

Laborator 1 –GPSS

1. Chestiuni de organizare:

- a. Laboratorul conteaza 30% din nota finala.
- b. Importanta prezentei: prezenta este obligatorie
- c. Bonus: pentru o “activitate buna” (data de rezolvarea problemelor primite in timpul laboratorului) in timpul semestrului se adauga un punct suplimentar la nota de laborator
- d. Nota: lucrare scrisa pe calculator primita in ultima saptamana a semestrului intai, constand intr-o problema de GPSS.

2. Introducere in laboratorul propriu-zis

- a. Scopul simularii este de a crea modele ale unor sisteme reale, care pot fi programate si prin intermediul carora poate fi studiata evolutia in timp a sistemelor.
- b. Pentru a programa un model de simulare se pot folosi diverse limbaje de programare.
- c. Unul dintre limbajele specifice Simularii este GPSS-ul. Limbajele specifice simularii au ca avantaj existenta unor entitati deja implementate, care caracterizeaza un model de simulare. Un program in GPSS consta din descrierea unui sistem iar asupra acestui sistem se pot face experimente (simulari) prin rularea programului.
- d. De unde poate fi luat soft-ul GPSS si documentatia:
<http://www.minutemansoftware.com/>.

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 “;”.

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!

Se deschide GPSS World, File, New, Model. In editorul care se deschide se scrie urmatorul program:

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

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 indicata (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 parcurge 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 uniforma pe intervalul [A-B,A+B]. Adica

timpul între generări poate lua orice valoare întreaga din acest interval. Toate valorile au aceeași probabilitate de a fi alese. Dacă B este un nume de repartiție statistică, atunci timpurile între generări vor fi valori ale variabilei aleatoare cu repartiția B și media A.

- C este lungimea intervalului de timp cu care generarea tranzacțiilor poate întârzia;
- D este limita de generare, numărul maxim de tranzacții care pot fi generate;
- E este prioritatea pe care o vor avea tranzacțiile generate.

QUEUE și DEPART

- sunt două blocuri care nu pot exista unul fără celălalt într-un model de simulare. Ele nu sunt elemente ale modelului, ci adună informații despre așteptarea clienților, adică informații despre cozile care eventual se formează. QUEUE înregistrează momentul la care un client intră în coadă și DEPART momentul la care un client iese din coadă.

- forma lor generală este

QUEUE A,B

DEPART A,B

unde

- A este numele cozii pentru care sunt adunate informații;
- B este numărul de tranzacții care sunt înregistrate pentru intrarea/iesirea din coadă. Dacă lipsește este 1.

SEIZE și RELEASE

- sunt două blocuri care nu pot exista unul fără celălalt într-un model de simulare. Ele simulează intrarea unui client (tranzacție) la un serviciu (facilitate), respectiv ieseirea unui client de la un serviciu (facilitate).

- forma lor generală este

SEIZE A

RELEASE A

unde A este numele sau numărul facilității.

ADVANCE

- este un bloc care simulează oprirea tranzacțiilor în sistem pentru un anumit interval de timp. Oprirea poate fi determinată de diverse cauze. De exemplu aici oprirea tranzacțiilor se face pentru simularea primirii unui serviciu.
- forma generală este

ADVANCE A,B

unde A si B au aceeași semnificație pe care o au operanzii A si B de la blocul GENERATE.

TERMINATE

- este un bloc care simulează ieșirea clienților din sistem, distrugând tranzacțiile create de GENERATE.
- forma generală este

TERMINATE A

unde A reprezintă numărul care se scade din numărul total de tranzacții pentru care se rulează modelul.

- fiecărui bloc GENERATE trebuie să-i corespundă un bloc TERMINATE.
- în orice model trebuie să existe un bloc TERMINATE cu operandul A diferit de 0, pentru că în acest mod se controlează timpul de rulare. Rularea unui program în GPSS se face prin trecerea unui anumit număr de tranzacții prin sistemul modelat.

Se salvează programul: File, Save.

Se compilează: Command, Create simulation.

Se rulează: Command, START. Apare o fereastră în care este scris START 1. Se înlocuiește 1 cu o valoare întreagă pozitivă reprezentând numărul de tranzacții (clienți) pentru care se face experimentul de simulare.

După rulare apare automat raportul creat în urma simulării. Dacă dorim o rulare fără afișarea automată a raportului se scrie în fereastră de start, de exemplu, START 100,NP.

La o privire rapidă peste raport observăm că:

- este afișat timpul de început și timpul de sfârșit al simulării, numărul de blocuri din model, numărul de facilități, numărul de entități STORAGE.
- sunt afișate 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 serviți)
 - 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 semnificatie pentru modele mai complicate si nu le discutam acum.
- informatii despre coada COADA
 - MAX - lungimea maxima a cozii;
 - CONT – lungimea cozii la sfrasitul simularii;
 - ENTRY – cate tranzactii au fost intregistrate pentru aceste date despre coada.
 - ENTRY(0) – cate tranzactii care nu au asteptat deloc au fost intregistrate pentru aceste date despre coada.
 - AVE. CONT – lungimea medie a cozii.
 - AVE. TIME – timpul mediu de asteptare.
 - AVE.(-0) – timpul mediu de asteptare in coada pentru clientii care au asteptat un timp mai mare decat 0.
- Informatii despre agenda evenimentelor viitoare: FEC (Future Events Chain). Se presupune ca urmatorul eveniment va fi sosirea celui de-al 101-lea client la momentul 5858.

Ferestre care pot fi vizualizate dupa si in timpul simularii:

- a blocurilor: Window, Simulation window, Blocks window;
- a facilitatilor: 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 si se observa modificarile care au loc in timpul rularii. Timpul de rulare este mai lung pentru ca este deschisa o fereastră grafica.

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

Sa observam de asemenea fereastra graficelor in care se pot reprezenta grafic diferite marimi ce caracterizeaza modelul simulat. Inainte de inceperea unei noi rulari se merge la: Window, Simulation Window, Plot Window.

De exemplu putem reprezenta grafic lungimea unei cozi. Pentru identificarea acestei lungimi se foloseste SNA-ul Q\$Nume_coada (in cazul nostru Q\$Coadă), unde \$ este un delimitator. In fereastra deschisa se completeaza “lungimea cozii” la Label si “Q\$Coadă” la Expression. Se tasteaza Plot, apoi Memorize, apoi OK. Apare graficul cu cele 2 axe. Se alege si Memorize pentru a folosi aceasta reprezentare grafica si pentru rulari ulterioare. Se da din nou START 100.

Sa se studieze 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 expresiilor 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?