UNIT-8
Simulation of Computer Systems

⊛. Simulation tools:

Simulation tool is a software which is used for simulating hardware, software and network. It is a type of software based on the process of modeling a real world phenomenon with mathematical formula. This program allows the user to observe the operation without actually performing that operation. Its main importance is: how it actually supports the model building. The tools commonly used for simulation are:

→ Processor simulation
→ Memory simulation
→ ALU simulation
→ Logic Network simulation
→ CPU Network simulation

⊛. Continuous System Modeling Program (CSMP): [Imp]

Continuous System Modeling Program (CSMP) is an early scientific computer software designed for modeling and solving differential equations numerically. This enables real-world systems to be simulated and tested with a computer.

Types of statements in CSMP:

1) Structural statements: Structural statements define the model. They consist of FORTRAN like statement and functional block designed for operations that frequently occur in a model definition. Structural statement can make use of the operation of addition, subtraction, multiplication, division and exponentiation, using the same notation and rule as used in FORTRAN. If the model include the equation $X = \frac{6Y}{W} + (Z-2)^2$. Then, the following statement would be used:

```
X=6.0*Y/W+(Z-2)**2.0
```

2) Data statements: Data statements assign numerical value to parameters, constant and initial conditions. For example one data statement called INCON can be used to set the initial value of integration function block.

3) Control statements: Control statements specify options in the assembly and execution of program and choice of inputs. For example; if printed output is required, control statements with PRINT and PRDEL are used followed by the names of variables to form the outputs.

⊛ General Purpose Simulation System (GPSS):

GPSS is a discrete time simulation general-purpose programming language where a simulation clock advances in discrete steps. A system is modeled as transactions enter the system and are passed from one service to another. This is particularly well suited for problems such as factory. GPSS is less flexible than simulation languages such as Simula and SIMSCRIPT but it is easier to use and most popular.

GPSS was designed especially for analysis who were not necessarily computer programmer. It is particularly suited for modeling traffic and queuing systems. A GPSS programmer does not write a program in some sense as SIMSCRIPT programmer does, instead he constructs a block diagram which is a network of interconnected blocks each performing a special simulation oriented function.

GPSS provides a set of 48 different blocks to choose from each of which can be used repeatedly. Each blocks has a name and specific task to perform. Moving through the system of block are entities called transaction are customer, messenger, machine etc.

A transaction is a GPSS object with a number of attributes. A transaction is like a customer entering into the process for service. A single transaction may represent several individual entities. Each transaction has to be generated either one at a time or in batches. Once they appear into the system they must be contained exactly in one action block. However a block may contain many transactions.

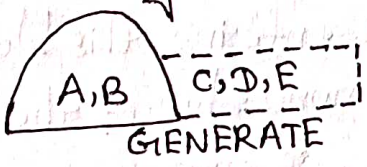
⊛. GPSS Blocks:

1) Generate Block: This block will produce a flow of transactions with inter-arrival times determined by the attributes value. The label is optional. The distribution of inter-arrival times follows a uniform probability distribution.

Syntax: GENERATE A,B,C,D,E

where, A = average value of uniform distribution
B = half-width of uniform distribution
C = time delay before first transaction is generated.
D = maximum number of transactions generated.
E = Priority allocated to transaction.

Symbol:



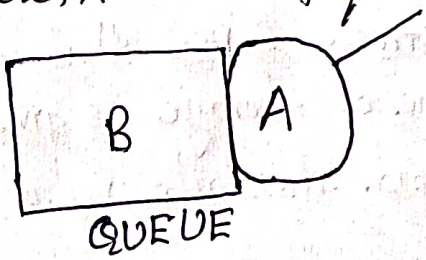
Example,
GENERATE 18,6.

2) QUEUE Block: This block will instruct GPSS to start gathering queuing statistics on the queue named in its attribute value. This label is optional but may be necessary if you have to refer to this line from somewhere else in the program.

Syntax: QUEUE A

where, A = name of queue.

Symbol:



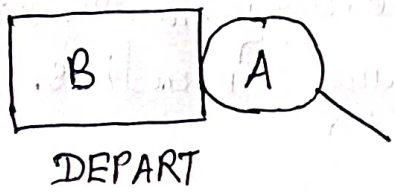
Example;
QUEUE Barber

3) **DEPART Block:** This block instruct GPSS that a transaction is leaving the queue named in its attribute value. This is necessary in order to compile the statistics on queue. The label is optional.

Syntax: DEPART A

where, A = name of the queue.

Symbol:



Example:

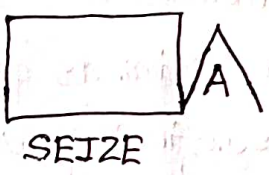
DEPART Barber.

4) **SEIZE Block:** This block allows the transaction to seize a facility if it is free. The label is optional. It may be a car "seizing" a "facility" such as a petrol pump or customer in supermarket "seizing" a facility such as the checkout assistant. When the car or customer is being serviced by facility it is said to "own the facility."

Syntax: SEIZE A

where, A = name of facility.

Symbol:



Example:

SEIZE Barber

5) **RELEASE Block:** A transaction entering this block inform GPSS that it is giving up the ownership of the facility named in its attribute value. This label is optional.

Syntax: RELEASE A

where, A = name of facility.

Symbol:

Example:

RELEASE Barber.

6) **Enter Block:** This block instructs GPSS that a transaction has entered storage. The name of storage is given by the first attribute value. The second attribute value gives the

the amount of storage incremented, when the transaction enters the ENTER block. A storage must be declared at beginning of program.

Syntax: ENTER A,B

where, A = name of storage
B = increment storage by this value.

Symbol:

Example:

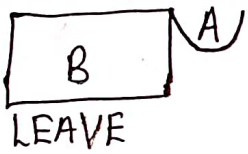
ENTER warehouse, 25.

7) LEAVE Block: This block instructs GPSS that a transaction is leaving a storage. The first attribute gives the name of the storage and second attribute decrements the storage by the value of the attribute.

Syntax: LEAVE A,B

where, A = name of storage
B = decrement storage by this value.

Symbol:



Example:

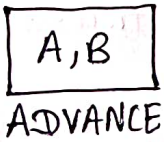
LEAVE warehouse, 20

8) ADVANCE Block: This block represents the servicing of a transaction. The servicing time follow a uniform probability distribution. The label is optional.

Syntax: ADVANCE A,B

where, A = average value of uniform distribution.
B = half-width of uniform distribution.

Symbol:

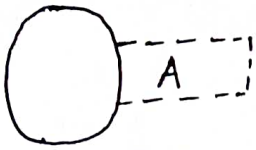


9) TERMINATE Block: This block destroys any transaction entering it and removes it from computer memory. Each time a transaction enters this block it decrements a computer by an amount equal to its attribute value. The counter is set by the user upon starting the simulation.

Syntax: TERMINATE A

where, A = decrements simulation counter by this amount.

Symbol:



TERMINATE

10> TEST Block: This block can test the logical condition of a queue or storage according to a particular refrence value. If a transaction enters the block it will check this condition and if it is true, it will send the transaction to one destination in the program and if the condition is false, it will send to another.

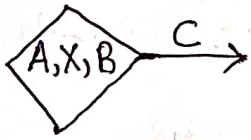
Syntax: TEST A,B,C

where, A = value and name of block being refrenced.

B = refrence value.

C = destination for the transaction if logical condition is not satisfied.

Symbol:



11> TRANSFER Block: This block will take transactions entering it and transfer them to each of two different destinations according to laid down properties.

Syntax: TRANSFER A,B,C

where, A = probability value (0 to 1)

B = proportion of (1-A) transactions transferred to this labelled location.

C = proportion A transactions transferred to this labelled location.

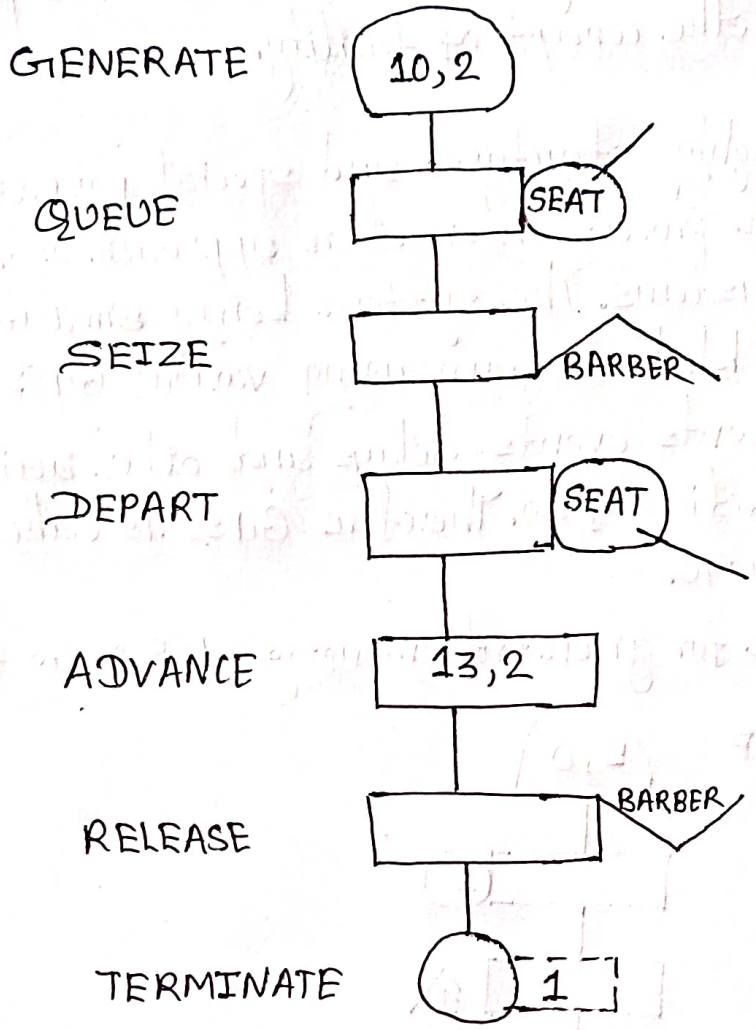
Symbol:

[Imp]

Q1. Create a GPSS model and program to simulate a barber shop for a day (9 am to 4 pm) where a customer enters the shop every 10 ± 2 minutes and a barber takes 13 ± 2 for a haircut.

Solution:

GPSS model to simulate a barber shop



Program

```
GENERATE   10,2
QUEUE      SEAT
SEIZE      BARBER
DEPART     SEAT
ADVANCE    15,3
RELEASE    BARBER
TERMINATE  1
```

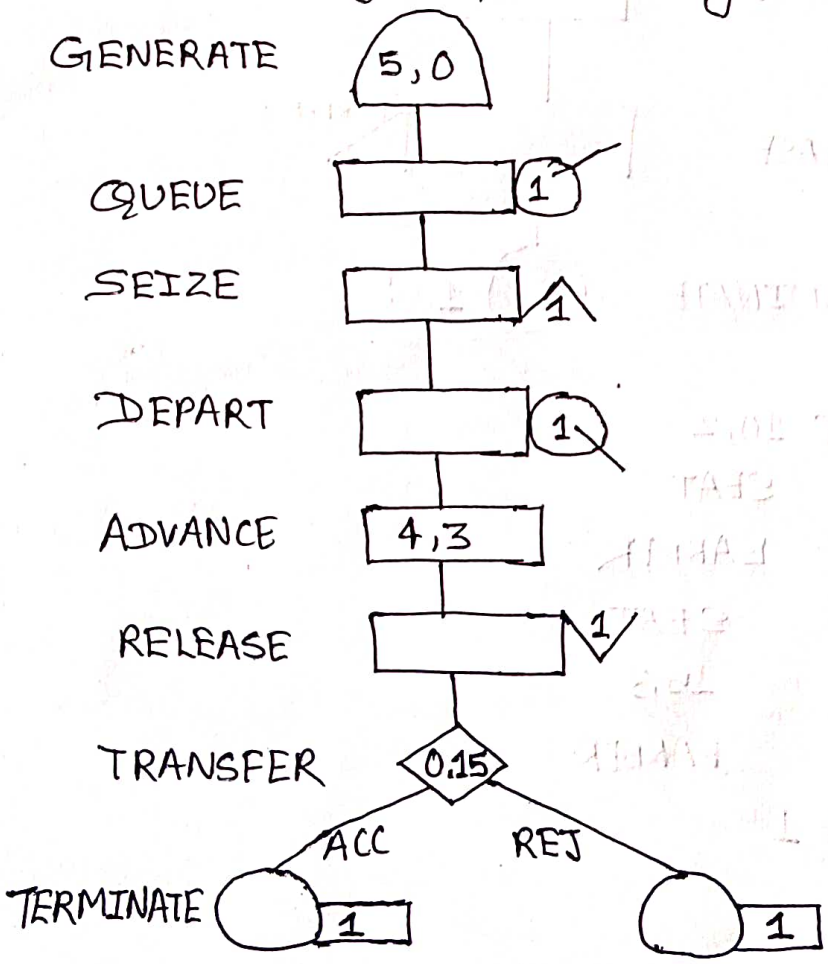
[Imp]

Q. Why is GPSS called transaction flow oriented language? A machine tool in a manufacturing shop is turning out parts at the rate of every 5 minutes. When they are finished, the parts are sent to an inspector, who takes 4±3 minutes to examine each one and rejects 15% of the parts. Draw and explain a block diagram and write a GPSS program to simulate using the concept of facility.

Solution:

GPSS is a highly structured and special purpose simulation language based on process interaction approach and oriented toward queuing systems. The system being simulated is described by the block diagram using various GPSS blocks. Each block represents events, delays and other actions that affect the transaction flow. Therefore GPSS is called transaction flow oriented language.

The block diagram for given problem using GPSS is as follows:



Explanation: A GENERATE block is used to represent the output of the machine by creating one transaction every five units of time. A QUEUE block places the transaction in the queue and SEIZE block allows a transaction to engage a facility if it is available. If more than one inspector is available, the transaction leaves the queue which is denoted by DEPART block and enters into ADVANCE block. An ADVANCE block with a mean of 4 and modifier of 3 is used to represent inspection. The time spent on inspection will therefore be any one of the values 1,2,3,4,5,6, or 7, with equal probability given to each value. Upon completion of the inspection, RELEASE block allows a transaction to disengage the facility and transaction go to a TRANSFER block with a selection factor of 0.15, so that 85% of the parts go to the next location called ACC, to represent accepted parts and 15% go to another next location called REJ to represent rejects. Both locations reached from the TRANSFER block are TERMINATE blocks.

Program:

```
GENERATE  5,0
 QUEUE    1
 SEIZE    1
DEPART    1
ADVANCE   4,3
RELEASE   1
TRANSFER  0.15 ACC REJ
ACC TERMINATE 1
REJ TERMINATE 1.
```

Q. Draw and describe the different types of GPSS blocks that are used to gather statistics?

Ans:- Describe: GENERATE block, QUEUE block, DEPART block, SEIZE block, RELEASE block, ADVANCE block and TERMINATE block in short that we studied earlier in GPSS Blocks.

Q. Draw and describe different types of GPSS blocks used to deal with queues?

Ans: Describe: GENERATE block, QUEUE block, DEPART block, ADVANCE block and TERMINATE block in short that we studied earlier in GPSS blocks.

If my notes really helped you,then you can support me on esewa for my hardwork.

Esewa ID: 9806470952