

# GPSS/H

## ***Lanturi utilizator***

Ordinea implicita de tratare a tranzactiilor este data de sosirea acestora si tine cont si de prioritate (FIFO cu prioritati). Daca se doreste folosirea unei politici LIFO sau tratarea in functie de valoarea unui anumit parametru (crescator sau descrescator) se folosesc lanturile utilizator.

### **LINK**

Blocul LINK extrage tranzactii din lantul evenimentelor curente si le pune temporar in repaus intr-un lant utilizator. Forma generala a sa este:

<b>LINK</b>	<b>A, B, C</b>
-------------	----------------

Unde

- A este lantul utilizator.
- B este parametrul folosit.
- C eticheta la care se transmite tranzactia daca nu exista nimic in lantul utilizator.

Atributele unui lant utilizator sunt:

CA – numarul mediu de tranzactii din lant.

CC – numarul total de intrari din lant.

CH – numarul de tranzactii din lant.

CM – numarul maxim de tranzactii din lant.

CT – timpul mediu per tranzactie in lant.

### **UNLINK**

Blocul UNLINK extrage tranzactii dintr-un lant utilizator si le repune in lantul evenimentelor curente. Forma generala a sa este:

<b>UNLINK</b>	<b>A, B, C, D, E, F</b>
---------------	-------------------------

Unde

- A este lantul utilizator.
- B eticheta la care se transmite tranzactia dezlegata.
- C e numarul de tranzactii dezlegate.
- D e numele unui parametru folosit.
- E e valoarea etalon a parametrului D.
- F e eticheta la care se merge daca nu se indeplineste conditia.

## Exemplu

```

SIMULATE
GENERATE      RVEXPO(2,20),,,,,,1PL
ASSIGN        TSERV,FRN(3)*9+12,PL
QUEUE         QAST
LINK          ORDINE,TSERV,NEXT
NEXT          SEIZE      RESURSA
              DEPART     QAST
              ADVANCE    PL(TSERV)
              RELEASE    RESURSA
TSTUNL        UNLINK     ORDINE,NEXT,1
              TERMINATE  1
              START      200
              END

```

Daca se doreste tratarea in ordine inversa se inlocuieste linia TSTUNL de mai sus cu urmatoarea:

```

. . . . .
TSTUNL  UNLINK      ORDINE,NEXT,1,BACK      tratare in ordine descrescatoare TSERV
. . . . .

```

Servirea conditionata se modeleaza astfel:

```

SIMULATE
COND      BARIABLE      PL(TSERV)'LE'15
*-- scrie antet rezultate
      PUPPIC      FILE=RLCOND,LINES=2
      SOSIRE      INCEPUT SERVIRE
      TIMP      NR      TSERV      NR      TSERV      LG.COADA
XXX      GENERATE      RVEXPO(2,20),,,,,,2PL
      ASSIGN        NUMAR,N(XXX),PL
      ASSIGN        TSERV,FRN(3)*9+12,PL
      BPUTPIC      FILE=RLCOND,C1,PL(NUMAR),PL(TSERV)
****.**** ** **
      QUEUE         QAST
      LINK          ORDINE,TSERV,NEXT
NEXT      SEIZE      RESURSA
      DEPART     QAST
      BPUTPIC      FILE=RLCOND,C1,PL(NUMAR),PL(TSERV),Q(QAST)
****.**** ** **
      ADVANCE      PL(TSERV)
      RELEASE      RESURSA
*-- serveste mai intai tranzactiile cu TSERV <= 15 si
*-- apoi pe celelate, in ordinea descrescatoare a TSERV
      UNLINK      ORDINE,NEXT,1,BV(COND),,TIP2
      TERMINATE   1
TIP2     UNLINK      ORDINE,NEXT,1,BACK
      TERMINATE   1
*-- executie simulare
      START      200
      END

```

Tratare conditionata de TIP:

```

SIMULATE
TIPUL VARIABLE RN(4)/250+1
*-- scrie antet rezultate
PUTPIC FILE=RLPAR,LINES=2
SOSIRE INCEPUT SERVIRE
TIMP NR TSERV NR TSERV LG.COADA
XXX GENERATE RVEXPO(2,20),,,,,,3PL
ASSIGN NUMAR,N(XXX),PL
ASSIGN TIP,V(TIPUL),PL
ASSIGN TSERV,FRN(3)*9+12,PL
BPUTPIC FILE=RLPAR,C1,PL(NUMAR),PL(TIP),PL(TSERV)
*****.**** ** ** ** **
QUEUE QAST
LINK ORDINE,TSERV,NEXT
NEXT SEIZE RESURSA
DEPART QAST
BPUTPIC FILE=RLPAR,C1,PL(NUMAR),PL(TIP),PL(TSERV),Q(QAST)
*****.**** ** ** ** **
ADVANCE PL(TSERV)
RELEASE RESURSA
*-- serveste mai intai tranzactiile cu TIP = 3 si
*-- apoi pe celelalte, in ordinea crescatoare a timpului de servire
UNLINK ORDINE,NEXT,1,TIP,3,REST
TERMINATE 1
REST UNLINK ORDINE,NEXT,1
TERMINATE 1
START 200
END

```

## Multiplicarea tranzactiilor

### SPLIT

Blocul SPLIT creaza un anumit numar de copii ale unei tranzactii. Forma generala a sa este:

```
SPLIT A,B,C,D
```

Unde

- A este numarul solicitat de copii (identice) ale tranzactiei originale.
- B este eticheta la care sunt trimise copiile (tranzactia originala este trimisa totdeauna la blocul urmator).
- C este un parametru optional in care se pastreaza numarul de ordine al copiilor.
- D este un parametru optional care stabileste numarul parametrilor tranzactiilor copii; acesti parametri sunt initializati cu valorile parametrilor tranzactiei originale (daca tranzactia initiala are prea putini parametri, ceilalti sunt initializati cu zero).

### ASSEMBLE

Blocul ASSEMBLE grupeaza un numar de tranzactii prin contopire si trimite tranzactia astfel obtinuta la blocul urmator. Forma generala a sa este:

```
ASSEMBLE A
```

Unde

- A este numarul de tranzactii care trebuie sa intre in bloc pentru grupare.

## Exemplu:

**ASSEMBLE 10**

Blocul asteapta 10 tranzactii pe care le contopeste intr-o singura tranzactie.

## GATHER

Blocul GATHER grupeaza un numar de tranzactii fara ale distruge si trimite grupul astfel obtinut la blocul urmator. Forma generala a sa este:

**GATHER A**

Unde

- A este numarul de tranzactii care trebuie sa intre in bloc pentru grupare.

## MATCH

Blocul MATCH are rolul de a sincroniza tranzactiile: o tranzactie care intra intr-un astfel de bloc este oprita pana ce o tranzactie (din aceeasi familie) intra in blocul MATCH conjugat.

**MATCH A**

Unde

- A este eticheta blocului cu care se face sincronizarea.

2 blocuri MATCH conjugate trebuie sa foloseasca aceeasi pereche de etichete:

```
et1    MATCH et2
...
et2    MATCH et1
```

## Exemplu

La depozitul unei societati comerciale cumparatorii sosesc cu o nota de comanda in doua exemplare; functionarul de la ghiseu preia comanda, da o copie unuia dintre lucratorii care impacheteaza produsele comandate, intocmeste factura si incaseaza contravaloarea comenzii. Cumparatorul poate pleca in momentul in care a terminat formalitatile de plata SI a primit pachetul cu produsele comandate.

```

SIMULATE
STORAGE          S(LUCR),2          Lucratori la impachetare
*--- statistici colectate, in cate 2 variante
TASTP   TABLE   M1,0,2,7          Timp Asteptare Pachet,
TQASTP  QTABLE   ASTP,0,1,7        din momentul terminarii platii
TASTF   TABLE   M1,0,2,7          Timp Asteptare Factura,
TQASTF  QTABLE   ASTF,0,1,7        de la terminarea impachetarii
GENERATE RVEXPO(2,8)              Sosire cumparator
QUEUE    TSERV    Colecteaza statistici timp servire
SEIZE    FUNC     Functionarul preia comanda
SPLIT    1,PREG   O copie a comenzii merge la lucratori
ADVANCE  FRN(3)*5+6 Se intocmeste factura
RELEASE  FUNC     Functionarul a terminat facturarea
MARK
QUEUE    ASTP
SFH      MATCH    SFP              Asteapta pachetul (daca nu e gata)
DEPART   ASTP
TABULATE TASTP    Actualizeaza statistici
DEPART   TSERV
TRANSFER ,GATA    Merge sa ia copia comenzii
*--- pregatire pachet
PREG     QUEUE    QLUCR            Colecteaza statistici asteptare lucrator
ENTER    LUCR     Un lucrator ia copia comenzii
DEPART   QLUCR
ADVANCE  FRN(4)*10+5 Pregateste pachetul conform comenzii
LEAVE    LUCR     Pachetul este gata
MARK
QUEUE    ASTF
SFP      MATCH    SFH              Asteapta factura (daca nu e gata)
TABULATE TASTF    Actualizeaza tabel
DEPART   ASTF
GATA     ASSEMBLE 2                Grupeaza factura si copia comenzii
TERMINATE
* --- control durata simulare
GENERATE 480
TERMINATE 1
* --- executie experiment simulare
START    1
END

```

## Modificari ale atributelor tranzactiilor

### COUNT

Blocul COUNT permite contorizarea entitatilor care au anumite proprietati. Forma generala a sa este:

```
COUNT rel A,B,C,D,E
```

Unde

- rel este un operator logic (LR, LS, U, NU, I, NI, SE, SNE, SF, SNF) sau relational (L, LE, E, NE, G, GE).
- A este numarul parametrului tranzactiei in care va fi depozitat rezultatul contorizarii.
- B si C reprezinta limitele inferioara si superioara ale entitatilor (prima si ultima entitate din intervalul considerat).
- D este etalonul folosit de relatia rel.
- E este argumentul care se compara cu D prin relatia rel.

## Exemplu

```
COUNT SF      2,10,20
```

Se numara resursele multiple pline, incepand cu cea cu numarul 10 si terminand cu cea cu numarul 20 (trebuie sa fie definite cel putin 20 de resurse multiple) iar rezultatul este depus in parametrul al doilea al tranzactiei curente.

```
COUNT LE      P1,1,8,X10,FC
```

In parametrul P1 al tranzactiei se introduce numarul de resurse simple din intervalul 1 – 8 in care au intrat mai putine tranzactii decat valoare precizata de X10.

## SELECT

Blocul SELECT permite selectarea primei entitati dintr-un interval de entitati care indeplineste o anumita conditie. Forma generala a sa este:

```
SELECT rel A,B,C,D,E,F
```

Unde

- rel este un operator logic (LR, LS, U, NU, I, NI, SE, SNE, SF, SNF), relational (L, LE, E, NE, G, GE), MIN sau MAX.
- A este numarul parametrului tranzactiei in care va fi depozitat numarul primei entitati care indeplineste conditia.
- B si C reprezinta limitele inferioara si superioara ale entitatilor (prima si ultima entitate din intervalul considerat).
- D este etalonul folosit de relatia rel, lipseste cand se foloseste MIN sau MAX.
- E este argumentul care se compara cu D prin relatia rel.
- F este blocul la care se trimite tranzactia daca nici o entitate nu indeplineste conditia.

## Exemplu

```
SELECT LE      1,2,10,15,Q
```

Din cozile aflate intervalul 2 – 10 se selecteaza numarul primei cozi care are lungimea cel mult egala cu 15; acest numar se depune in parametrul 1 al tranzactiei.

```
SELECT MAX     1,2,10,FR
```

In parametrul P1 al tranzactiei se introduce numarul resursei simple din intervalul 2 – 10 care are gradul maxim de utilizare.

## Exemplu

Utilizarea blocului SELECT pentru a alege o coada:

```

SIMULATE
GENERATE          RVEXPO(2,2),,,,,,1PB
*- se aseaza la coada de lungime minima
SELECT MIN        UNDE,1,3,,Q
INTRA  QUEUE      PB(UNDE)
        SEIZE      PB(UNDE)
        DEPART     PB(UNDE)
        ADVANCE    4.5,4
        RELEASE    PB(UNDE)
        TERMINATE  1
GENERATE          5,,,,,1PB
ASSIGN           UNDE,3,PB      Utilizeaza resursa 3
TRANSFER        ,INTRA
START           200
END

```

Selecteaza prima resursa simpla disponibila:

```

SIMULATE
LIBER  BARIABLE    FNU(1)+FNU(2)+FNU(3)
GENERATE RVEXPO(2,2),,,,,,1PB
QUEUE  QAST
TEST E BV(LIBER),1
SELECT NU UNDE,1,3
SEIZE  PB(UNDE)
DEPART QAST
ADVANCE 5,4
RELEASE PB(UNDE)
TERMINATE 1
START 200
END

```

Selecteaza depozitul unde exista suficient loc:

```

SIMULATE
STORAGE S(1),20/S(2),20/S(3),20
NUMAR  VARIABLE    RN(3)/100+1
LOC    BARIABLE    (R(1)'GE'PB(NECESAR))+ (R(2)'GE'PB(NECESAR))_
        + (R(3)'GE'PB(NECESAR))
GENERATE RVEXPO(2,2),,,,,,2PB
ASSIGN  NECESAR,V(NUMAR),PB
QUEUE  QAST
TEST E BV(LOC),1
SELECT LE UNDE,1,3,20-PB(NECESAR),S
ENTER   PB(UNDE),PB(NECESAR)
DEPART  QAST
ADVANCE 10,4
LEAVE  PB(UNDE),PB(NECESAR)
TERMINATE 1
START 200
END

```

## Grupuri de tranzactii

Tranzactiile pot fi grupate (blocul GROUP) pe baza unor atribute comune sau a unor criterii si pot fi referite in grup pentru simplificarea unor operatii (blocurile JOIN, REMOVE, EXAMINE, SCAN, ALTER)

### JOIN

Blocul JOIN permite unei tranzactii sa adere la un grup. Forma generala a sa este:

```
JOIN A
```

Unde

- A este numarul grupului.

## Exemplu

```
JOIN 10
```

Tranzactia curenta adera la grupul cu numarul 10.

## REMOVE

Blocul REMOVE permite unei tranzactii sa iasa dintr-un grup. Forma generala a sa este:

```
REMOVE A,B,C,D,E,F
```

Unde

- A este numarul grupului.
- B e numarul de tranzactii care se scot din grup. Daca se precizeaza ALL atunci se scot toate tranzactiile (sau doar acelea care indeplinesc conditia specificata de D si E).
- C nu se foloseste pentru grupuri de tranzactii.
- D este atributul tranzactiei care este testat pentru scoaterea conditionata din grup. El poate fi PR (prioritatea) sau un numar, reprezentand numarul parametrului implicat.
- E este valoarea cu care se testeaza atributul D; la egalitate se realizeaza scoaterea din grup.
- F este blocul la care se trimite tranzactia curenta daca nici o entitate nu indeplineste conditia.

## Exemplu

```
REMOVE 1
```

Tranzactia curenta este scoasa din grupul cu numarul 1, daca face parte din el.

```
REMOVE 2,ALL,,PR,3,5
```

Se scot toate tranzactiile din grupul 2 care au prioritatea 3 iar daca nici o tranzactie nu are prioritatea 3, tranzactia curenta e trimisa la blocul 5.

## EXAMINE

Blocul EXAMINE verifica daca tranzactia curenta apartine unui grup. Forma generala a sa este:

```
EXAMINE A,B,C
```

Unde

- A este numarul grupului.
- B nu se foloseste pentru grupuri de tranzactii.
- C este eticheta la care se face transferul daca tranzactia nu e membra a grupului (daca lipseste se foloseste blocul urmator).

## Exemplu

```
EXAMINE 1,,ABC
```

Se verifica daca tranzactia curenta este membra in grupul cu numarul 1, iar daca nu face parte din el se trimite catre eticheta ABC.

## SCAN

Blocul SCAN determina daca intr-un grup exista o tranzactie ale carei atribute au numite valori, modificand traseul tranzactiei curente daca in grup nu exista tranzactii cu proprietatea mentionata.



SCAN A,B,C,D,E,F

Unde

- A este numarul grupului.
- B e parametrul a carei valoare se compara cu C.
- C este valoarea care se compara cu parametrul B.
- D este atributul tranzactiei membre a carui valoare se va memora in tranzactia curenta.
- E este numarul parametrului in care se face memorarea lui D.
- F este blocul la care se trimite tranzactia curenta daca nici o tranzactie nu indeplineste conditia (daca lipseste se foloseste blocul urmator).

## Exemplu

SCAN 1,PR,6,5,1

Se verifica tranzactiile din grupul cu numarul 1 pentru a se gasi tranzactia cu prioritatea 6 iar pentru prima tranzactie gasita, valoarea parametrului sau 5 este depusa in parametrul 1 al tranzactiei curente. Daca in grupul 1 nu exista nici o astfel de tranzactie, tranzactia curenta trece la blocul urmator.