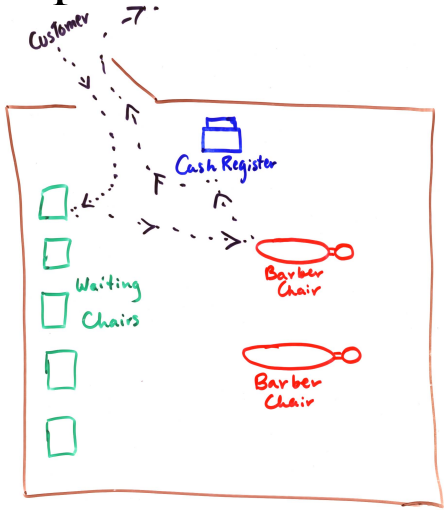
Lecture 3

Discrete Event Simulation
(Law 1.3 & 1.4, Appendix 1A)

Discrete Event Models

- Continuous time base modeled as discrete
  - Discrete Event Systems (DES) Events are instantaneous, i.e. no duration
  - Long periods of time when nothing happens
  - Time modeled as
    - Fixed-increment time advance
    - Next-event time advance

Example: The Barber Shop



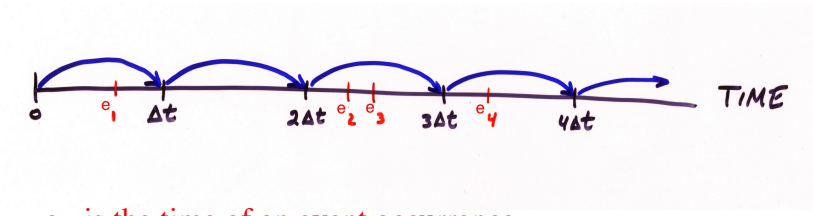
What are the events?

- Depends on what you want to know
  - How long does it take for a customer to get a haircut?
- Basic events
  - Arrival of customer in shop
  - Departure of customer from shop
  - Customer gets haircut
  - Customer finishes haircut
- Could be other events
  - Customer waits to pay for haircut
  - Customer pays for haircut

Two Ways of Keeping Time

- Fixed-increment Time Advance
- Next-event Time Advance

Fixed-increment Time Advance Overview

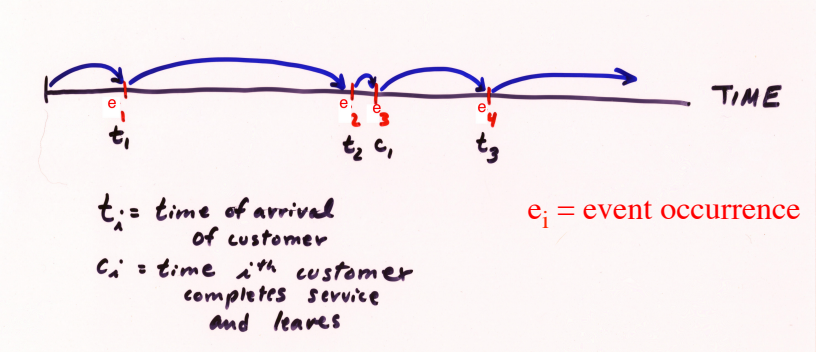


e_i is the time of an event occurrence
Δt is the time unit of clock advance

Fixed-increment Time Advance

- All events occur at a fixed increment
- Events occurring *between* the increment must be handled as if they occurred *at* the increment
- Simplest but not very accurate program
- Time is implicit in the computation

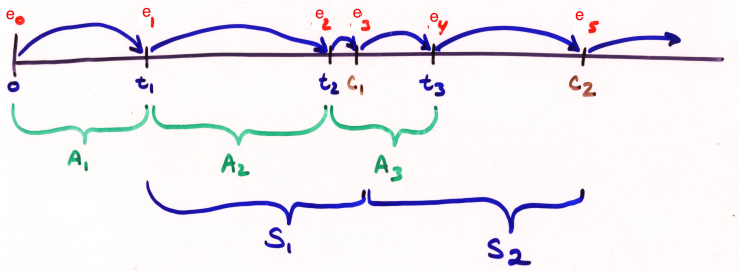
Next-event Time Advance Overview



Next-event Time Advance

- Events are modeled more accurately
- Events are scheduled when they occur
- Events are scheduled based on a random number
- Time is considered a variable in the simulation

Next-event Time Advance



e_i = time of event occurrence

t_A: = time of arrival of i^{th} customer

$C_i = t_i + D_i + S_i$ = time i^{th} customer completes service & departs

$A_i = t_i - t_{i-1}$ = interarrival time between customers

S_i = time spent serving customer

D_i = delay in queue of i^{th} customer

What do we need to create a Next-event Time Advance?

- Determine event types and understand what happens as the result of the event
- Generate event times and keep track of time
- Need some way to schedule events
- Save information created during simulation for overall analysis

Components of a DES Program with Next-event Time Advance

- Data
  - Simulation clock
    - Current value of simulated time
  - System state (descriptive variables)
    - Describes the system at the current time
    - For a queuing system, represented as queues and servers
  - Event list
    - For each event type, gives time of next event
  - Statistical counters
    - Performance of simulation: delay in queue, etc.

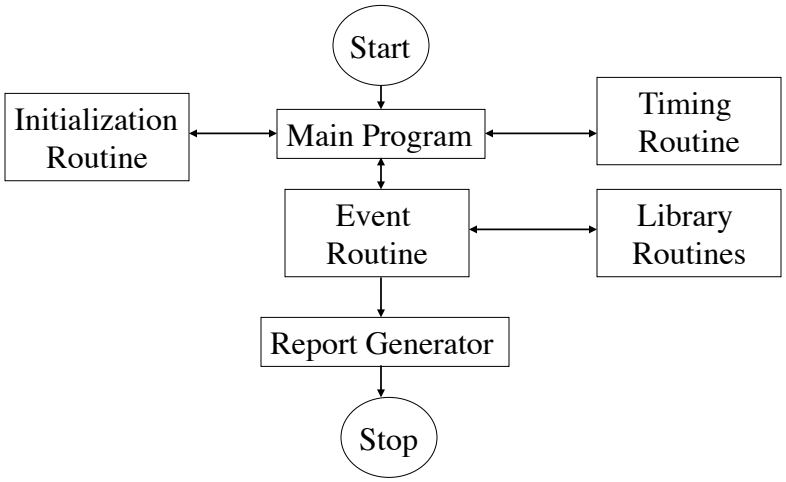
Components of a DES Program - continued

- Procedures
  - Initialization routine
    - Starts program at simulation time 0
  - Timing routine
    - Determines the next event from the event list
  - Event routines
    - Updates system state when a particular type of event occurs
    - Separate routines for each event type
  - Library routines
    - Generate PRN from probability distribution functions

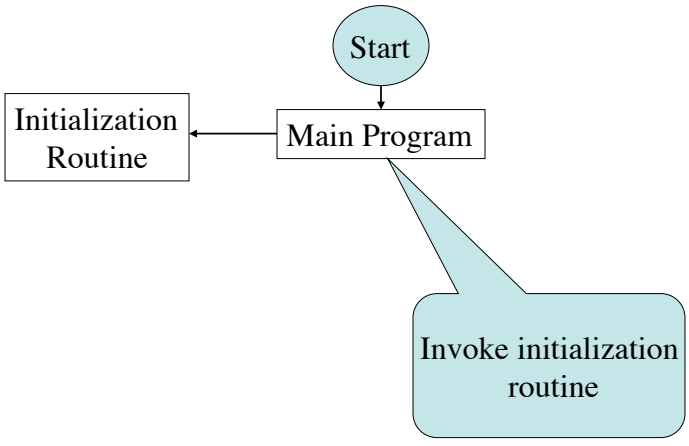
Components of a DES Program - continued

- Procedures
  - Report generator
    - Reports measures of performance from statistical counters when simulation ends
  - Main program
    - Invokes timing routine to determine next event and then transfers control to proper event routine

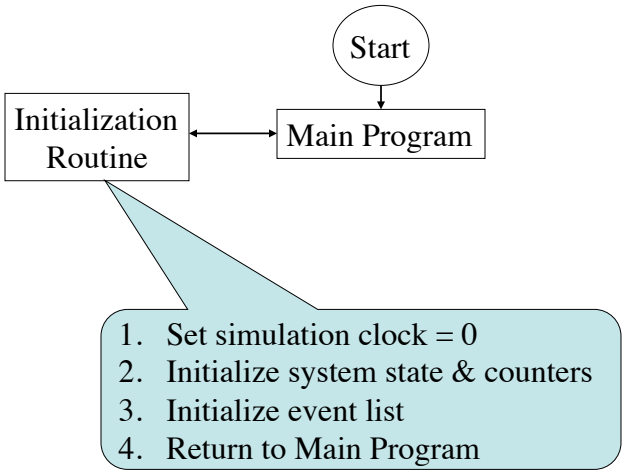
Flow of control for Next-Event



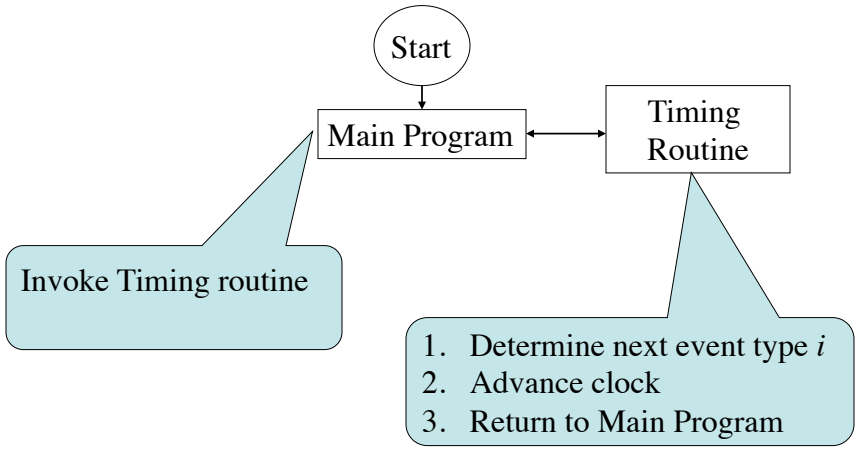
Flow of control for Next-Event



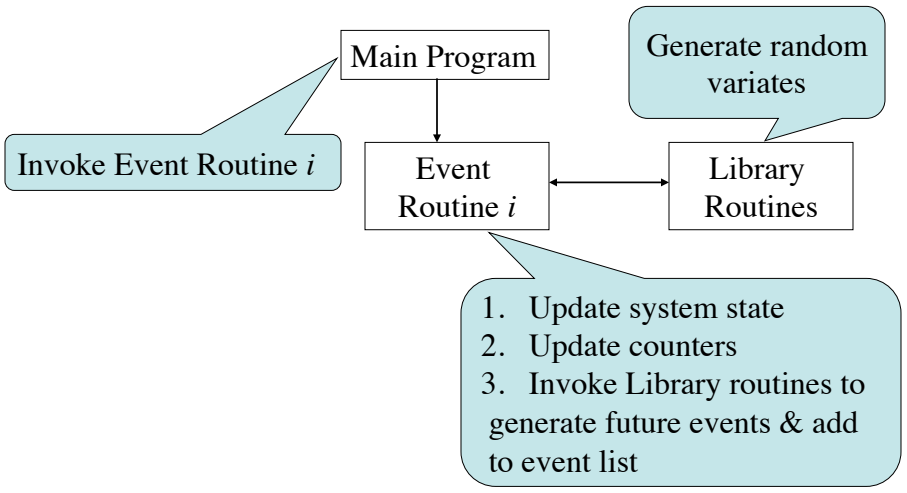
Flow of control for Next-Event



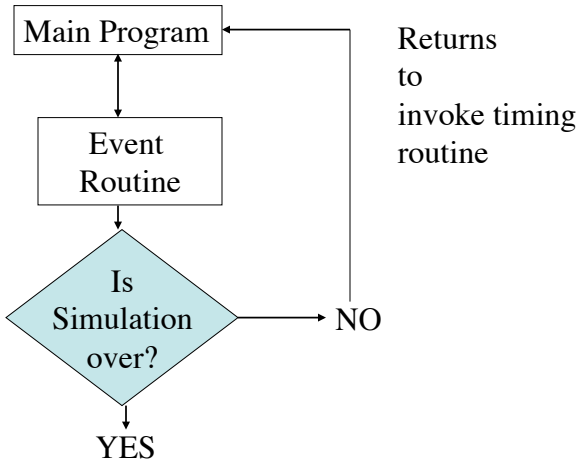
Flow of control for Next-Event



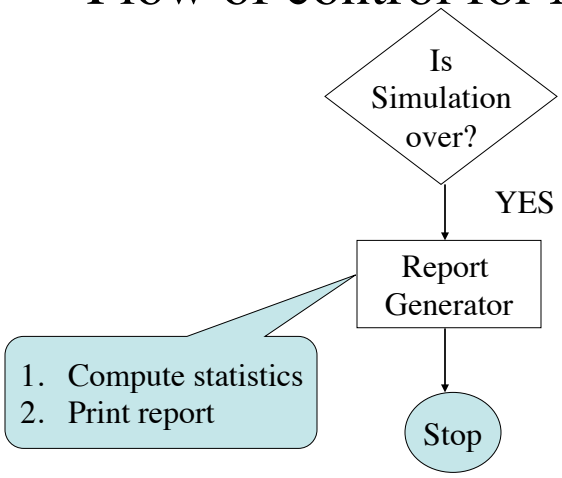
Flow of control for Next-Event



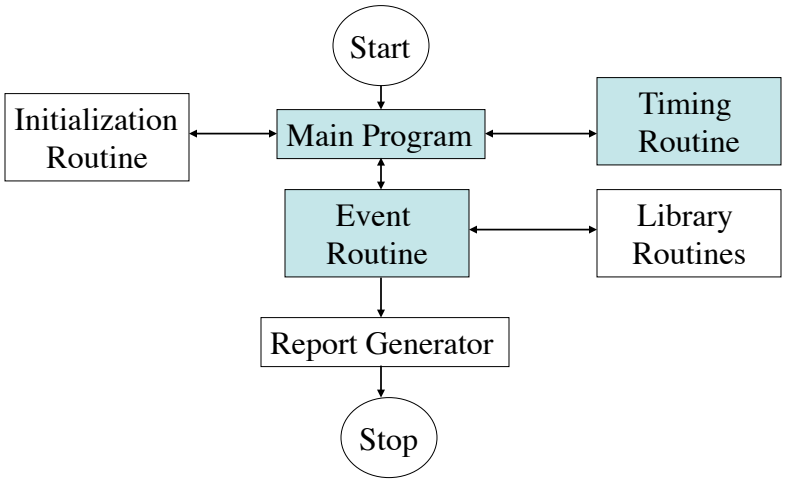
Flow of control for Next-Event



Flow of control for Next-Event



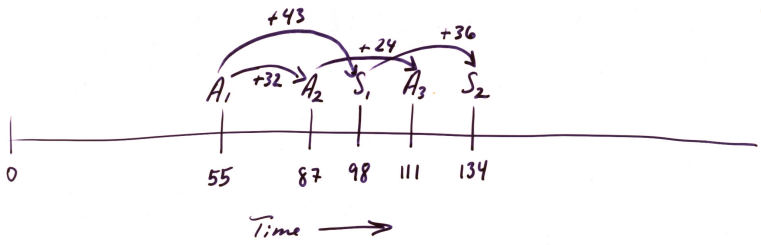
Flow of control for Next-Event where the action is



Major Observation

- Flow of control is driven by the events list
- Events are added to the list by the Event Routine generators for each event type *i*
- Timing routine takes events from the list

Example



Interarrival Times	Service Times
$A_1 = 55$	$S_1 = 43$
$A_2 = 32$	$S_2 = 36$
$A_3 = 24$	$S_3 = 34$

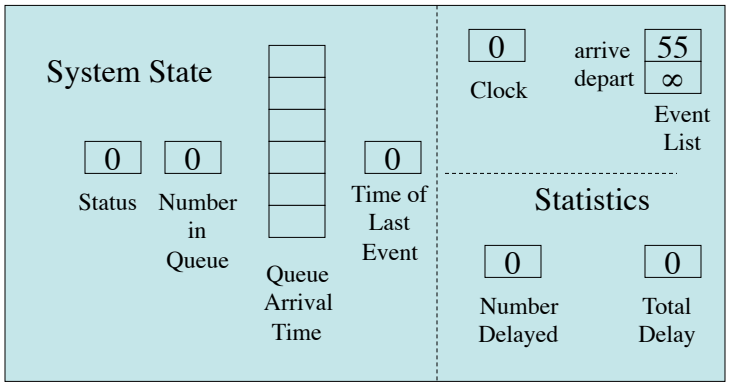
STEP 1

Interarrival Times
$A_1 = 55$
$A_2 = 32$ (clock = 87)
$A_3 = 24$ (clock = 111)

Service Times
$S_1 = 43$
$S_2 = 36$
$S_3 = 34$

Time = 0

A_1 to arrive at time 55



STEP 2

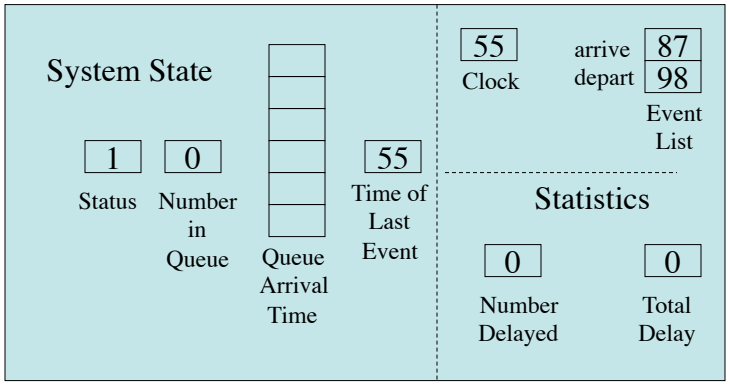
Interarrival Times
$A_1 = 55$
$A_2 = 32$ (clock = 87)
$A_3 = 24$ (clock = 111)

Service Times
$S_1 = 43$
$S_2 = 36$
$S_3 = 34$

Time = 55

A_1 in service & departs in 43 mins

A_2 to arrive in 32 mins



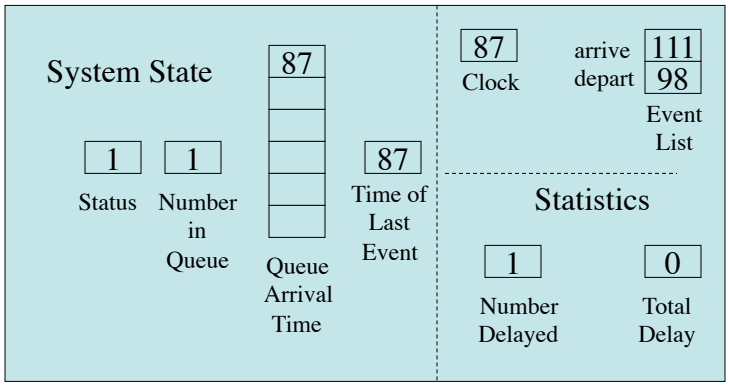
STEP 3

Interarrival Times
$A_1 = 55$
$A_2 = 32$ (clock = 87)
$A_3 = 24$ (clock = 111)

Service Times
$S_1=43$
$S_2=36$
$S_3=34$

Time = 87

A_1 in service
A_2 goes to queue
A_3 to arrive in 24 mins



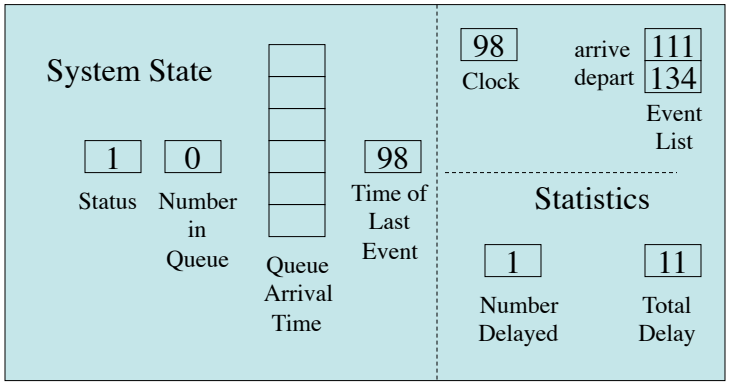
STEP 4

Interarrival Times
$A_1 = 55$
$A_2 = 32$ (clock = 87)
$A_3 = 24$ (clock = 111)

Service Times
$S_1=43$
$S_2=36$
$S_3=34$

Time = 98

A_1 departs
A_2 goes to service
A_3 to arrive in 13 mins



STEP 5

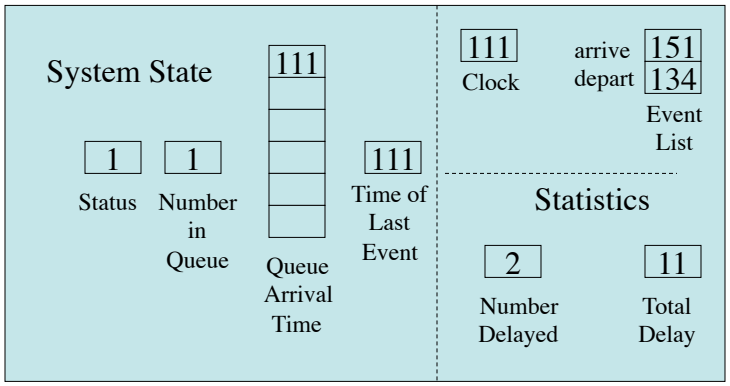
Interarrival Times
$A_1 = 55$
$A_2 = 32$ (clock = 87)
$A_3 = 24$ (clock = 111)

Service Times
$S_1=43$
$S_2=36$
$S_3=34$

Time = 111

A_2 still in service

A_3 arrives & goes to queue



STEP 6

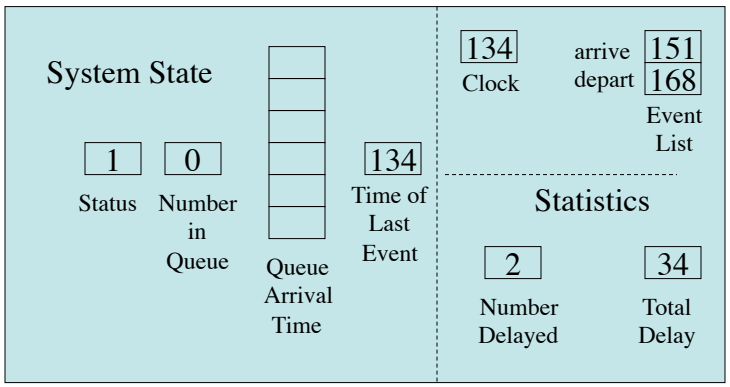
Interarrival Times
$A_1 = 55$
$A_2 = 32$ (clock = 87)
$A_3 = 24$ (clock = 111)

Service Times
$S_1=43$
$S_2=36$
$S_3=34$

Time = 134

A_2 departs

A_3 goes to service



Some additional comments on DES Modeling

- Real systems are composed of *entities*
- Barber shop
  - Real system entities are customers, barbers, etc.
  - DES model for the barber shop has entities that are events, queues and servers
  - Customers don't really exist!

Some additional comments on DES Modeling

- Real systems entities are described by *attributes*
- Barber shop
  - Real system attributes are types of customers such as male/female, etc.
  - DES model for the barber shop has attributes for the arrival-event of customers and for the queue itself such as number-in-queue
  - DES model will carry the customer attributes as part of the event record
  - *Attributes* are part of the system state

Issues in Programming

- Termination
  - Number of events processed OR
  - Overall duration
- Jockeying: changing queues
- Reneging: leaving queue without service
- Seasonal variations in behavior
  - Time series analysis
- Tie Breaking in event scheduling

Tie Breaking

- Two events scheduled at the same time
  - Choose event based on entity attributes
    - Example: take customer who is oldest
  - Choose event based on event type
    - Example: choose departure before arrival
  - FIFO
  - Specify state directly

Queues: more modeling concerns

- If you run to get in a short line, it will suddenly become a long line.
- While you are waiting in a long line, the people behind you will be directed to a new, short line.
- If you step out of a short line for a second, it will become a long line.
- If you are in a short line, the people in front of you will let their friends in and make it a long line.
- A short line outside a building becomes a long line inside.
- If you stand in one place long enough, you will make a line