

Un program in GPSS este format dintr-o insiruire de instructiuni scrise de obicei pe o singura linie.

Partile unei instructiuni in GPSS se numesc campuri. Aceste campuri sunt: numarul instructiunii (optional), eticheta (de cele mai multe ori optionala), un verb care semnifica actiunea care are loc, operanzi, comentarii. Comentariile sunt precedate de “;”. Instructiunile in GPSS se impart in comenzi si in instructiuni bloc.

Entitati importante pentru un model descris de un program in GPSS: blocuri, tranzactii, SNA, facilitati:

- Actiunea care defineste o instructiune este exprimata ori printr-un bloc, ori printr-o comanda. Blocul este elementul de baza al descrierii unui model in GPSS.
- Tranzactiile sunt entitatile care dinamizeaza modelul descris de GPSS prin trecerea lor prin blocuri. Fiecarei tranzactii ii este asociat un numar. Tranzactiile de cele mai multe ori sunt asociate cu clientii dintr-un sistem de asteptare modelat de un program in GPSS.
- SNA – System Numeric Attributes sunt entitatile care dau diferite informatii numerice despre sistemul modelat de program in GPSS. Se folosesc in operanzi sau expresii.
- Facilitatile sunt entitati cu ajutorul carora se simuleaza serviciul cu un singur punct de servire.

Primul program in GPSS (eu il dictez linie cu linie ca studentii sa se obisnuiasca cu modul de lucru in GPSS).

Se deschide GPSS World, File, New, Model. In editorul care se deschide se scrie urmatorul program care poate fi dictat fara explicatii si apoi studentii pot fi intrebati ce cred ei ca face.

```
GENERATE 58  
QUEUE COADA  
SEIZE BENZINA  
DEPART COADA  
ADVANCE 51  
RELEASE BENZINA  
TERMINATE 1
```

Se dau explicatii. Este vorba de simularea unui sistem de asteptare, in particular de functionarea unei benzinarii la care clientii vin la fiecare 58 unitati de timp (in GPSS nu conteaza care este unitatea de timp, conteaza sa fie aceeasi in tot programul), sunt inregistrati in coada, sunt preluati de facilitatea (punctul de serviciu), se marcheaza iesirea lor din coada, primesc serviciul care dureaza 51 de unitati de timp, parasesc facilitatea si, prin ultima instructiune este marcata iesirea lor din sistemul asociat functionarii unei benzinarii.

Definitiiile blocurilor:

GENERATE

- creeaza tranzactii care vor percurge modelul;
- are forma generala:

GENERATE A,B,C,D,E

unde

- A reprezinta timpul mediu intre generari. Daca operandul B nu exista, atunci generarile se vor face la intervale constante de timp, egale cu A;

- B poate fi un numar sau un nume de repartitie statistica. Daca este un numar, atunci generarile se vor face dupa o repartitie unuiforma pe intervalul $[A-B, A+B]$. Adica timpul intre generari poate lua orice valoare intreaga din acest interval. Toate valorile intregi au aceeasi probabilitate de a fi alese. Daca B este un nume de repartitie statistica, atunci timpii intre generari vor fi valori ale averiabilei aleatoare cu repartitia B si media A.
- C este lungimea intervalului de timp cu care generarea tranzactiilor poate intarzia;
- D este limita de generare, numarul maxim de tranzactii care pot fi generate;
- E este prioritatea pe care o vor avea tranzactiile generate.

QUEUE si DEPART

- sunt doua blocuri care nu pot exista unul fara celalalt intr-un model de simulare. Ele nu sunt elemente ale modelului, ci aduna informatii despre asteptarea clientilor, adica informatii despre cozile care eventual se formeaza. QUEUE inregistreaza momentul la care un client intra in coada si DEPART momentul la care un client iese din coada.
- forma lor generala este

$$\begin{array}{l} \text{QUEUE A,B} \\ \text{DEPART A,B} \end{array}$$
unde
- A este numele cozii pentru care sunt adunate informatii;
- B este numarul de tranzactii care sunt inregistrate pentru intrarea/iesirea din coada. Daca lipseste este 1.

SEIZE si RELEASE

- sunt doua blocuri care nu pot exista unul fara celalalt intr-un model de simulare. Ele simuleaza intrarea unui client (tranzactie) la un serviciu (facilitate), respectiv iesirea unui client de la un serviciu (facilitate).
- forma lor generala este

$$\begin{array}{l} \text{SEIZE A} \\ \text{RELEASE A} \end{array}$$
unde A este numele sau numarul facilitatii.

ADVANCE

- este un bloc care simuleaza oprirea tranzactiilor in sistem pentru un anumit interval de timp. Oprirea poate fi determinata de diverse cauze. De exemplu aici oprirea tranzactiilor se face pentru simularea primirii unui serviciu.
- forma generala este

$$\text{ADVANCE A,B}$$
unde A si B au aceeasi semnificatie pe care o au operanzii A si B de la blocul GENERATE.

TERMINATE

- este un bloc care simuleaza iesirea clientilor din sistem, distrugand tranzactiile create de GENERATE.
- forma generala este

$$\text{TERMINATE A}$$
unde A reprezinta numarul care se scade din numarul total de tranzactii pentru care se ruleaza modelul.
- fiecarui bloc GENERATE trebuie sa-i corespunda un bloc TERMINATE.
- in orice model trebuie sa existe un bloc TERMINATE cu operandul A diferit de 0, pentru ca in acest mod se controleaza timpul de rulare. Rularea unui program in GPSS se face prin trecerea unui anumit numar de tranzactii prin sistemul modelat.

Se salveaza programul: File, Save.

Se compileaza: Command, Create simulation.

Se ruleaza: Command, START. Apare o fereastră în care este scris START 1. Se înlocuieste 1 cu 100, acest număr reprezentând numărul de tranzacții (clienți) pentru care se face experimentul de simulare.

Dupa rulare apare automat raportul creat în urma simulării. Dacă dorim o rulare fără afisarea automata a raportului se scrie în fereastră de start START 100,NP.

La o privire rapidă peste raport observăm că:

- este afisat timpul de început și timpul de sfârșit al simulării, numărul de blocuri din model, numărul de facilități, numărul de entități STORAGE.
- sunt afisate blocurile din model și în dreptul fiecăruia numărul de tranzacții care au trecut pe la acel bloc.
- sunt date informații despre facilitatea BENZINA:
 - ENTRIES – câte tranzacții au trecut pe la această facilitate (câți clienți au fost serviti)
 - UTIL. – ce fracțiune (raportat la 1) din timpul total de simulare a fost folosită facilitatea
 - AVE. TIME – timpul mediu de servire.
 - AVAIL., OWNER, PEND, INTER, RETRY, DELAY au semnificație pentru modele mai complicate și nu le discutăm acum.
- informații despre coada COADA
 - MAX - lungimea maximă a cozii;
 - CONT – lungimea cozii la sfârșitul simulării;
 - ENTRY – câte tranzacții au fost înregistrate pentru aceste date despre coadă.
 - ENTRY(0) – câte tranzacții care nu au așteptat deloc au fost înregistrate pentru aceste date despre coadă.
 - AVE. CONT – lungimea medie a cozii.
 - AVE. TIME – timpul mediu de așteptare.
 - AVE.(-0) – timpul mediu de așteptare în coadă pentru clienții care au așteptat un timp mai mare decât 0.
- Informații despre agenda evenimentelor viitoare: FEC (Future Events Chain). Se presupune că următorul eveniment va fi sosirea celui de-al 101-lea client la momentul 5858.

Sunt arătate diferitele ferestre pe care le pot vizualiza după și în timpul simulării:

- a blocurilor: Window, Simulation window, Blocks window;
- a facilităților: Window, Simulation window, Facilities window;
- a cozilor: Window, Simulation window, Queues window;

Se mai face o rulare a programului cu una dintre aceste ferestre deschise și se observă modificările care au loc în timpul rularii. Timpul de rulare este mai lung pentru că este deschisă o fereastră grafică.

Pentru oprirea experimentului se tastează F4. Pentru continuarea lui F2.

Este aratăta fereastră graficelor în care se pot reprezenta grafic diferite mărimi ce caracterizează modelul simulat. Înainte de începerea unei noi rulari se merge la: Window, Simulation Window, Plot Window.

De exemplu putem reprezenta grafic lungimea unei cozi. Pentru identificarea acestei lungimi se folosește SNA-ul Q\$Nume_coadă (în cazul nostru Q\$Coadă), unde \$ este un delimitator. În fereastră deschisă se completează “lungimea cozii” la Label și “Q\$Coadă” la Expression. Se tastează Plot, apoi Memorize, apoi OK. Apare graficul cu cele 2 axe. Se alege și Memorize pentru a folosi această reprezentare grafică și pentru rulari ulterioare. Se dă din nou START 100.

Este aratata fereastra expresiilor: Window, Simulation Window, Expression Window. In fereastra deschisa se completeaza "timp" la Label si "c1" (ceasul simularii) la Expression. Se tasteaza View, apoi Memorize. Observam ca la "Memorized Expressions" apare si lungimea cozii, se selecteaza View, apoi OK. Cu fereastra expressiilor deschisa se mai da odata START 100.

Se pot vedea valorile diferitelor expresii si intr-un alt mod: Command, SHOW, c1. In fereastra de compilare, JOURNAL, este afisata valoarea ceasului simularii. Analog, Command, SHOW, QM\$Coadă (lungimea maxima a cozii).

Acum in fereastra expresiilor se trece si "Tranzactia Activa" la Label, iar la Expression XN1, apoi View, apoi Memorize.

Se mai face o rulare cu fereastra expressiilor deschisa.

Exercitii:

1. Sa se modifice programul astfel incat se se formeze o coada de lungime mai mare. Sa se afiseze graficul lungimii cozii si sa se compare cu cel precedent. Sa se observe modificarile valorilor din raport.
2. Sa se modifice programul astfel incat timpii intre sosiri sa aiba o distributie uniforma pe intervalul [50,58]. Sa se observe modificarile valorilor din raport.
3. Intr-un program pot exista mai multe tipuri de tranzactii. Sa se modifice programul astfel incat la benzinarie sa soseasca doua tipuri de clienti: clienti cu masini rosii, care sosesc la fiecare 58 de unitati de timp, si clienti cu masini albe care sosesc la fiecare 50 de unitati de timp. Sa se ruleze programul pentru 100 de masini albe. Cate masini rosii au fost generate? Cate masini rosii au trecut prin sistem?