หนังสืออ้างอิง

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

1. ฐิติ เลิศสัจจาฐาณ รายงานการศึกษาเกี่ยวกับการจำหน่ายตั๋วล่วงหน้า. ฝ่ายการเดินรถ การรถไฟแห่งประเทศไทย

2. Karlin Samuel, A First Course in Stachatic Process. 2nd ed New York : Academic Press, 1969

3. T.H.Nylor, J.L.Balinfy, D.H.Burdich, K.Chu. Computer Simulation Technique. New York : John Willey & Sons, 1966

4. Robert J.Thierauf., and Richard A.Grosses. Decision Making Through Operation Research. New York : John Willey & Sons, 1970

5. K.M.Gwilliam., and P.J.Mackie. Economic and Transport Policy. George Allen & Unwin Ltd., 1975

6. Thomas L.Saaty. Elements of Queueing Theory with Applications. New York : Mc Graw Hill, 1961

7. Joe H.Mize., and J.Grady Case. Essential of Simulation. New Jersey : Englewood Cliffs, Prentice Hall, 1968

8. Donald Gross., and Carl M.Harris. Fundamentals of Queueing Theory. New York : John Willey & Sons, Inc., 1974

9. Russel L.Ackoff., and Mourice W.Sasieni. Fundamentals of Operations Research. New York : John Willey & Sons, inc., 1968

10. J.M.Hammersley., and D.C.Handcomb. Monte Carlo Method. London : Melhven & Co., 1964

11. Mc clain. and Thomas. Operation Management. New Jersey : Prentice Hall, 1980

12. Mourice Sasieni, Arther Yaspan., and Lawrence Friedman. Operation Research Methods and Problems. New York : John Willey & Sons, Inc., 1969

13. Hamdy A.Taha. Operations Research An Introduction. London : Collier Mac Millan Publishers

14. Harvey M.Wagner. Principles of Operation Research with Applications to Managerial. New Jersey : Prentice Hall, Eaglewood Cliffs, 1969

15. E.Page. Queueing Theory in Operations Research. Butterworth Group, 1972

16. Tom Caso. Statistics Methods in Management. London : Cassel, 1974

17. Geoffrey Gordon. System Simulation. New Jersey : Englewood Cliffs, Prentice Hall, 1969

18. Claud Mc Millan., and Richard F.Gonzalez. System Analysis. Illinois., Richard D.Irwin, 1965

19. Meier., Newell., and Pazer. Simulation in Business and Economic. New Jersey : Prentice Hall Inc., Englewood Cliffs, 1969

20. Deo Narsingh. System Simulation with Digital Computer. New Delhi : Prentice Hall of India Private Limited, 1979

21. J.Schriber Thomas. Simulation Using GPSS. New York : John Willey & Sons, 1974

22. K.D.Tacher., Hodder., and Stoughton. The Art of Simulation. Reprinted 1975

23. Univac 1100 Series (User Manual) : Sperry Rand Corporation.

ภาคผนวก ก

GPSS (General Purpose Systems Simulator)

1. หลักการของ GPSS

GPSS เป็นภาษาที่ใช้ในการ Simulation ภาษาหนึ่งในบรรดาหลายๆภาษาที่ถูกสร้างและพัฒนาขึ้นมาเพื่อใช้การ และอาจกล่าวได้ว่าโดยแท้จริงแล้ว GPSS เป็นโปรแกรมมากกว่าที่จะเป็นภาษาคอมพิวเตอร์ ทั้งนี้เพราะผู้ใช้ไม่จำเป็นต้องเขียนโปรแกรมเอง หรือมีพื้นความรู้เกี่ยวกับการใช้ภาษาคอมพิวเตอร์เขียนโปรแกรมมากนัก เพียงแต่เลือกใช้ block ต่าง ๆ เฉพาะคำสั่งที่ถูกต้องลงในบัตรคอมพิวเตอร์เพื่อเป็นสื่อให้เครื่องคอมพิวเตอร์เข้าใจ

GPSS ถูกคิดค้นโดย Geoffrey Gordon และเริ่มเผยแพร่เป็นเอกสารในปี ค.ศ. 1961-2 (ด้วยเครื่องคอมพิวเตอร์ IBM704,709 และ 7090) ต่อมาได้มีการปรับปรุงให้มีประสิทธิภาพมากขึ้นกลายเป็น GPSS II, GPSS III (1965), GPSS/360 (1967), GPSS/360's 2nd version และ GPSS V ซึ่งเป็น version ล่าสุด และถือว่าทันสมัยที่สุด อย่างไรก็ตาม GPSS ทุกๆรุ่นต่างมีพื้นฐานแบบเดียวกัน คือเป็น Block-diagram structure

GPSS เหมาะสำหรับผู้ที่ไม่ชำนาญเกี่ยวกับการเขียนโปรแกรม และเหมาะสำหรับงานด้านการศึกษาระบบบริการทั่วๆไป GPSS ประกอบด้วย Block-diagram ซึ่งใน GPSS V มีถึง 48 blocks ที่ต่างกัน แต่ละ block มีชื่อเฉพาะและใช้ในกรณีต่างๆกัน แต่ละ block อาจถูกใช้มากกว่า 1 ครั้งในแบบจำลอง ลักษณะของ block ที่ใช้ในการวิจัยนี้ได้แก่.-

ADVANCE

ทำหน้าที่หาช่วงเวลาที่ใช้ในการให้บริการสำหรับลูกค้า (Customer) แต่ละราย โดยมีการกระจายแบบ Uniform มีค่าอยู่ภายในช่วง (A-B, A+B) ในกรณีที่มีการกระจายแบบอื่น ค่า B จะเป็นชื่อฟังก์ชัน ซึ่งจะอธิบายถึงความน่าจะเป็นสะสมของการกระจายแบบนั้น ๆ และค่า A คือค่าเฉลี่ย

QUEUE

ทำหน้าที่สร้างแถวคอย เมื่อลูกค้าเข้ามาถึงระบบและต้องรอคอย โดยที่แถวคอยนั้นชื่อ A และจำนวนลูกค้าที่เข้าแถวคอยในขณะนั้นมีจำนวนเท่ากับ B ในกรณีที่ B = 1 อาจไม่จำเป็นต้องแสดง Operand B

DEPART

เมื่อเจ้าหน้าที่รับบริการว่าง พร้อมที่จะให้บริการแก่ลูกค้าในแถวคอย Block นี้จะทำหน้าที่ให้ลูกค้าที่อยู่ในแถวคอย A ออกจากแถวคอย A นั้นเสีย จำนวนลูกค้าที่ออกจากแถวคอย A ในขณะนั้นเท่ากับ B ในกรณีที่ B = 1 ก็ไม่จำเป็นต้องแสดงค่า B

SEIZE

ทำหน้าที่ให้ผู้ใช้บริการเข้ารับบริการยังหน่วยบริการที่ให้ชื่อว่า A (ในกรณีนี้ผู้ให้บริการในระบบเป็นแบบ Single Server)

RELEASE

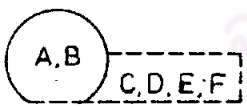
ทำหน้าที่ให้ลูกค้าที่เสร็จสิ้นการใช้บริการจากเจ้าหน้าที่ A ออกจากหน่วยบริการ บล็อคนี้จะใช้คู่กับ SEIZE เสมอ

ENTER

ทำหน้าที่เช่นเดียวกับ SEIZE แต่เป็นกรณีที่ระบบให้บริการมีผู้ให้บริการแบบขนานกัน (Multiple servers) ให้ชื่อว่า Storage A และจำนวนลูกค้าที่เข้ารับบริการใน Storage A ในขณะนั้นเท่ากับ B

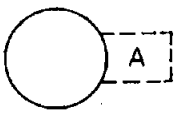
LEAVE

เช่นเดียวกับ RELEASE และใช้คู่กับ ENTER เสมอ



GENERATE

ทำหน้าที่หาช่วงเวลาที่ผู้ใช้บริการแต่ละรายมาถึงระบบโดยมีลักษณะการกระจายเป็น Uniform ซึ่งอยู่ในช่วงเวลา (A-B, A+B) ในกรณีที่มีการกระจายแบบอื่น A จะหมายถึงค่าเฉลี่ยของการกระจายนั้น ส่วน B จะหมายถึงค่าความผันผวนจะเป็นสะสม และมีลูกค้า D ราย เข้ามาสู่ระบบ ลูกค้ารายแรกเข้ามา ณ เวลา C ลูกค้าแต่ละรายมีค่า F เป็น Parameter โดย $0 \leq F \leq 100$ และมี Priority E ในช่วง $0 \leq E \leq 127$



TERMINATE

ทำหน้าที่นำลูกค้าออกจากระบบ โดยมีจำนวนลูกค้าที่ออกจากระบบในขณะนั้นเท่ากับ A ราย

สิ่งที่เข้ามาใช้บริการในระบบแถวคอย เช่น ลูกค้าในห้างสรรพสินค้า, ชิ้นส่วนประกอบของเครื่องจักรที่ส่งเข้ามาเก็บในคลังพัสดุ หรือยวดยานพาหนะซึ่งผ่านจุดที่เราสนใจ เช่นตรงบริเวณสี่แยก เหล่านี้ในภาษา GPSS ถือว่าเป็น Entities ที่เรียกว่า Transactions ซึ่งจะเคลื่อนที่ผ่านเข้ามาในระบบแถวคอย โดยการผ่านบล็อคซึ่งพยายามสร้างให้คล้ายเหตุการณ์จริงมากที่สุด ฉะนั้นเหตุการณ์ที่เกิดขึ้นต่อเนื่องกันในวิธีการจำลองระบบ ก็คือการที่ Transaction เคลื่อนผ่านทุกบล็อคอย่างต่อเนื่องนั่นเอง ภายใน Block diagram สามารถมี Transaction ที่แตกต่างกันมากมาย แต่มีข้อจำกัดว่า ณ เวลาหนึ่ง ๆ แต่ละ Transaction ต้องอยู่ในบล็อคใดบล็อคหนึ่งเท่านั้น แต่ในบล็อคหนึ่งอาจจะมี Transaction มากกว่าหนึ่งในเวลาเดียวกันได้ โดยปกติ Transaction จะเป็น Temporaty entities ซึ่งเคลื่อนที่ระหว่างบล็อค บล็อคที่ประกอบด้วย Transaction อันเดียวในเวลาหนึ่ง ๆ เรียกว่า Facilities เช่น ช่องบริการแบบเดี่ยว (Single server) ในระบบแถวคอย ส่วนบล็อคที่มี Transaction มากกว่าหนึ่งเรียกว่า Storage เช่น Multiple servers ในระบบแถวคอย

GPSS สามารถคำนวณเวลาล่วงหน้าได้โดยใช้ ADVANCE block กล่าวคือ เมื่อ Transaction ผ่านเข้าสู่ block นี้ เวลาที่ใช้ในการให้บริการจะถูกคำนวณและรวมเข้ากับเวลาที่เก็บไว้เดิม (Current time) ซึ่งจะได้เป็นเวลาที่ลูกค้าออกจากแถวหน่วยบริการเข้าสู่บล็อคต่อไป GPSS ยังสามารถเก็บรวบรวมสถิติในระหว่างการจำลองแบบปัญหาได้ด้วย เช่น เก็บรวบรวมจำนวนผู้ใช้บริการ (Transaction) ในหน่วยบริการและความยาวของแถวคอย

โดยปกติผลการจำลองระบบจะถูกพิมพ์ออกมาในรูปแบบมาตรฐาน แต่ผู้ใช้ก็สามารถกำหนดให้ผลการคำนวณออกมาในรูปที่ต้องการได้ ถ้ามีการเรียก block ต่าง ๆ โดยไม่ถูกต้อง ผู้ใช้จะทราบได้ระหว่างการ Compile และถ้าเป็นการผิดพลาดในการคำนวณซึ่งเป็นผลมาจากการเรียกใช้คำสั่ง GPSS ที่ไม่ถูกต้อง ก็จะทราบได้จากเวลาที่ใช้ในการคำนวณ ซึ่งมักจะมากหรือน้อยเกินไป สำหรับข้อผิดพลาดที่ไม่สำคัญนัก ผู้ใช้จะทราบได้โดย Warning messages

เนื่องจาก GPSS มีพื้นฐานมาจาก Flowchart ดังนั้น GPSS จึงสามารถที่จะเรียนรู้ได้เร็วกว่า Simulation language อื่น ๆ นอกจากนั้นโปรแกรม GPSS ยังสั้นและเหมาะสมที่จะนำมาใช้กับระบบที่ผู้ใช้บริการเคลื่อนผ่านเข้าสู่ block ที่กำหนด แต่สำหรับงานที่ไม่มีลักษณะดังกล่าว ภาษาสำหรับการจำลองระบบแบบอื่นอาจจะดีกว่า

2. ส่วนประกอบที่สำคัญของ GPSS ที่ควรกล่าวถึงในที่นี้

2.1 GPSS Entities

Transaction (หรือ Dynamic entities) หมายถึงส่วนประกอบของระบบที่ก่อให้เกิดกิจกรรม อาจหมายถึงลูกค้าที่มารับบริการในห้างสรรพสินค้า, ยวดยานพาหนะในระบบการจราจร และผลิตภัณฑ์ที่ผลิตขึ้นในโรงงาน

Equipment entities อาจแยกได้เป็น 3 ประเภท คือ.-

ก. FACILITY - หน่วยบริการที่ให้บริการได้ครั้งละหนึ่งราย เช่น แคชเชียร์ในร้านค้า

ข. STORAGE - หน่วยบริการที่ให้บริการได้ครั้งละหลายราย เช่น ระบบบริการแบบ Multiple servers

ค. LOGIC SWICH - มีลักษณะเป็น 2 แบบ คือ Set กับ Reset หรือ On กับ Off ซึ่งใช้ควบคุมการเคลื่อนที่ของ Transaction หรือการเกิด Activity อื่น ๆ ขึ้นในระบบ เช่น สัญญาณจราจร

Operational entities มีหน้าที่ควบคุมกิจกรรมในระบบ รวมทั้งแสดงถึงความสัมพันธ์ภายในระบบ และใช้อธิบาย Logical structure ภายในแบบจำลองได้ด้วย Operational entities ที่สำคัญมี 4 แบบ คือ.- QUEUE

USER CHAIN

GROUP

SAVEX

Data entities มีหน้าที่สำคัญคือ นำระบบข้อมูลเข้าสู่แบบจำลอง แสดงถึงความสัมพันธ์ของข้อมูล รวมทั้งบันทึกข้อมูลที่เกิดขึ้นในช่วงการจำลองระบบด้วย Data entities ที่สำคัญมี 3 แบบ คือ.- FUNCTION

VARIABLE

TABLE

2.2 GPSS Statement อาจแยกได้ 3 ประเภทใหญ่ ๆ คือ.-

1. Data definition statement

2. Model statement

3. Model control statement

เมื่อนำ Statement เหล่านี้มาสร้างแบบจำลอง จะสามารถจัดลำดับได้โดยทั่วๆไป ดังนี้.-

JOB

DATA DEFINITION (อาจเป็น CAPACITY, VARIABLE, FUNCTION, TABLE Statement)

MODEL STATEMENT (GENERATE, ADVANCE)

MODEL CONTROL STATEMENT (START, CLEAR, RESET, SAVES)

END

Data definition statement Statements เหล่านี้มีคุณสมบัติเพื่อให้รายละเอียดเกี่ยวกับ Input และ Output ซึ่งมีรูปแบบในการเรียกมาใช้โดยทั่วไป (General format) ดังนี้.-

Name description (, options)[1/] datafield(s)

เมื่อ name - ชื่อที่สร้างขึ้นโดยผู้ใช้

description - อาจเป็น Statement หนึ่งจาก CAPACITY, VARIABLE, TABLE, FUNCTION

options - ขึ้นกับการใช้ Description

datafield(s) - ประกอบด้วยข้อมูลที่กำหนดโดยผู้ใช้

Model Statement กำหนด block ที่ผู้ใช้ใน block diagram นั่นคือจะต้องกำหนด 1 model statement สำหรับ 1 block และมีรูปแบบในการเรียกมาใช้คือ.-

(label) block type (, options) datafield (time field) (go to field)

เมื่อ label - เลือกใช้โดยผู้ใช้ เพื่อแสดงถึง block

block type - GPSS block type name

options - เป็น Subfield ของ block type name

1/ ภายในวงเล็บ ( ) อาจใช้หรือไม่ใช้ก็ได้

datafield : - แสดงถึงรายละเอียดของ block นั้น โดยมากจำนวนของ subfield จะขึ้นกับ block type และมักกำหนดโดยอักษร A,B,C,.....

time field - processing time :- TIME (mean (,mod))

go to field - ใช้สำหรับ Transaction routing

Model control statement Statement เหล่านี้ใช้เพื่อเริ่มและสิ้นสุดการจำลองแบบปัญหา และยังช่วยรักษาสถานภาพของระบบในช่วงเวลาหนึ่ง ๆ ระหว่างการจำลองแบบปัญหา การเรียก Statement นี้จะมีรูปแบบโดยทั่วไป (General format) ดังนี้.-

task (, options) datafield(s)

เมื่อ task - one of the control tasks allowed

options - subfield of task

datafield - information supplied by analyst to some tasks

2.3 ตัวอย่างกิจกรรมที่เกิดขึ้นในการจำลองระบบโดยใช้ GPSS

Transaction Creation หมายถึงการเริ่มนำผู้ใช้บริการเข้าสู่ระบบบริการ และการมาถึงระบบของผู้ใช้บริการจะกำหนดโดยช่วงระยะเวลาที่ลูกค้าแต่ละรายเข้ามาใช้บริการ ช่วงเวลานี้จะหาได้โดยใช้ GENERATE block ซึ่งมีรูปแบบคือ GENERATE A,B,C,D,E,F เมื่อ A-F เป็นเลขจำนวนเต็มบวก ซึ่งอธิบายไว้แล้วในตอนต้น สำหรับระบบที่ไม่ยุ่งยากนักอาจจะใช้เฉพาะ Operand A และ B นั่นคือเริ่มให้ลูกค้ารายแรกเข้ามาสู่ระบบโดยนับเวลาจาก 0 ส่วนจำนวนลูกค้าที่จะเข้ามาสู่ระบบจะถูกกำหนดโดย Control card ที่ชื่อ START ในกรณีที่ช่วงเวลาห่างกันของลูกค้าแต่ละรายที่มาถึงระบบ มีการกระจายแบบอื่นที่มิใช่ Uniform ค่า A จะหมายถึงค่าเฉลี่ยของการกระจายนั้น ส่วน B คือค่าความน่าจะเป็นสะสมของการกระจายนั้น

Queueing and Service Block ที่เกี่ยวข้องกับ Queue มี 4 Block ที่สำคัญซึ่งทำหน้าที่ควบคุมการใช้ Facility และ Storage ดังนี้.-

- ENTER A,B คือการให้ Transaction ผ่านเข้ามาใน Storage A ด้วยจำนวน B
- LEAVE A,B คือการให้ Transaction ออกจาก Storage A ด้วยจำนวน B
- SEIZE A ทำหน้าที่คล้าย ENTER แต่ใช้กับ Facility
- RELEASE A ทำหน้าที่คล้าย LEAVE " "

ส่วน Block ที่เกี่ยวกับการให้บริการ คือ ADVANCE มีรูปแบบในการเรียกใช้คือ ADVANCE A,B ดังอธิบายละเอียดในตอนต้นแล้ว

Statistics ค่าสถิติจะสามารถหาได้โดยการใช้ Statement ต่อไปนี้.-

- Queue A,B เมื่อ A เป็นชื่อของ Queue ที่ผู้ใช้ตั้งขึ้นเอง อาจเป็น Integer หรือ Variable ก็ได้ ส่วน B คือจำนวน Transaction ที่เพิ่มเข้าในแถวคอย A (ในกรณี B ไม่ใช้ หมายถึง B = 1)
- DEPART A,B เมื่อ Transaction เข้ารับบริการแล้ว จำนวน Transaction ในแถวคอย A จะลดลงเท่ากับค่า B
- TABULATE A Statement นี้ต้องใช้คู่กับ TABLE Statement เมื่อ A คือชื่อของ TABLE ใช้คำนวณหาช่วงเวลาระหว่างเวลาที่มาถึงของ Transaction ณ MARK block กับเวลาที่มาถึง TABULATE block และจะบันทึกเวลาเหล่านี้ใน TABLE A
- MARK A บันทึกเวลาการมาของ Transaction เมื่อมาถึง MARK block
- TERMINATE A หมายถึงการนำ Transaction ออกจากระบบ และลดค่าจำนวนนับของ Transaction ซึ่งกำหนดไว้ใน START control card ลงเท่ากับจำนวน A
- START A ใช้กำหนดจำนวนทั้งหมดของ Transaction ที่สร้างขึ้นมาในระหว่างทำการจำลองปัญหา
- TRANSFER A,B,C แสดงถึงให้เคลื่อนย้าย Transaction ไปสู่ Location B หรือ C โดยทำการสุ่มตัวเลขจาก Uniform dist $\frac{n}{}$ ในช่วง (0,1) มา ถ้าค่าที่ได้น้อยกว่าค่า A Transaction นั้นจะเคลื่อนย้ายไปยัง Location C ในทางตรงกันข้าม จะไปยัง Location B

การใช้ GPSS เพื่อจำลองระบบการสำรองที่และจำหน่ายตั๋วลงหน้าที่สถานีรถไฟกรุงเทพ จะต้องใช้คำสั่งดังแสดงไว้บางส่วนในภาคผนวก 3.

ภาคผนวก ข.

วิธีการผลิตเลขสุ่ม

Multiplicative congruential method (บางครั้งเรียกว่า Power-residue generator) วิธีนี้จะหาเลขสุ่มโดยทำการคำนวณจากสมการ $x_{i+1} = x_i.a$ (Modulo m) .....(1)

เมื่อ x_i เป็นเลขคล้ายสุ่มตัวที่ i

x_{i+1} เป็นเลขคล้ายสุ่มตัวถัดไป

a เป็นตัวคูณคงที่

Modulo m หมายความว่า ค่า $(x_i.a)$ ถูกหารด้วย m จนกระทั่งเหลือเศษน้อยกว่าค่า m เลขที่เหลือเศษจึงเป็นเลขคล้ายสุ่มตัวต่อไปคือ x_{i+1}

วิธีการเริ่มต้นโดยค่าเริ่มต้น x เรียกว่า Seed ตัวอย่างการผลิตเลขสุ่มโดยวิธีการนี้ เช่น กำหนดให้ Seed x = 3, parameter a = 7 และ m = 15 ดังนั้น กลุ่มเลขคล้ายสุ่มจะได้ดังนี้.-

$x_1 = 3 \times 7 = 21 \equiv 6$

$x_2 = 6 \times 7 = 42 \equiv 12$

$x_3 = 12 \times 7 = 84 \equiv 9$

$x_4 = 9 \times 7 = 73 \equiv 3$

เช่นนี้เรื่อยไป จะได้เลขคล้ายสุ่มที่มีค่าอยู่ระหว่าง 0 - 14

จากการใช้สมการ (1) จะเห็นว่า เลขคล้ายสุ่มที่ได้ จะเป็นเลขจำนวนเต็มค่าใดค่าหนึ่งในช่วง 0,1,2,.., m-1 หลังจากนั้นแล้วจะได้เลขคล้ายสุ่มชุดเดิมนี้อีก ฉะนั้น คาบของเลขคล้ายสุ่มที่ได้จึงมีค่าไม่เกิน m (แต่อาจจะมีค่าน้อยกว่า m ก็ได้ เมื่อเลือกค่า a และ x ไม่ดีนัก) การเลือกค่า m, a และ x จึงมีความสำคัญในการผลิตเลขคล้ายสุ่มที่มีคาบใกล้เคียงกับค่า m มากที่สุด

ได้มีการทดลองเลือกใช้ค่า m, a และ x ที่จับคู่ต่างๆกัน เพื่อใช้ผลิตเลขคล้ายสุ่มตามสมการที่ (1) พบว่าถ้าเลือก x เป็นเลขคี่ และ $m = 2^r$ (เมื่อ $r > 2$) และ $a = k.8 \pm 3$ (เมื่อ k เป็นเลขจำนวนเต็มบวกใดๆ) จะได้คาบของเลขคล้ายสุ่มมากที่สุด และเท่ากับ 2^{r-2} วิธีการต่อไปนี้เป็นวิธีเลือกค่า Parameter ทั้ง 3 ตัว เพื่อที่จะได้กลุ่มของเลขคล้ายสุ่มที่ดี

1. เลือกค่า m ซึ่งมากกว่าหนึ่ง จากเลขที่ใหญ่ที่สุดใน 1 word ที่คอมพิวเตอร์สามารถรับได้ ตัวอย่างเช่น ในเครื่อง IBM 360/370 system 1 word มีความยาว 32 bit bit สุดท้าย 1 bit ใช้สำหรับแสดงเครื่องหมาย ดังนั้น เลขจำนวนเต็มที่ใหญ่ที่สุดใน 1 word ที่คอมพิวเตอร์รับได้คือ $2^{31}-1 = 2,147,483,647$ นั่นคือ ค่า m ควรมีค่า = 2,147,483,648 ในกรณีที่ใช้เครื่อง IBM 1130/1800 system ซึ่งมี 16 bit-word ควรเลือกค่า $m = (2^{15}-1)+1 = 32,768$

2. ค่า seed x ควรมีค่าที่เป็น prime กับค่า m (Relatively prime to m) เมื่อ m เป็นค่ากำลังของ 2 (จาก $m = 2^r$) ดังนั้น x จึงควรมีค่าเป็นเลขจำนวนเต็มบวกที่เป็นเลขคี่ (ในกรณีที่ใช้ x เป็นเลขคู่ จะพบว่าทุกๆค่า x_i ตัวต่อไปจะเป็นเลขคู่เสมอ จึงไม่มีคุณสมบัติเป็นเลขสุ่ม)

3. ค่าคงที่ที่ใช้เป็นตัวคูณ a (Constant multiplier) ควรมีค่าเป็น prime กับค่า m ด้วย นั่นคือ ค่า a ต้องเป็นเลขคี่ พบว่าวิธีเลือกที่ดีที่สุดสำหรับค่า a เมื่อใช้ความสัมพันธ์

$$a = k.8 \pm 3 \qquad \text{.....(2)}$$

และมีค่าใกล้เลขจำนวนเต็ม $2^{b/2}$ (เมื่อ b เป็นจำนวน bit ใน 1 word ในเครื่องคอมพิวเตอร์) เช่นเครื่อง IBM 360/370 system เป็น 32 bit binary machine เราเลือกใช้ค่า a

$$a = 2^{16}+3 = 65,539$$

จะขอยกตัวอย่าง Subprogram เพื่อใช้ทำการผลิตเลขสุ่มในภาษา FORTRAN โดยที่จะได้เลขสุ่มที่มีการกระจายแบบสม่ำเสมอในช่วง (0,1) (เมื่อใช้กับเครื่อง IBM 7044 36-bit binary)

```
      FUNCTION RNDY1 (DUM)
      DATA A,X /189277, 1175092016l/
      X   =  A * X
      AX  =  X
      RNDY1  =  AX/34359738368.0
      RETURN
      END
```

คำสั่งในบรรทัดที่ 3 ใน Program จะได้ค่า A = 189277 (มาจาก 23,660 x 8 - 3 และอยู่ในช่วง 2^{17} และ 2^{18}) ในคำสั่งเดียวกันนี้กำหนด seed มีค่า = 11,750,920,161 ซึ่งเป็นเลขคี่จำนวนมาก ในคำสั่งบรรทัดที่ 5 เป็นการเปลี่ยนค่า Integer เป็น real ส่วนคำสั่งบรรทัดที่ 6 เพื่อหาเลขทศนิยมโดยการหารด้วย 2^{35} = 34,359,738,368 ดังนั้น เลขจำนวนเต็มที่สุ่มได้ (ซึ่งอยู่ในช่วง 0 ถึง 2^{35-1}) จึงได้เป็นเลขทศนิยมในช่วง (0,1)

เลขสุ่ม 50 ตัวแรก (แต่ละตัวมีทศนิยม 8 ตำแหน่ง) ที่สุ่มได้จากโปรแกรมข้างต้น (โดยใช้เครื่อง IBM 7044) จะมีค่าดังตารางข้างล่างนี้

0.12605673	0.63945656	0.41933851	0.13560773	0.42469418
0.84082449	0.74020635	0.03859064	0.32124565	0.41372764
0.12712769	0.34750068	0.88700985	0.56569849	0.71384876
0.15439663	0.73127418	0.38438318	0.89630192	0.34040167
0,20715574	0.81669794	0.13634698	0.34709622	0.33082132

ข้อควรสังเกต เลขสุ่มที่ผลิตได้โดยวิธี Multiplicative congruential method นี้มีคุณสมบัติเป็นเพียงเลขคล้ายสุ่ม ดังนั้น จึงควรใช้เมื่อต้องการเลขสุ่มที่มีจำนวนหลักมาก ๆ

โปรแกรมสำหรับผลิตเลขสุ่มในเครื่องคอมพิวเตอร์แต่ละเครื่องอาจมีมากกว่า 1 โปรแกรม ตัวอย่างเช่น RANDU เป็นโปรแกรมย่อยสำหรับผลิตเลขสุ่มที่ใช้ในเครื่อง IBM

PLAN - " - " - " - " - " - " - " - ICL

RAND - " - " - " - " - " - " - " - GE225

จากเลขคล้ายสุ่มที่มีการกระจายแบบสม่ำเสมอในช่วง (0,1) ข้างต้นสามารถนำมาหาเลขคล้ายสุ่มที่มีการกระจายแบบสม่ำเสมอในช่วง (A,B) ได้โดยสมการ $Y_i = A + (B - A) . U_i$ เมื่อ U_i เป็นเลขสุ่มที่มีการกระจายแบบสม่ำเสมอในช่วง (0,1)

<u>การผลิตเลขสุ่มที่ไม่ใช่ลักษณะการกระจายแบบสม่ำเสมอ</u> (Generation of non-uniformly distributed random number)

การจำลองแบบปัญหาส่วนมากต้องใช้ตัวเลขที่มีการกระจายเป็นแบบอื่นที่ไม่ใช่การกระจายแบบสม่ำเสมอ เช่น การกระจายแบบปกติ, Exponential, Beta, Gamma, Chi-square, Cauchy

หรือ Weiball Dist^{n} เลขสุ่มซึ่งมีการกระจายดังกล่าวอาจจะหาได้โดยตรงจากเลขสุ่มที่มีการกระจายแบบสม่ำเสมอในช่วง (0,1) มีวิธีการหลายแบบที่จะทำการเปลี่ยนเลขสุ่มที่มีการกระจายแบบสม่ำเสมอไปเป็นเลขสุ่มที่มีการกระจายแบบอื่น ๆ ซึ่งจะกล่าวถึงเฉพาะวิธีการที่นิยมกัน ซึ่งได้แก่.-

ก. The inverse transformation method (หรือ Probability-integral transformation method)

วิธีการนี้จะใช้ Cummulative distribution function, F(x) ของการกระจายที่ต้องการมาผลิตเลขสุ่มขึ้น เมื่อ F(x) มีค่าอยู่ในช่วง (0,1) เราสามารถจะผลิตเลขสุ่มที่มีการกระจายแบบสม่ำเสมอ (R) ขึ้น เลขสุ่มนี้จะมีค่าอยู่ในช่วง (0,1) ด้วย จากนั้นจึงกำหนดให้ F(x) = R และค่า x จะสามารถหาได้เพียงค่าเดียว จากความสัมพันธ์ F(x) = R และ $x = F^{-1}(R)$ จะเป็นตัวแปรที่ได้จากการกระจายที่ต้องการ ความยากของวิธีการนี้จึงขึ้นกับการหา Inverse function คือ $F^{-1}(R) = x$ นั่นเอง ถ้า Inverse fu^{n} นี้สามารถหาได้ ก็เริ่มต้นโดยการหาเลขสุ่มที่มีการกระจายแบบสม่ำเสมอ แล้วจึงเปลี่ยนให้เป็น Inverse fu^{n} เพื่อที่จะหาเลขสุ่มจากการกระจายที่ต้องการ

ตัวอย่าง ให้หาเลขสุ่มจากฟังก์ชั่นนี้โดยใช้วิธี Inverse transformation

เมื่อ

$$f(x) = \begin{cases} 3x^2 & ; \quad 0 \leq x \leq 1 \\ 0 & ; \quad \text{Elsewhere} \end{cases}$$

ดังนั้นจะได้

$$F(x) = \int_0^x 3t^2 \, dt$$

$= x^3$ เป็น Cummulative distribution function ของ f(x)

สามารถสร้าง Inverse transformation ได้ดังนี้.-

$$R = x^3$$

$$x = F^{-1}(R) = (R)^{\frac{1}{3}}$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

นั่นคือ เมื่อต้องการตัวแปรสุ่มจาก Probability density $f(x) = 3x^2$ จะหาตัวแปรที่มีการกระจายแบบสม่ำเสมอก่อน (เรียกว่า R) แล้วตัวแปรสุ่มที่ผลิตได้ก็คือ $x = (R)^{\frac{1}{3}}$

การใช้ Inverse transformation สามารถใช้หาตัวแปรสุ่มจากการกระจายหลายแบบ แต่ถ้าไม่สามารถหา Inverse function $F^{-1}(R)$ ได้ หรือหาได้แต่ยุ่งยากไม่เหมาะในทางปฏิบัติ ก็จำเป็นต้องใช้วิธีการอื่น

ข. Exponential Distribution

Probability Density และ Cummulative distribution function ของ Exponential มีดังนี้.-

$$f(x) = \lambda e^{-\lambda x} \qquad \text{เมื่อ} \quad x \geq 0$$

$$F(x) = 1 - e^{-\lambda x} \qquad " \quad x \geq 0$$

การหาตัวแปรสุ่มจาก Exponential distribution จะอาศัยวิธี Inverse transformation ในข้อ ก. โดยในที่นี้ $F(x)$ จะเท่ากับ $1 - e^{-\lambda x}$ และเมื่อได้เลขสุ่มจากการกระจายแบบสม่ำเสมอ (R) แล้ว จึงกำหนดให้ $R = F(x) = 1 - e^{-\lambda x}$

Inverse transformation ในที่นี้คือ.-

$$1 - R = e^{-\lambda x}$$

โดยที่ $(1 - R)$ ยังเป็นเลขสุ่มที่มีการกระจายแบบสม่ำเสมอ ซึ่งมีขอบเขตจำกัดคือ (0,1) ดังนั้น จึงอาจแทน $1 - R$ ด้วย R และสมการข้างต้นจะเปลี่ยนเป็น

$$R = e^{-\lambda x}$$

$$x = -\frac{\ln R}{\lambda}$$

ดังนั้น $F^{-1}(R) = -\frac{\ln R}{\lambda} = x$ ..... (2)

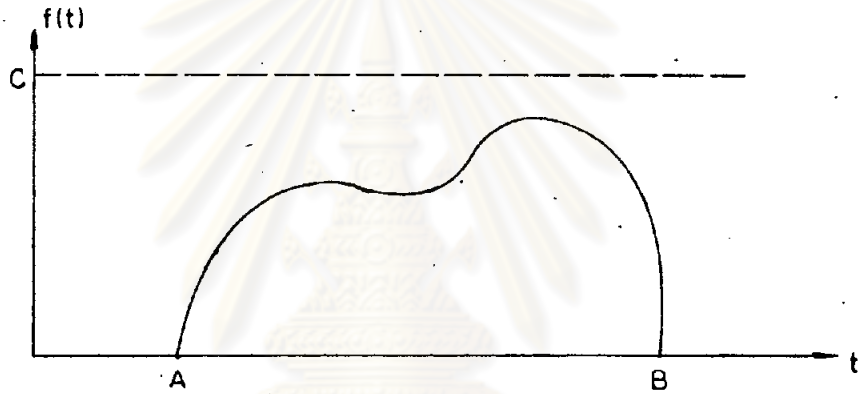
ฉะนั้น วิธีการหาเลขสุ่มที่มีการกระจายแบบ Exponential จึงต้องหาเลขสุ่มที่มีการกระจายแบบสม่ำเสมอ (R) ขึ้นก่อน แล้วหาเลขสุ่มที่มีการกระจายแบบ Exponential (x) โดยใช้สมการ (2)

นอกจากจะสามารถใช้ Inverse transformation หาเลขสุ่มจากการกระจายแบบ Exponential แล้ว วิธีการนี้ยังสามารถใช้ได้ดีกับ Weiball, Geometric Dist^{n} ด้วย โดยมีวิธีการดังกล่าวข้างต้น

ค. The Rejection Method

ใช้หาเลขสุ่มที่มีการกระจายแบบสม่ำเสมออีกวิธีหนึ่ง วิธีการก็คงคล้ายคลึงกับวิธีการแรก คือต้องหาเลขสุ่มจากการกระจายแบบสม่ำเสมอขึ้นก่อน แล้วนำมาตรวจสอบว่าจะยอมรับเป็นเลขสุ่มหรือไม่ การหาเลขสุ่มตัวต่อไปจากการกระจายแบบสม่ำเสมอจะต้องดำเนินการในทำนองเดียวกัน

วิธีการนี้จะใช้ได้ก็ต่อเมื่อ Probability density fu^{n} f(t) ไม่เป็น 0 และต้องอยู่ในขอบเขตจำกัด ดังตัวอย่างในรูปต่อไปนี้ ซึ่งจะเห็นได้ว่าขอบเขตจำกัดของ t คือ A และ B ส่วนขอบเขตจำกัดของ f(t) คือ C และ f(t) จะถูกจำกัดด้วย upper limit C ดูรูป 2-1 ประกอบ



Probability function

วิธีการใช้ Rejection Method มีดังนี้.-

1. หาเลขสุ่ม 2 จำนวนจาก Uniform dist^{n} (สมมติว่าคือ U_1, U_2) ในช่วง (0,1)
2. ใช้ U_1 เพื่อหาจุด P ในแกนนอน ดังนี้

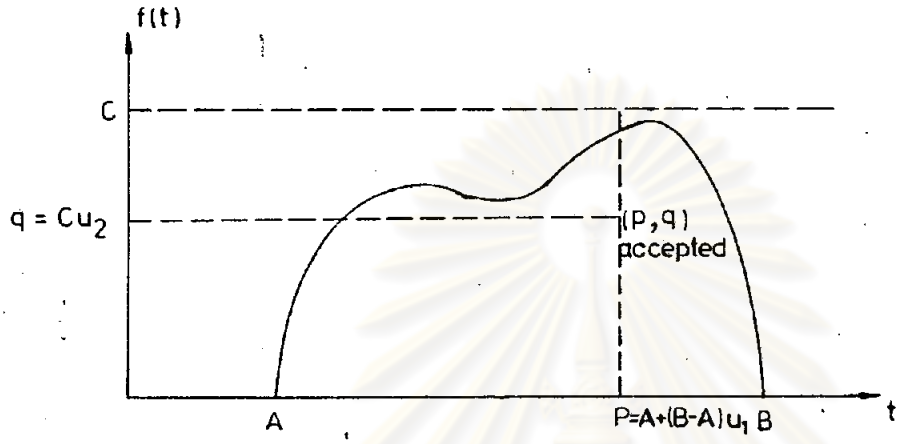
$$P = A + (B - A).\ U_1$$

3. ใช้ U_2 เพื่อหาจุด q ในแกนตั้ง โดย

$$q = C.U_2$$

4. ถ้า $q > f(P)$ แสดงว่า U_1, U_2 คู่นี้ใช้ไม่ได้ ต้องเริ่มต้นใหม่ แต่ถ้าพบว่าค่า P เป็นเลขสุ่มตามลักษณะการกระจายที่กำหนด ก็จะหาเลขสุ่มตัวต่อไปโดยทำซ้ำเช่นนี้ไปเรื่อย ๆ จนกระทั่งได้เลขสุ่มครบจำนวนตามที่ต้องการ

รูปข้างล่างนี้แสดงวิธีการหาเลขสุ่มดังกล่าวข้างต้น



รูปที่ 2.2 Rejection method

วิธีการนี้สามารถหาเลขสุ่มที่การกระจายแบบ Beta และ Gamma ได้ดี นอกจาก 2 วิธีข้างต้นนี้แล้ว ยังมีวิธีการอื่น ๆ อาทิเช่น The Composition Method ใช้กับการกระจายแบบ Poisson และ Erlang รวมทั้ง Binomial Distribution และวิธี Box-Muller transformation ซึ่งใช้กับการกระจายแบบปกติเท่านั้น วิธีการเหล่านี้จะไม่กล่าวถึงไว้ในที่นี้.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค.

แสดง Input และ Output ของโปรแกรมที่ใช้ในการวิจัยบางโปรแกรม เพื่อเป็นประโยชน์แก่ผู้ที่จะทำการศึกษาต่อไป และเพื่อประโยชน์แก่การรถไฟฯในการตรวจสอบผลการวิจัยเพิ่มเติมตามที่ได้กล่าวถึงไว้ในบทที่ 4 และ 5

ผลการคำนวณ สำหรับกรณีอื่นที่ระบุไว้ใน Coding form ค.1 จะปรากฏในลักษณะคล้ายคลึงกับ Printoutใน ค.2 ถึง ค.7

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

FORTRAN

PROGRAMMING FORM

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

LOCATION	OPERATION	A B C D E F
	JOB	
EXPON	FUNCTION RF$1(1)	
+0.0	1 10 4 2 2 2 2 3 3 3 5 5 4 5 0 9 5 6 9 6 9 1 5 7 1 2 7 5 1 3 8	
+.8 1	6 8 6 1 8 3 8 8 2 1 2 9 2 1 3 9 2 2 1 5 2 9 6 2 8 1 9 5 2 9 9 1 9 6 3 2	
+.9 7 3	5 9 8 3 9 9 4 6 9 9 5 5 3 9 9 8 6 2 9 9 9 7 9 9 7 8	
MEANS	FUNCTION RF$1(0.0,1.0/2.0,1.2/0,5.0/8.5,2.4/0,7.2/6.1,3.6/0,8.4/2.9,4.8/0)	
+.9.0 4 6	6 0 0 9 4 6 7 2 0 9 7 0 9 8 4 0 9 7 5 1 9 6 0 9 7 9 2 1 0 8 0 9 8 7 5 1 2 0 0	
+.9.9 5 8	1 3 2 0 9 9 6 0 1 4 4 0 1 5 6 0	
S1	CAPACITY 2	
S2	CAPACITY 2	
	GENERATE 0,TIME(4,6,FN$EXPON)	
	QUEUE LINE,1	
	ENTER S1	
	ADVANCE TIME(FN$MEANS)	
	LEAVE S1	
	QUEUE LINE,2	
	ENTER S2	
	ADVANCE TIME(FN$MEANS)	
	TERMINATE R	
	START 100	
	RESET	
	START 5000	
S1	CAPACITY 3	
S2	CAPACITY 2	
	CLEAR	
	START 100	
	RESET	
	START 5000	
S1	CAPACITY 3	
S2	CAPACITY 3	
	CLEAR	
	START 100	
	RESET	
	START 5000	
	END	

ค. 1 โปรแกรมการคำนวณสำหรับทางสายเหนือ แสดงโปรแกรมใน Coding form เพื่อเจาะลงในบัตรคอมพิวเตอร์

ข. 2 แสดง Print out ของโปรแกรมการจำลองปัญหาแถวคอยในตัวอย่างที่ 1 ที่เขียนด้วยภาษา GPSS (ต่อ)

```
GGPSS
 GPSS 4.1    -08/04-11:34_(000)

    1                JOB
    2          EXPON FUNCTION   RF$1,
    3         +0,0 .1,.104 .2,.22_2 .3,.355 .4,.509 .5,.69 .6,.915 .7,1.2 .75,1.38
    4         +.8,1.6 .84,1.83 .88,2.12 .9,2.3 .92,2.52 .94,2.8_1 .95,2.99 .96,3.2
    5         +.97,3.5 .98,3.9 .99,4.6 .995,5.3 .998,6.2 .999,7 .9997,8
    6          MEANS FUNCTION,C RF$1,0_0 .1035,60 .4884,120 .6977,180 .8372,240
    7         +.9225,300 .9535,360 .9690,420 .97_10,480 .9767,540 .9845,600
    8         +.9922,660 .9922,720 1.,780
    9          S1     CAPACITY      2
   10          S2     CAPACITY      2
   11     1           GENERATE      0 TIME(46,FN$EXPON)
   12     2           QUEUE         LINE1
   13     3           ENTER         S1
   14     4           ADVANCE       TIME(FN$MEANS)
   15     5           LEAVE         S1
   16     6           QUFUE         LINE2
   17     7           ENTER         S2
   18     8           ADVANCE       TIME(FN$MEANS)
   19     9           LEAVE         S2
   20    10           TERMINATE,R
   21                 START         100


STORAGES:
S1              S2


QUEUES:
LINE1           LINE2


FUNCTIONS:
EXPON           MEANS


NUMBER OF TRANSACTIONS ALLOWED:  4139
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

RELATIVE CLOCK TIME	ABSOLUTE CLOCK TIME	TERMINATION COUNT
9523	9523	100

BLOCK #	CURR TRAN	TOTAL TRAN	BLOCK #	CURR TRAN	TOTAL TRAN	BLOCK #	CURR TRAN	TOTAL TRAN	BLOCK #	CURR TRAN	TOTAL TRAN	BLOCK #	CURR TRAN	TOTA TRA
1	0	207	2	82	207	3	0	125	4	2	125	5	0	12
6	22	123	7	0	101	8	1	101	9	0	100	10	0	10

STORAGE NAME	MAXIMUM CONTENTS	AVERAGE CONTENTS	MAXIMUM CAPACITY	AVERAGE CAPACITY	AVERAGE UTILIZATION	TOTAL ENTRIES	TOTAL TRANS	AVERAGE ENT/TRANS	AVERAGE TIME/ENT	CURRENT CONTENT
S1	2	1.98	2	2.00	.9919	125	125	1.00	151.14	2
S2	2	1.97	2	2.00	.9858	101	101	1.00	185.90	1

QUEUE NAME	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	ZEROS PERCENT	AV. TIME/ENT (ALL)	AV. TIME/ENT (NON ZERO)	CURRENT CONTENTS	TABL NAME
LINE1	84	41.04	207	5	2.42	1888.09	1934.82	82	
LINE2	23	10.05	123	4	3.25	778.01	804.16	22	

RANDOM GENERATOR	RANDOM MULTIPLIER	RANDOM INCREMENT	RANDOM SEED
1	1220703125	0	2423881973 3
2	3141592653	2718281829	3141592653
3	2718281829	3141592653	2718281829
4	10604499373	7261067085	10604499373
5	17249876309	7261067085	17249876309
6	30517578125	7261067085	30517578125
7	2565727293	35981228	2565727293
8	107936437	4292354	107936437
9	22438762221	6891	22438762221
10	621444377	92111326	621444377

ค.3 แสดง Output เมื่อใช้คำสั่งตาม ค.2

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
  22              RESET
  23              START       500


STORAGES:
S1              S2

QUEUES:
LINE1           LINE2

FUNCTIONS:
EXPON           MEANS

NUMBER OF TRANSACTIONS ALLOWED:  4139
```

ค.4 แสดงการ Reset ผลการคำนวณใน ค.3 และให้ทำการคำนวณตามโปรแกรมใน ค.2 ใหม่ โดยหยุดคำนวณเมื่อมีผู้โดยสารผ่านระบบ 500 คน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

RELATIVE CLOCK TIME	ABSOLUTE CLOCK TIME	TERMINATION COUNT
37797	47320	500

BLOCK #	CURR TRAN	TOTAL TRAN	BLOCK #	CURR TRAN	TOTAL TRAN	BLOCK #	CURR TRAN	TOTAL TRAN	BLOCK #	CURR TRAN	TOTAL TRAN	BLOCK #	CURR TRAN	TOT TR
1	0	821	2	398	903	3	0	505	4	2	507	5	0	5
6	27	527	7	0	500	8	1	501	9	0	500	10	0	5

STORAGE NAME	MAXIMUM CONTENTS	AVERAGE CONTENTS	MAXIMUM CAPACITY	AVERAGE CAPACITY	AVERAGE UTILIZATION	TOTAL ENTRIES	TOTAL TRANS	AVERAGE ENT/TRANS	AVERAGE TIME/ENT	CURREN CONTEN
S1	2	2.00	2	2.00	1.0000	507	507	1.00	149.10	2
S2	2	2.00	2	2.00	1.0000	501	501	1.00	150.89	1

QUEUE NAME	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	ZEROS PERCENT	AV. TIME/ENT (ALL)	AV. TIME/ENT (NON ZERO)	CURRENT CONTENTS	TAB NAM
LINE1	399	241.76	903	0	.00	10119.45	10119.45	398	
LINE2	29	15.17	527	0	.00	1087.79	1087.79	27	

RANDOM GENERATOR	RANDOM MULTIPLIER	RANDOM INCREMENT	RANDOM SEED
1	1220703125	0	2805782585
2	3141592653	2718281829	3141592653
3	2718281829	3141592653	2718281829
4	10604499373	7261067085	10604499373
5	17249876309	7261067085	17249876309
6	30517578125	7261067085	30517578125
7	2565727293	35981228	2565727293
8	107936437	4292354	107936437
9	22438762221	6891	22438762221
10	621444377	92111326	621444377

ค.5 ผลการคำนวณตามโปรแกรมใน ค.4

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
  24        S1     CAPACITY      3
***** WARNING    AN OVERLAY HAS BEEN MADE
  25        S2     CAPACITY      2
***** WARNING    AN OVERLAY HAS BEEN MADE
  26               CLEAR
  27               START       100


STORAGES:
S1              S2


QUEUES:
LINE1           LINE2


FUNCTIONS:
EXPON           MEANS


NUMBER OF TRANSACTIONS ALLOWED:  4139
```

ค.6 แสดงการ **Clear** โปรแกรมใน ค.2 และเปลี่ยนใช้จำนวนผู้ให้บริการเป็น 3,2 และกำหนดให้โปรแกรมหยุดคำนวณเมื่อผู้ใช้บริการผ่านระบบ 100 คน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

RELATIVE CLOCK TIME	ABSOLUTE CLOCK TIME	TERMINATION COUNT
8656	8656	100

BLOCK #	CURR TRAN	TOTAL TRAN	BLOCK #	CURR TRAN	TOTAL TRAN	BLOCK #	CURR TRAN	TOTAL TRAN	BLOCK #	CURR TRAN	TOTAL TRAN	BLOCK #	CURR TRAN	TOTA TRA
1	0	138	2	25	188	3	0	163	4	3	163	5	0	16
6	59	160	7	0	101	8	1	101	9	0	100	10	0	10

STORAGE NAME	MAXIMUM CONTENTS	AVERAGE CONTENTS	MAXIMUM CAPACITY	AVERAGE CAPACITY	AVERAGE UTILIZATION	TOTAL ENTRIES	TOTAL TRANS	AVERAGE ENT/TRANS	AVERAGE TIME/ENT	CURRENT CONTENT
S1	3	2.96	3	3.00	.9853	163	163	1.00	156.98	3
S2	2	1.97	2	2.00	.9867	101	101	1.00	169.12	1

QUEUE NAME	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	ZEROS PERCENT	AV. TIME/ENT (ALL)	AV. TIME/ENT (NON ZERO)	CURRENT CONTENTS	TABL NAME
LINE1	26	10.80	188	6	3.19	497.30	513.69	25	
LINE2	59	27.69	160	3	1.87	1498.16	1526.78	59	

RANDOM GENERATOR	RANDOM MULTIPLIER	RANDOM INCREMENT	RANDOM SEED
1	1220703125	0	2423881973
2	3141592653	2718281829	3141592653
3	2718281829	3141592653	2718281829
4	10604499373	7261067085	10604499373
5	17249876309	7261067085	17249876309
6	30517578125	7261067085	30517578125
7	2565727293	35981228	2565727293
8	107936437	4292354	107936437
9	22438762221	6891	22438762221
10	621444377	92111326	621444377

ค.7 ผลการคำนวณตาม ค.6

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียน

ชื่อ นางสาว อุมาพร มาระวิชัย

การศึกษา สำเร็จการศึกษาชั้นปริญญาตรี วิทยาศาสตร์บัณฑิต (สถิติ) จากมหาวิทยาลัยเชียงใหม่ ปีการศึกษา 2522

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย