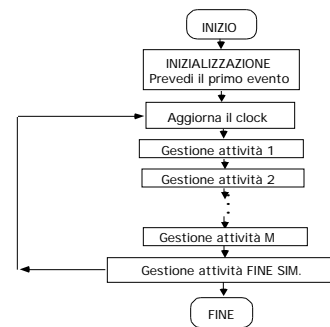
Simulazione per scansione di attività

- La simulazione per scansione di attività si basa sulla descrizione delle attività delle componenti del modello.
- Se è applicato il meccanismo di avanzamento per intervalli fissi, ad ogni avanzamento vengono esaminate tutte le attività per stabilire le condizioni di inizio / fine attività. Le relative procedure aggiornano lo stato delle entità.
- Si può applicare un meccanismo di avanzamento del tempo distribuito nelle procedure delle attività (analogo all'avanzamento per eventi).

Attività procedura (segmento di programma)

S2.9

Schema 3 - esempio



S2.10

Modello di simulazione

La definizione di un **modello di simulazione** include

- processo stocastico da studiare
- parametri e caratteristiche operative
- eventi
- variabili di stato dipendenti dagli eventi
- variabili endogene e regole di generazione
- regole di trasformazione di stato
- regole di trasformazione delle variabili endogene

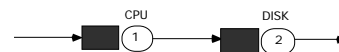
Il programma di simulazione include

- la lista degli eventi
 $E = \{(e_1, t_1), (e_2, t_2), (e_3, t_3), \dots\} \quad t_i \leq t_{i+1}, i \geq 1$
- un meccanismo di manipolazione degli eventi
- generatori di numeri pseudocasuali
- routine per l'analisi statistica dei risultati

S2.11

Esempio

Consideriamo un sistema di elaborazione uniprocessore rappresentato come un sistema di congestione aperto formato da due centri di servizio corrispondenti, rispettivamente, alla CPU e ad una unità periferica



- I programmi arrivano dall'esterno. Ogni programma (job) richiede
- un 'servizio' alla CPU (elaborazione di un segmento di programma)
- ed uno alla periferica (operazioni di I/O) e quindi lascia il sistema

Scopo dello studio è la valutazione delle prestazioni del sistema, in termini di analisi statistica del numero di job presenti in ogni componente del sistema e di altri indici, quali il

- tempo di elaborazione complessiva di un programma,
- l'utilizzazione del sistema e delle sue componenti

S2.12

Esempio

Definiamo le **variabili di stato**, ad un istante t 0

$n_i(t)$ numero di job nel centro di servizio i al tempo t
 $a_i(t)$ numero di job arrivati al centro i fino al tempo t
 $u_i(t)$ numero di job serviti dal centro i fino al tempo t
 $s_i(t)$ stato del servente del centro i al tempo t
 $(s_i(t)=0$ per servente libero, $s_i(t)=1$ occupato)

dove $n_i(t), a_i(t), u_i(t) \geq 0, i=1,2$.

- **Processo stocastico** da studiare

$P(t) = \{n_1(t), n_2(t), s_1(t), s_2(t) | t > 0\}$

- **Parametri e caratteristiche operative**

A distribuzione degli arrivi esterni al centro 1,
 di media $E[t_{a1}]$, varianza $Var[t_{a1}]$

S_i distribuzione del tempo di servizio nel centro i ,
 di media $E[t_{s_i}]$, varianza $Var[t_{s_i}], i=1,2$

discipline di servizio ai centri

p.es. : arrivi Poissoniani, A =esponenziale di media $E[t_{a1}]=10\text{sec}$,

tempo di elab. alla CPU S_1 =esponenziale, $E[t_{s1}]=50\text{ms}$,

tempo di elab. ai dischi S_2 =esponenziale, $E[t_{s2}]=100\text{ms}$, discipline di servizio **FIIFO**

Esempio

Eventi (scelta del livello di astrazione del modello)

e_i : fine servizio al centro $i, i=1,2$

e_3 : arrivo di un job dall'esterno alla CPU

e_4 : fine simulazione

Variabili di stato

t_i : tempo di occorrenza dell'evento $e_i, i = 1 \text{ a } 4$

clock : tempo simulato

Il modello di simulazione include la lista degli eventi, lista astratta delle coppie (e_i, t_i) in ordine cronologico (t_1, t_2, \dots) .

Considerando un tempo iniziale $t=0$, associamo ad ogni evento

un numero d'ordine di occorrenza $h=1,2,\dots$

p.es. $h=1$ evento di tipo e_3 arrivo di un job,

$h=2$ evento di tipo e_3 arrivo di un altro job,

$h=3$ evento di tipo e_1 servizio alla CPU, etc.

$x(h)$ denota il valore della variabile x dopo l'occorrenza dell' h -esimo evento

S2.14

Esempio

Variabili endogene

sequenza di numeri generati secondo la distribuzione A

$\langle X_1, X_2, \dots \rangle$

sequenza di numeri generati secondo la distribuzione S_i

$\langle Y_1^i, Y_2^i, \dots \rangle$

Regole di generazione:

$X_h = f(X_{h-1}, A) \quad h > 1$

$Y_h^i = f(Y_{h-1}^i, S_i) \quad h > 1, i=1,2$

Regole di trasformazione di stato e delle variabili endogene

1) selezione dell'evento successivo

2) aggiornamento delle variabili di stato

S2.15

Esempio

1) **selezione dell'evento successivo**

-avanzamento **per eventi:**

$e(h+1) = e | (e, t) \in E, t = \min_i t_i, t > \text{clock}(h)$

$\text{clock}(h+1) = t$

tempo futuro più vicino al tempo attuale

criterio di gestione di eventi simultanei

-avanzamento **per intervalli fissi:**

$\text{clock}(h+1) = \text{clock}(h) + \Delta t$

$e(h+1) = e | (e, t) \in E, \text{clock}(h) < t < \text{clock}(h+1)$

Se non esiste alcun evento e si itera il procedimento di

avanzamento del tempo, altrimenti si trattano tutti gli eventi occorsi

2) **aggiornamento delle variabili di stato**

in base al tipo di evento $e_i, i = 1 \text{ a } 4$:

S2.16

Esempio - trasformazione di stato

e_i fine servizio al centro $i = 1, 2$

$n_i(h+1) = n_i(h) - 1$

se $i=2 \quad n_1(h+1) = n_1(h)$

se $i=1 \quad n_2(h+1) = n_2(h) + 1$

se $i=2 \quad a_j(h+1) = a_j(h) \quad j=1,2$,

se $i=1 \quad a_1(h+1) = a_1(h); a_2(h+1) = a_2(h) + 1$

$u_i(h+1) = u_i(h) + 1$

$u_j(h+1) = u_j(h) \quad j \neq i$

$s_i(h+1) = 1$ se $n_i(h) > 1, s_i(h+1) = s_i(h)$ altrimenti

$s_j(h+1) = s_j(h) \quad j \neq i$ se $i=2; s_2(h+1) = 1$ se $i=1$

$\text{clock}(h+1) = t_i$

$t_i = \text{clock}(h+1) + Y_{u_i(h+1)}^i$

S2.17

Esempio - trasformazione di stato

e_3 : arrivo di un job dall'esterno alla CPU

$n_1(h+1) = n_1(h) + 1$

$n_2(h+1) = n_2(h)$

$a_1(h+1) = a_1(h) + 1$

$a_2(h+1) = a_2(h)$

$u_j(h+1) = u_j(h) \quad j=1,2$

$s_1(h+1) = 1$

$s_2(h+1) = s_2(h)$

$\text{clock}(h+1) = t_3$

$t_3 = \text{clock}(h+1) + X_{a1(h+1)}$

e_4 : fine simulazione

specifico del calcolo delle statistiche per l'analisi dei risultati e la produzione

delle variabili endogene.

Le regole di trasformazione per ogni evento includono le istruzioni necessarie

per tali analisi statistiche.

Variabili endogene: lunghezza di coda al centro i , tempo di risposta,

utilizzo, throughput

S2.18

Esempio - struttura simulatore

Struttura del simulatore per sequenziamento di eventi :
[lista degli eventi](#)
 ad ogni evento è associato il tipo (e, l i 4) e il tempo di occorrenza

[una routine per ogni tipo di evento](#)
 che viene attivata quando occorre un evento di quel tipo

[clock](#)

[programma di controllo](#)

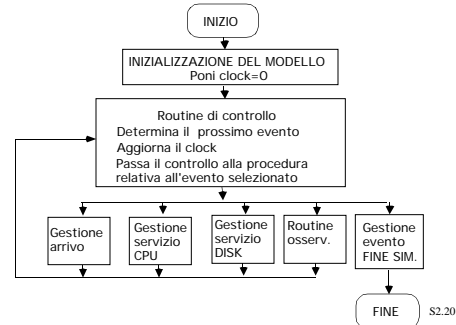
seleziona il prossimo evento nella lista
 aggiorna il clock
 trasferisce il controllo alla routine relativa all'evento selezionato

La lista degli eventi è accessibile in lettura da tutte le routine, ma in scrittura solo dalla routine di controllo.

Per fini statistici, una routine di "osservazione" viene attivata periodicamente (introdurre un evento "osservazione")

S2.19

Schema



S2.20

Esempio - struttura simulatore

Dichiarazione variabili e strutture dati

<variabili di stato>
 <variabili esogene>
 <lista eventi>
 <code>
 <clock>

Definizione delle procedure di evento

Programma di controllo

```
begin
<inizializzazione>
<prevedi il primo evento di arrivo>
while clock<t, do
begin
<seleziona prossimo evento e con tempo t >
<clock t >
<esegui la procedura relativa all'evento e>
end;
<esegui la procedura di fine simulazione>
end;
```

S2.21

Esempio - struttura simulatore

Dichiarazione variabili e strutture dati

<variabili di stato>
 <variabili esogene>
 <lista eventi>
 <code>
 <clock>

Definizione delle procedure di evento

Programma di controllo

```
begin
<inizializzazione>
<prevedi il primo evento di arrivo>
while clock<t, do
begin
<seleziona prossimo evento e con tempo t >
<clock t >
<esegui la procedura relativa all'evento e>
end;
<esegui la procedura di fine simulazione>
end;
```

S2.22

Esempio - struttura simulatore a processi

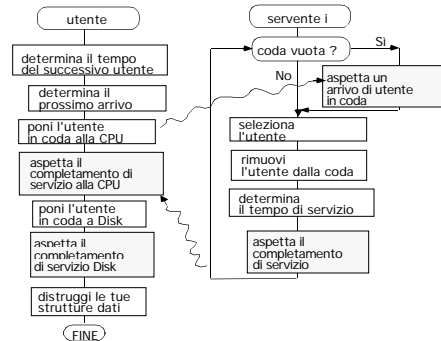
Struttura del simulatore orientato ad interazioni fra processi :

Entità processo

Processo utente
 Processo servente CPU
 Processo servente Disk

S2.23

Esempio - struttura simulatore a processi



S2.24