

**П р о г р а м н е
з а б е з п е ч е н н я GPSS
(General Purpose Simulation System)**

Мова імітаційного моделювання GPSS

- GPSS (General Purpose Simulation System) – система моделювання складних об'єктів загального призначення, що розроблена Джефрі Гордоном приблизно у 1960 році. Спочатку розроблювалась і підтримувалась компанією IBM.
- GPSS розроблений низькорівневою мовою програмування Assembly (Assembler).
- Мова GPSS є мовою загального призначення для моделювання мереж масового обслуговування, що набула широкого розповсюдження завдяки книзі [Schriber, T. J. Simulation using GPSS. New York, 1974, Wiley], в якій розглядається велика кількість прикладів моделювання систем різного призначення мовою GPSS.
- Найбільш відомі версії GPSS V, GPSS World (Windows), JGPSS, aGPSS (Mac)
- Переваги GPSS: 1) процесно-орієнтоване представлення моделі, 2) гнучкість формування звітів, 2) реалізація для різних платформ.
- Недоліки: обмежений набір операторів, лінійна структура програми.

Основні правила мови GPSS

- Вимоги, що надходять на обслуговування у мережу масового обслуговування, в мові GPSS називаються *транзактами*.
- Процес обслуговування транзакту описується послідовністю *операторів* мови GPSS, які визначають усе, що відбувається з транзактом з моменту його надходження на обслуговування до моменту завершення обслуговування.
- Оператори мови GPSS характеризують процеси обробки вимог – виникнення транзактів, затримки їх в пристроях, очікування транзактів в черзі, вихід зі СМО.

Структура операторів GPSS

- У записі оператора виділяють три частини:

мітка, назва, поле змінних

AAA ADVANCE 20, 5

- Мітка – використовується в операторах if і loop для вказування на оператор, в який слід зайти. Отже, мітка не завжди потрібна і часто відсутня в описі оператора.
- Назва – завжди з переліку дозволених операторів
- У полі змінних можуть використовуватись як числа, так і змінні.

Оператори GENERATE TERMINATE

- **GENERATE 15,6,120,50,1** - генерація транзактів, інтервали часу між надходженнями транзактів розподілені рівномірно в діапазоні (15-6, 15+6), перший транзакт з'явиться із затримкою в 120 одиниць модельного часу, всього буде створено 50 транзактів, пріоритет транзактів рівний одиниці.
- **GENERATE 6, FN\$FFF,120,50,1** - те ж, але інтервал часу між появами транзактів є ціла частина добутку числа 6 та значення функції FFF.
- **TERMINATE 1** - видалення транзакта з системи, при цьому значення підсумкового лічильника зменшується на одиницю, а моделювання закінчується, якщо значення лічильника стане рівним або менше нуля.

Оператор FUNCTION

- опис функції FTIM, її аргументом являється випадкова величина (на це указує значення RN1), рівномірно розподілена в діапазоні (0,1), функція є неперервною числовою (вказівник C), заданою чотирма точками:

(0;0), (0.1; 0.8), (0.5, 1.6), (1.0; 1.9)

FTIM FUNCTION RN1,C4

0,0/0.1,0.8/0.5,1.6/1.0,1.9

FTIM FUNCTION *2,D4

0,12/1,9/2,8/3,6

- те ж, але аргументом є значення другого параметру транзакта, для якого розраховується значення дискретної величини (D) числової функції FTIM, заданої чотирма вузловими точками. Це поточне значення округляється до найближчого значення аргументу у вузловій точці.

Оператори QUEUE and DEPART

- **QUEUE QAA** - оператор реєстрації черги, довжина черги QAA збільшується на одиницю.
- **QUEUE QAA,2** – те ж , але довжина черги QAA збільшується на 2 одиниці.
- **DEPART QAA** - оператор реєстрації черги, довжина черги QAA зменшується на одиницю.
- **DEPART QAA,4** - те ж, але довжина черги QAA зменшується на 4 одиниці.

Оператори SEIZE, ADVANCE, RELEASE

- **SEIZE DEV** - зайняття пристрою DEV транзактом; якщо пристрій зайнятий, то транзакт затримується в черзі до цього пристрою.
- **ADVANCE A,B** - затримка транзакта на час, визначений вмістом полів A та B, зміст величин, які записані в цих підполях, такий же, як і в блоці GENERATE.
- **RELEASE DEV** - звільнення пристрою DEV транзактом.

Оператори STORAGE, ENTER, LEAVE

- **MEM STORAGE 40** - опис блоку пристроїв WORK ємністю 40 одиниць.
- **ENTER MEM,12** - зайняття транзактом 12 одиниць ємності в накопичувачі MEM.
- **LEAVE MEM,*2** - звільнення k одиниць пам'яті в накопичувачі MEM, де k - значення 2-го параметра транзакта.

Оператори TRANSFER and TEST

- **TRANSFER ,MIT** - безумовна передача управління оператору з міткою (номером) MIT.
- **TRANSFER BOTH,LAB1,ONE** - перехід до оператора з міткою LAB1, якщо він неможливий, то до оператора з міткою ONE, якщо і він неможливий, то транзакт затримується до наступного моменту модельного часу, в який повторюються указані спроби переходу.
- **TRANSFER .4,AAA,END** - транзакт з ймовірністю 0,4 переходить до оператора з міткою END та з ймовірністю 0,6 до оператора з міткою AAA.
- **TRANSFER PICK,FIN7,FIN21** - перехід із рівною ймовірністю до операторів з номерами FIN7, FIN 7+1, FIN 7+2, . . . , FIN 21.
- **TRANSFER FN,AAA,2** - перехід до оператору, мітка якого рівна сумі значень функції AAA і числа 2.
- **TRANSFER P,4,41** - перехід до оператору, мітка якого рівна сумі значень параметра з номером 4 транзакта і числа 41.
- **TRANSFER SBR,PRC,7** - перехід до оператора PRC із записом в параметр з номером 7 транзакта мітки даного оператора.
- **TEST E V7,K256,END** - перехід за умовою (умовна передача управління): в позиціях 13-18 записується знак відношення, в перших двох підполях поля змінних записуються величини, що порівнюються. Якщо умова виконується, то перехід не виконується, інакше - перехід здійснюється до оператора з міткою LAB. Символи відношень: G - більше, L - менше, E - дорівнює, NE – не дорівнює, LE - менше або дорівнює, GE - більше або дорівнює. В даному прикладі перехід не виконується, якщо V7 = 256, інакше перехід виконується до оператора з номером END.

Оператор LOOP

- **LOOP** **6,MIT** - організація циклу – кожний раз при вході в оператор LOOP перевіряється значення шостого параметру: якщо значення не дорівнює нулю, то транзакт переходить до блока з міткою MIT, а значення параметра зменшується на одиницю. Якщо значення шостого параметру при вході в блок LOOP дорівнює нулю, то транзакт слідує в наступний за блоком LOOP перехід

Оператори VARIABLE, ASSIGN, SAVEVALUE

- **5 VARIABLE X2-,25** - розрахунковий оператор, в даному випадку з величини за номером 2, що зберігається, віднімається число 25 і результат присвоюється змінній за номером 5.
- **ASSIGN 2,APP** - змінювання параметрів транзактів, в даному випадку другий параметр транзакта отримає значення APP.
- **SAVEVALUE 5+,*3** - величина за номером 5, що зберігається, збільшується на значення третього параметра транзакта.

Приклад: обслуговування клієнтів службою таксі

Служба замовлення таксі має 5 каналів для одночасного прийняття замовлень по телефону. Час між спробами виклику таксі розподілений за законом Ерланга другого порядку із середнім 180 секунд. Абонент затрачає 30 секунд для набиравання номера і, якщо застає всі канали служби замовлення зайнятими або після з'єднання з'ясовує, що черга на обслуговування перевищує 10 замовлень (в такому випадку замовлення не приймаються), то через 60 секунд він повторює набиравання номера. Після п'яти спроб абонент припиняє набиравання. Служба замовлення таксі має у своєму розпорядженні 30 машин таксі для обслуговування замовлень. Час, витрачений на проїзд до клієнта, залежить від відстані до нього. Імовірності можливих відстаней розподіляються таким чином: 2 км – з імовірністю 0,1, 8 км - з імовірністю 0,2, 9 км - з імовірністю 0,25, 11 км - з імовірністю 0,17, 12 км - з імовірністю 0,23, 20 км - з імовірністю 0,05. Вартість проїзду до клієнта клієнтом не сплачується. Швидкість руху машин рівномірно розподілена в інтервалі 45 ± 5 км/год. Час обслуговування клієнта рівномірно розподілений в інтервалі 50 ± 20 хвилин. Вартість попереднього замовлення складає 2 гривні, вартість проїзду 1 км складає 2 гривні.

Метою моделювання є визначення такої кількості операторів-телефоністів та водіїв таксі, при якій максимізується прибуток служби замовлення.

Приклад. Код мовою GPSS

```
GPSS World - [TAXI1]
File Edit Search View Command Window Help

TAXI      STORAGE  30      ;кількість машин таксі
TEL       STORAGE  5       ;кількість телефоністів
DIS       FUNCTION  RN1,C24 ;функція експоненціального закону розподілу
0.,0./ .100,.104/.200,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/.750,1.38
.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/.9998,8
DOBSL     FUNCTION  RN2,C2  ;функція час обслуговування клієнта в секундах
0.,1800/.9999,4200
DSPEED    FUNCTION  RN3,C2
0.,11.111/.9999,13.888      ;функція швидкості руху машини в метрах на секунду
VIDST     FUNCTION  RN4,D6  ;функція, що задає відстань до клієнта в метрах
0.1,5000/0.3,8000/0.55,9000/0.72,11000/0.95,12000/1.00,20000
CHAS      VARIABLE  FN$VIDST/FN$DSPEED ;час руху машини до клієнта
COST      VARIABLE  FN$DOBSL#FN$DSPEED#0.2 ;вартість обслуговування в копійках за метр
          GENERATE  90,FN$DIS ;з'явився клієнт, що бажає таксі
          ADVANCE   90,FN$DIS ;затримка ерлангового розподілу
          ASSIGN    1,5      ;не більше п'яти спроб набирання номера
DOZVON    ADVANCE   30      ;набирання номера
          TRANSFER  BOTH,VIDP,VIDM ;якщо телефоніст вільний то відповідь інакше відмова
VIDP      ENTER     TEL     ;зайняти телефоніста
          ADVANCE   30      ;тривалість розмови в секундах
          LEAVE     TEL     ;звільнити телефоніста
          TEST L     Q$KLIENT,10,VIDM ;якщо черга на машини більше 10, то відмова
          SAVEVALUE DOHOD+,200 ;підрахувати дохід за попереднє замовлення
          TRANSFER  ,OBSL   ;перейти до обслуговування
VIDM      ADVANCE   60      ;затримка після невдалого дзвінка
          LOOP      1,DOZVON ;цикл, що здійснює дозвон
          SAVEVALUE NEOBSL+,1 ;підрахувати необслужованих клієнтів
          TERMINATE ;вихід із системи клієнта, що не дозволився
OBSL      QUEUE     KLIENT  ;очікувати вільне таксі
          ENTER     TAXI    ;таксі виконує замовлення
          DEPART    KLIENT  ;зменшити кількість очікуючих клієнтів на одиницю
          ADVANCE   V$CHAS  ;таксі прямує до клієнта
          ADVANCE   FN$DOBSL ;таксі обслуговує клієнта
          LEAVE     TAXI    ;звільнити таксі
          SAVEVALUE DOHOD+,V$COST ;підрахувати дохід за обслуговування
          SAVEVALUE KLOBS+,1 ;підрахувати кількість обслужованих клієнтів
          TERMINATE ;вихід із системи клієнта, що обслуговувався
          GENERATE  864000  ;час моделювання 240 годин
          TERMINATE 1
```

Фрагмент звіту

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	9627		0	0
	2	ADVANCE	9627		4	0
	3	ASSIGN	9623		0	0
DOZVON	4	ADVANCE	28823		0	0
	5	TRANSFER	28823		0	0
VIDP	6	ENTER	28487		0	0
	7	ADVANCE	28487		1	0
	8	LEAVE	28486		0	0
	9	TEST	28486		0	0
	10	SAVEVALUE	6833		0	0
	11	TRANSFER	6833		0	0
VIDM	12	ADVANCE	21989		1	0
	13	LOOP	21988		0	0
	14	SAVEVALUE	2788		0	0
	15	TERMINATE	2788		0	0
OBSL	16	QUEUE	6833		10	0
	17	ENTER	6823		0	0
	18	DEPART	6823		0	0
	19	ADVANCE	6823		2	0
	20	ADVANCE	6821		28	0
	21	LEAVE	6793		0	0
	22	SAVEVALUE	6793		0	0
	23	SAVEVALUE	6793		0	0
	24	TERMINATE	6793		0	0
	25	GENERATE	1		0	0
	26	TERMINATE	1		0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE. (-0)	RETRY
KLIENT	10	10	6833	30	9.306	1176.647	1181.835	0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
TAXI	30	0	0	30	6823	1	29.949	0.998	0	10
TEL	5	4	0	5	28487	1	0.989	0.198	0	0

SAVEVALUE	RETRY	VALUE
DOHOD	0	52311875.943
KLOBS	0	6793.000
NEOBSL	0	2788.000

- Для самостійного опрацювання