General Purpose Simulation System (GPSS) (Part 7)

Dr. Çağatay ÜNDEĞER

Öğretim Görevlisi
Bilkent Üniversitesi Bilgisayar Mühendisliği Bölümü

e-mail: cagatay@undege.com
cagatay@cs.bilkent.edu.tr

Discrete Event Simulations (Outline)

- Simulation Languages
- GPSS Simulation Language
- A Sample Simulation Model
Simulation Language

• Describes the operation of a simulation on a computer.
• There are two major types of simulation:
  – Discrete-event and
  – Continuous simulations.
• Some modern languages can handle combinations.

Some Discrete-Event Simulation Languages

• GPSS
• AutoMod
• eM-Plant
• Rockwell Arena
• GASP
• SimPy
• SIMSCRIPT II.5
• Simula
• Poses++
Some Continuous Simulation Languages

- Advanced Continuous Simulation Language (ACSL):
  - Supports textual or graphical model specification.
- SimApp:
  - Simple simulation of dynamic systems and control systems.
- Simgua:
  - A simulation toolbox and environment that supports Visual Basic.
- VisSim:
  - A visually programmed block diagram language.

Some Hybrid Simulation Languages

- AnyLogic:
  - Multi-method simulation tool, which supports System dynamics, Discrete event simulation, Agent-based modeling.
- Simio:
  - A software for discrete-event, continuous, and agent-based simulation.
- Modelica:
  - An open-standard object-oriented language for modeling of complex physical systems.
- Saber Simulator:
  - Simulates physical effects in different engineering domains (hydraulic, electronic, mechanical, thermal, etc.).
General Purpose Simulation System (GPSS)

- A discrete time simulation language,
  - Where a simulation clock advances in discrete steps.
- A system is modelled as transactions (processes) that;
  - Enter the system and
  - Are passed from one service (represented by blocs) to another.

General Purpose Simulation System (GPSS)

- Can be applied effectively and easily to a broad range of problems.
- Particularly well suited for problems such as a factory optimization.
- There are many versions such as:
General Purpose Simulation System (GPSS)

- Features a software package that is designed to simplify matters for the user.
- User translates his problem into a conceptual model, which is a block diagram.
- Then GPSS software package:
  - Processes this block diagram,
  - Executes the simulation run, and
  - Produces statistics.

Block

- The basic structural element (statement) of the simulation language.
- A GPSS model is given by its block diagram.
- Some blocks:
  - GENERATE
  - TERMINATE
  - ASSIGN
  - SEIZE
  - RELEASE
  - QUEUE
  - DEPART
  - ADVANCE
  - START
  - END

A block diagram sample

- GENERATE X$INTMEAN,FN$EXP0
- ASSIGN 1,V$CALCWAIT
- RECALC ASSIGN 4,V$NORMALVAR
- TEST GE P4,5,RECALC
- TEST E F$COMPUTER1,1,LETSWAIT
- TEST E F$COMPUTER2,1,LETSWAIT
- TEST E F$COMPUTER3,1,LETSWAIT
- TEST E F$COMPUTER4,1,LETSWAIT
- TRANSFER V$DECIDE,IMMEQUIT
- LETSWAIT ENTER CHAIRS,1
- QUEUE WAITING
- TRANSFER .ENTCOM!
Transaction

• A process that represents the real-world system you are modeling.
• Executed by moving from block to block.
• Each transaction in the model is contained in exactly one block,
• But one block may contain many transactions.

<table>
<thead>
<tr>
<th>Line</th>
<th>Action</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>GENERATE</td>
<td>10,2</td>
</tr>
<tr>
<td>110</td>
<td>SIEZE</td>
<td>MACHA</td>
</tr>
<tr>
<td>120</td>
<td>ADVANCE</td>
<td>4</td>
</tr>
<tr>
<td>130</td>
<td>RELEASE</td>
<td>MACHA</td>
</tr>
<tr>
<td>140</td>
<td>TERMINATE</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>GENERATE</td>
<td>20,5</td>
</tr>
<tr>
<td>160</td>
<td>SIEZE</td>
<td>MACHB</td>
</tr>
<tr>
<td>170</td>
<td>ADVANCE</td>
<td>6</td>
</tr>
<tr>
<td>180</td>
<td>RELEASE</td>
<td>MACHB</td>
</tr>
<tr>
<td>190</td>
<td>TERMINATE</td>
<td></td>
</tr>
</tbody>
</table>

Operands of Blocks

• Some blocks with their operands
  – GENERATE A,B,C,D,E
  – SIEZE A
  – QUEUE A
  – .......
The Format of GPSS Computer Code

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Label</th>
<th>Operation</th>
<th>Operands</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 NEW</td>
<td></td>
<td>GENERATE</td>
<td>10</td>
<td>; comments</td>
</tr>
<tr>
<td>100 NEW</td>
<td></td>
<td>GENERATE</td>
<td>, , 10</td>
<td>; comments</td>
</tr>
</tbody>
</table>

Generating and Terminating Transactions

• GENERATE block generates transactions
• TERMINATE block destroys the transactions

GENERATE 300 ; creates a transaction every 300 time units

TERMINATE ; terminates the transaction
GENERATE A,B,C,D,E

• Generates transactions.

A: Mean interval between generation of two transactions
B: Half width of uniform distribution or function modifier used to generate random interval between generation of transactions
C: Delay starting time
D: Limit on total transactions to be created
E: Priority level

If only the A Operand is specified, it is evaluated numerically and used as the time increment.

GENERATE 300 ; interval times = 300
GENERATE A,B,C,D,E

• If B does not specify a function,
  – both A and B are evaluated numerically and a random number between A-B and A+B is used as the time increment.

GENERATE 300,100 ; interval times = [200,400]

GENERATE A,B,C,D,E

• When Operand B is an function,
  – It is a special case called a “function modifier”.
  – Time increment is calculated by multiplying the result of function by Operand A.

GENERATE 100, FN$EXPO
 ; interval times = exponential with mean 100
TERMINATE A

- Destroys the active transaction.

A: Termination Count decrement (optional).

- The Termination Count of the simulation is set to 1 by a prior START Command.
- When the termination count reaches 0, the simulation ends

TERMINATE ; transaction ends
TERMINATE 1 ; simulation ends

ADVANCE A,B

- Delays the progress of a transactions for a specified amount of simulated time.

A: The mean time increment.
B: Half width of uniform distribution or function modifier used to generate random delay

ADVANCE 100
ADVANCE 100, 50
ASSIGN A,B,C

- Used to place or modify a value in a Transaction Parameter (Local variable).
- If no such Parameter exists, it is created.

A: Parameter number of the Active Transaction. Operand must be Name, Positive Integer, etc....
B: Value assigned.
C: Function number.

ASSIGN 200, 3
ASSIGN 200+, 3
ASSIGN TEXT, "Sample text"
ASSIGN 1, V$COMPUTE_WAIT

SAVEVALUE A,B

- Changes the value of a Savevalue Entity (Global variable).

A: Savevalue Entity number. Operand must be Name, Positive Integer, etc....
B: Value to be stored.

SAVEVALUE 10, 5
SAVEVALUE CUSTOMERCOUNT+, 1
SAVEVALUE TEXT, "Sample text"
Using Facilities

• GPSS provides the facility modeling concept to represent limited availability of a service.
• A facility is a resource that can be used by only one transaction at a time.
• To request a facility transaction should enter in SEIZE block.
• Once a transaction has entered the SEIZE block, it owns the facility and other transactions are not allowed to enter this block.
• This is called blocking.

Using Facilities

• When a transaction no longer needs a facility,
  – It releases ownership of the facility, making it ready to be seized by other transactions.
• The RELEASE block is used for this purpose.

  SEIZE A ; gets the ownership of A

  ....

  ....

  RELEASE A ; release the ownership of A
SEIZE A

• Waits for or acquires ownership of a Facility Entity.

A: Facility name or number.

SEIZE Teller1
SEIZE Server1

RELEASE A

• Releases ownership of a facility.

A: Facility name or number.

RELEASE Teller1
RELEASE Server1
Blocking Sample

Only one transaction at a time is allowed to own the facility MECHANIC

GENERATE 300,20
SEIZE MECHANIC
ADVANCE 280,200
RELEASE MECHANIC
TERMINATE 1

Using Storages

• GPSS provides the storage modeling concept to represent a limited number of unit capacity.
• A storage is a resource that can be used by several transactions at a time until it becomes empty.
• To get units from a storage, transaction should call an ENTER block.
• To put units to a storage, transaction should call a LEAVE block.
**ENTER A,B**

- Attempts to enter an ENTER Block.
- Either takes or waits for a specified number of storage units.

**A:** Storage Entity name or number.
**B:** Number of units by which to decrease the available storage capacity.

Toolkit STORAGE 6
ENTER Toolkit, 2

* If there are not enough storage units remaining in the storage entity, Transaction comes to rest on the delay chain.

**LEAVE A,B**

- Increases the accessible storage units at a storage entity.

**A:** Storage Entity name or number.
**B:** Number of units by which to increase the available storage capacity.

LEAVE Toolkit, 3

* When storage becomes available, delay chain of the storage entity is examined in decreasing priority for transactions whose demands can now be met.
Collecting Time Statistics

- To collect data about how long it takes transactions to traverse a given segment of model,
  - Mark;
    - The beginning of the segment with a QUEUE block and
    - The end with a DEPART block.

```
QUEUE A ; A is name of queue
...
...
DEPART A ; A is name of queue
```

QUEUE A,B

- Updates queue entity statistics to reflect an increase in content.
  A: Queue Entity name or number.
  B: Number of units by which to increase the content of the Queue Entity. Default value is 1.

```
QUEUE WaitingLine, 1
* The content of the Queue Entity named WaitingLine is increased by one and the associated statistics accumulators are updated.
```
DEPART A,B

- Register statistics, which indicates a reduction in the content of a Queue Entity.

A: Queue Entity name or number.
B: Number of units by which to decrease the content of the Queue Entity. Default value is 1.

DEPART WaitingLine, 1

* The content of the Queue Entity named WaitingLine is reduced by one and the associated statistics accumulators are updated.

Queue-Depart Sample

GENERATE 300,20
QUEUE GARAGE ;Store time trans. entered
SEIZE MECHANIC
DEPART GARAGE ;Store time trans. left
ADVANCE 280,200
RELEASE MECHANIC
TERMINATE 1

* When this program is run, GPSS collects data about the amount of time each transaction spends waiting for MECHANIC
Branching with Transfer

- To branch on random basis,
  - use TRANSFER block.

```
TRANSFER   P, LabelA, LabelB
....
LabelA ....
....
LabelB ....
....
```

Jumps to location LabelB with probability P,
Jumps to LabelA with probability 1 - P.

TRANSFER A,B,C,D

- Causes the Active Transaction to jump to a new Block location.

A: Transfer Block mode. Optional. The operand must be BOTH, ALL, PICK, FN, P, SBR, SIM, fraction, Name, Positive Integer, etc.
  The meaning of operands B and C depend on the mode.
  When Operand A is omitted,
    TRANSFER Block operates in “Unconditional Mode”, and
    Transaction always jumps to the location specified by B.

B: Block number or location. Parameter name or number when in P Mode.

C: Block number or location. Increment value in FN or P Mode.

D: Block number increment for ALL Mode. Default is 1.
TRANSFER Samples

; Jump to Place unconditionally.
TRANSFER , Place

; Jump to Place with 75% probability, and
; Continue next block with 25% probability.
TRANSFER 0.75, , Place
.... ; next block

; Jump to Place2 with probability in variable V$DECIDE, and
; Jump to Place1 with 1-probability in variable V$DECIDE.
TRANSFER V$DECIDE, Place1, Place2

Branching with Test

• To branch on some condition of the system,
  – Use TEST block.

    TEST GE C1, 10, ElseLabel
    ....
    ....
    ElseLabel ....
    ....
    
    Transaction continues to next sequential program block if test is successful,
    Or jumps to location ElseLabel if test specified is unsuccessful.
    Transaction is blocked if test is unsuccessful and label is omitted.

    TEST L Q$GARAGE, 4
    Continue to next block only if queue GARAGE has less than 4 entries, otherwise transaction is blocked (not allowed to leave TEST block).
TEST O A,B,C

- Compares values, and controls the destination of the active transaction based on the result of the comparison.

O: Relational operator. Relationship of Operand A to Operand B for a successful test. The operator must be E, G, GE, L, LE, or NE.

A: Test value.

B: Reference value that A is going to be compared.

C: Destination Block number.

Transaction continues to next sequential program block if test is successful, or jumps to location indicated by C if test is unsuccessful.

If test is unsuccessful and label is omitted, transaction is blocked until the test becomes successful.

Branching Sample

Program segment for a queuing system with limited rework time. 5% of repairs must be reworked.

```
GENERATE 300.20
QUEUE REPAIRSYS
INPUTQ SEIZE REPAIRMAN
ADVANCE 270.30
RELEASE REPAIRMAN
TRANSFER 0.95, , GOOD ; 5% are reworked
TEST GE M1, 700, INPUTQ ; Test for time in process.
DEPART REPAIRSYS ; Remove twice repaired and still bad parts.
TERMINATE
GOOD DEPART REPAIRSYS
TERMINATE 1
```

The TEST block only sends bad parts that have been in system for less than 700 time units to the repair queue.

M1: Indicates how much time has passed since a transaction was generated.
Some of The System Variables

- **Blocks:**
  - N$label: Count of transactions entered the block
  - W$label: Current # of transactions in the block
- **Facilities:**
  - F$facility: Current content (0 or 1)
- **Queues:**
  - Q$queue: Current content
  - QM$queue: Maximum content so far
- **Transactions:**
  - M1: Time since current transaction was created
  - PR: Priority of transaction
- **Storages:**
  - R$storage: Available storage capacity
  - S$storage: Amount of storage currently in use

If Queue is long, refuse to enter

- GENERATE 500,50
- TEST LE Q$waiting, 10, REFUSE
- SEIZE waiting
- ADVANCE 200,10
- DEPART waiting
- REFUSE TERMINATE 1
Random Numbers

- RN_i is the index of a random number stream.
- Each RN_i is independent of each other.
- Value of RN is between 0..9999
- If RN is used with a FUNCTION, its value is between 0..1

Random Queue Processing

GENERATE 200,50
PRIORITY RN1 ; assigns a random priority to active transaction
SEIZE RANQ
ADVANCE 300
DEPART RANQ
TERMINATE 1
Probability Distribution Functions

- FUNCTION statement is used to specify other probability distributions.

```latex
\begin{align*}
\text{Label} & \quad \text{Random Number} & \quad \text{C for Continuous function} \\
\text{Stream number 1} & \quad \text{(function argument)} & \quad \text{using 6 points (D for discrete functions)} \\
\text{EXPO FUNCTION RN1,C6} & \quad 0, 0 / .359,.445 / .638,1.016 / .838,1.82 / .958,3.164 / .999,7.559 \\
\text{f(x)} & \quad x & \quad \text{Delimiter}
\end{align*}
```

Graphics Representation

EXPO FUNCTION RN1,C6
0.0/.359,.445/638,1.016/.838,1.82/.958,3.164/.999,7.559

Cumulative distribution function
Discrete Functions

<table>
<thead>
<tr>
<th>Value</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prob.</td>
<td>10%</td>
<td>40%</td>
<td>30%</td>
<td>20%</td>
</tr>
</tbody>
</table>

D for Discrete Functions

```plaintext
EXPO FUNCTION RN1,D6
.1,4 / .5,5 / .8,6 / 1.0,7
```

Savevalues

- Using global variables
- INITIAL X$name, initialvalue
- SAVEVALUE name, value
- SAVEVALUE name+, value
- SAVEVALUE name-, value
Variables

• Savevalues can only be set, increased or decreased.
• For complex functions, variables are used.

label VARIABLE function
   ( such as 5+Q$queue )

Exception : # is used for multiplication

A Sample Simulation Model

• Problem: An internet cafe optimization

• Internet cafe having 4 computers and 6 waiting chairs.
• Customers arrive at the cafe in some intervals depending on the time of day.
A Sample Simulation Model

• When a new customer arrives and there is any free computer,
  – He immediately starts to use one of the free computers and leave after some time.
• If there are not any free computers, but there are free chairs,
  – He may decide to wait for a limited period of time or immediately leave the cafe.
• If there are also no free chairs,
  – He immediately leaves without waiting.

A Sample Simulation Model

• If a customer uses a computer,
  – The price starts with 4 TL (opening price) and
  – Increases 1 TL for each 10 minutes of time (period of time).
Objective

• The Internet cafe owner wants to increase the profit of the cafe within a day by;
  – Either increasing the “opening price” or
  – Increasing the “period of time” without changing the cost of 1 hour for a customer (10 TL).

Objective

• The Internet cafe owner;
  – Wants to increase the profit of the cafe within a day by;
    • Either increasing the “opening price” or
    • Increasing the “period of time” without changing the cost of 1 hour for a customer (10 TL).
  – Also wonders how much the profit may increase if one more computer is added to the Internet cafe.
Objective

• The Internet cafe owner;
  – Wants to increase the profit of the cafe within a day by;
    • Either increasing the “opening price” or
    • Increasing the “period of time” without changing the cost of 1 hour for a customer (10 TL).
  – Also wonders how much the profit may increase if one more computer is added to the Internet cafe.

Solution

• One day of the internet cafe will be simulated by GPSS World (Student Addition), and
• The results will be analyzed.
GPSS World

• A Minuteman Software product.
Problem Details

• Internet cafe is opened at 9:00 in the morning and closed at 1:00 at night.
• The mean of inter-arrival times of customers;
  – Changes in various time of day,
  – Is usually less in the afternoon and evening, high in the morning and late at night.

Problem Details

• The mean service time is usually about 1.5 hours.
• In the morning and late at night, the mean reduces to 1 hour.
Problem Details

- There are totally 6 chairs.
- If all the chairs are full, the customer immediately leaves.
- A too much waited customer may decide to leave the cafe without using a computer.
- Sometimes customers do not wait although there are still free chairs.
- The probability of immediately leaving the cafe is proportional to the number of customers waiting in the queue.

Inter Arrival Times

- Inter-arrival time distribution is assumed to be exponential about the mean.
- Mean changes in different times of day in a discrete manner.

<table>
<thead>
<tr>
<th>Time Period</th>
<th>Inter-arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 - 10:00</td>
<td>40 minutes</td>
</tr>
<tr>
<td>10:00 - 11:00</td>
<td>45 minutes</td>
</tr>
<tr>
<td>11:00 - 12:00</td>
<td>50 minutes</td>
</tr>
<tr>
<td>12:00 - 13:00</td>
<td>25 minutes</td>
</tr>
<tr>
<td>13:00 - 14:00</td>
<td>20 minutes</td>
</tr>
<tr>
<td>14:00 - 15:00</td>
<td>20 minutes</td>
</tr>
<tr>
<td>15:00 - 16:00</td>
<td>40 minutes</td>
</tr>
<tr>
<td>16:00 - 18:00</td>
<td>60 minutes</td>
</tr>
</tbody>
</table>
Inter Arrival Times

```
INITIAL X$INTMEAN 60
....
GENERATE X$INTMEAN,FN$EXPO TIME = X$INTMEAN x FN$EXPO
....
EXPO FUNCTION RN1,C24 : Exponential Distribution
0.0/.1.0.104/.2.222/.3.355/.4.509/.5.69/.6.915/.7.1.2/.751.38/.8,1.6/.84
,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9/,99,
4.6/.995,5.3/.998,6.2/.999,7.9997,8
```

A random number is taken with exponential distribution

Service Times

- There are 4 computers in the cafe.
- Because the customers give 4 TL in advance,
  - They don’t usually leave early and
- The mean computer usage time;
  - Is about 1.5 hours in daytime, and
  - Reduces to 1 hour early in the morning and late at night.
- Service time distribution is assumed to be normal around the mean
  - With standard deviation = 80% of the mean.
Service Times

INITIAL X$COMMEAN 60
NORMALVAR VARIABLE X$COMMEAN+(X$COMMEAN#0.80)#FN$SNO RM
RECALC GENERATE X$INTMEAN,FN$EXPO ASSGN 4,V$NORMALVAR
TEST GE P4,5,RECALC
SEIZE COMPUTER1 ; Customer starts using computer
ADVANCE P4 ; Use computer sometime
RELEASE COMPUTER1 ; Customer leaves

SNORM FUNCTION RN4,C25 ; Standard normal dist. function
0,-5/.00003,-4/.00135,-3/.00621,-2.5/.02275,-2.06681,-1.5/.06681,-1.2/.11507,-1/.21186,-.8/.27425,-.6/.34458,-.4/.42074,-
-.2/.50002,.2/.57926,.4/.72575,.6/.78814,.8/.84134,1/.88493,1.2/
.93319,1.5/.97725,2/.99379,2.5/.99865,3/.99997,4/1,5

Loops until service time ≥ 5

Too Much Waited Customers

- After entering a queue by sitting on a chair,
  – the customer waits for a free computer.
- But if waiting time becomes so long,
  – Customer usually leaves the cafe.
- Mean waiting time for a customer is assumed to be 30 minutes.
- The distribution is assumed to be normal around the mean
  – With standard deviation = 5 minutes.
Too Much Waited Customers

<table>
<thead>
<tr>
<th>CHAIRS</th>
<th>STORAGE</th>
<th>VARIABLE</th>
<th>CALCWAIT</th>
<th>STORAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>X$WAITMEAN+5#FN$WAITTIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,V$CALCWAIT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GENERATE X$INTMEAN, FN$EXPO

ASSIGN 1, V$CALCWAIT ; max waiting time

LETSWAIT ENTER CHAIRS, 1

TRANSFER , ENTCOM1

WAIT

ADVANCE 1

TEST LE M1,P1,QUIT ; Enough wait?

QUIT LEAVE CHAIRS, 1

TERMINATE

ENTCOM1 TEST E F$COMPUTER1, 0, ENTCOM2

ENTCOM2 TEST E F$COMPUTER2, 0, ENTCOM3

ENTCOM3 TEST E F$COMPUTER3, 0, ENTCOM4

ENTCOM4 TEST E F$COMPUTER4, 0, WAIT

* M1 = Time since current transaction was created

Balking

- Many customers immediately leave the cafe without entering the queue (balking).
- There are 6 chairs to sit.
- So at most 6 customers can wait.

- Probability to balk = 0.01 + 0.0275 x Waitings^2

  * Waitings = number of customers waiting in the queue
  * If Waitings = 0, Probability to balk = 0.01
  * If Waitings = 6, Probability to balk = 1.00
### Balking

<table>
<thead>
<tr>
<th>DECIDE VARIABLE</th>
<th>0.01+0.0275#Q$WAITING#Q$WAITING</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATE</td>
<td>X$INTMEAN,F$SEXPO</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>1,V$CALCWAIT</td>
</tr>
<tr>
<td>RECALC ASSIGN</td>
<td>4,V$NORMALVAR</td>
</tr>
<tr>
<td>TEST GE</td>
<td>P4,5,RECALC</td>
</tr>
<tr>
<td>TEST E</td>
<td>F$COMPUTER1,1,LETSWAIT</td>
</tr>
<tr>
<td>TEST E</td>
<td>F$COMPUTER2,1,LETSWAIT</td>
</tr>
<tr>
<td>TEST E</td>
<td>F$COMPUTER3,1,LETSWAIT</td>
</tr>
<tr>
<td>TEST E</td>
<td>F$COMPUTER4,1,LETSWAIT</td>
</tr>
<tr>
<td>TRANSFER</td>
<td>V$DECIDE,IMQUIT</td>
</tr>
<tr>
<td>IMQUIT TERMINATE</td>
<td></td>
</tr>
<tr>
<td>LETSWAIT ENTER</td>
<td>CHAIRS,1</td>
</tr>
<tr>
<td>QUEUE WAITING</td>
<td></td>
</tr>
<tr>
<td>DEPART WAITING</td>
<td></td>
</tr>
<tr>
<td>LEAVE CHAIRS,1</td>
<td></td>
</tr>
</tbody>
</table>

### Calculating Bill

**General Formula:**

\[
\text{Bill} = \text{opening price} + \text{payment increase} \times \text{INT(usage time/period of time)}
\]

**Formula for Default:**

\[
\text{Bill} = 4 + 1 \times \text{INT(usage time/10)}
\]
Simulating Change in One Day

- A generate command is placed at the end of the program,
- Only generates one transaction in each day (within working hours).
- At the end of day, transaction is terminated decreasing the simulation counter by 1 (TERMINATE 1).

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONEDAYGENERATE</td>
<td>Internet Cafe Open, 16°60 Min = 1 Day</td>
</tr>
<tr>
<td>SAVEVALUE</td>
<td>INTMEAN,60; After 09:00 Inter Arrival Mean = 60 min</td>
</tr>
<tr>
<td>SAVEVALUE</td>
<td>COMMEAN,60; Computer Usage Mean = 60 min</td>
</tr>
<tr>
<td>ADVANCE</td>
<td>60; 1 Hours</td>
</tr>
<tr>
<td>SAVEVALUE</td>
<td>INTMEAN,45; After 10:00 Inter Arrival Mean = 45 min</td>
</tr>
<tr>
<td>ADVANCE</td>
<td>60; 1 Hours</td>
</tr>
<tr>
<td>SAVEVALUE</td>
<td>INTMEAN,30; After 11:00 Inter Arrival Mean = 30 min</td>
</tr>
<tr>
<td>ADVANCE</td>
<td>180; 3 Hours</td>
</tr>
<tr>
<td>SAVEVALUE</td>
<td>INTMEAN,25; After 14:00 Inter Arrival Mean = 25 min</td>
</tr>
<tr>
<td>ADVANCE</td>
<td>240; 4 Hours</td>
</tr>
<tr>
<td>SAVEVALUE</td>
<td>INTMEAN,20; After 16:00 Inter Arrival Mean = 20 min</td>
</tr>
<tr>
<td>ADVANCE</td>
<td>180; 3 Hours</td>
</tr>
<tr>
<td>SAVEVALUE</td>
<td>INTMEAN,30; After 21:00 Inter Arrival Mean = 30 min</td>
</tr>
<tr>
<td>ADVANCE</td>
<td>120; 2 Hours</td>
</tr>
<tr>
<td>SAVEVALUE</td>
<td>INTMEAN,40; After 23:00 Inter Arrival Mean = 40 min</td>
</tr>
<tr>
<td>ADVANCE</td>
<td>60; 1 Hours</td>
</tr>
<tr>
<td>SAVEVALUE</td>
<td>COMMEAN,60; Computer Usage Mean = 60 min</td>
</tr>
<tr>
<td>ADVANCE</td>
<td>60; 1 Hours</td>
</tr>
<tr>
<td>TERMINATE</td>
<td>1; Internet Cafe Closed At 01:00</td>
</tr>
</tbody>
</table>
Analyzing Results

• After running the simulation for 1 day,
  – Statistics of queue, facilities, balking and profit are examined.
• The total profit was 341 TL.
• 116 TL was from opening price, and
• 225 TL was from Internet usage time.

• Changed;
  – Opening price to 3 TL and
  – Period of time to 8.5 minutes.
• The cost of 1 hour didn’t change.
• The total profit increased from 341 TL to 354 TL.
Analyzing Results

• Changed;
  – Opening price to 5 TL and
  – Period of time to 12 minutes.
• The cost of 1 hour didn’t change again.
• The total profit increased from 341 TL to 331 TL.

• Decreasing the opening price increases the profit if the average service time remains the same (keep some opening price).

Analyzing Results

• Last attempt was to increase the number of computers by 1.
• The profit was 348 TL (7 TL increase).
• For one month,
  – Increase will be at most 210 TL.
• One computer costs about 1500 TL.
• The cafe has to work about 7 months to gain from 1 additional computer.
• Seems not worthed?