

PUC

Série: Monografias em Ciência da Computação
Nº8/78

ESTUDO COMPARATIVO DA APLICAÇÃO DE LINGUAGENS
DE SIMULAÇÃO EM SISTEMAS DE FILAS

por

H. M. Quintella

M. C. L. Gomes

W. C. Koelsch

Departamento de Informática

Pontificia Universidade Católica do Rio de Janeiro

Rua Marquês de São Vicente, 225 - CEP 22453

Rio de Janeiro — Brasil

PUC/RJ - DEPARTAMENTO DE INFORMÁTICA

Série: Monografias em Ciência da Computação
Nº 8/78

Series Editor: Michael F. Challis

Abril, 1978

ESTUDO COMPARATIVO DA APLICAÇÃO DE LINGUAGENS
DE SIMULAÇÃO EM SISTEMAS DE FILAS*

por

Heitor M. Quintella
William C. Koelsch
M.^a Cristina L. Gomes

* Trabalho parcialmente financiado pela FINEP.

For copies contact:

Rosane T.L.Castilho
Head, Setor de Documentação e Informação
Depto. de Informática - PUC/RJ
Rua Marques de São Vicente, 209 - Gávea
22453 - Rio de Janeiro - RJ - Brasil

SUMÁRIO

1 - INTRODUÇÃO AOS SISTEMAS DE FILAS.....	1
2 - TRATAMENTO TEÓRICO DO SISTEMA.....	1
3 - TRATAMENTO DO SISTEMA POR MEIO DE LINGUAGENS DE SIMULA- ÇÃO.....	3
4 - LGS (UMA LINGUAGEM GERAL DE SIMULAÇÃO).....	4
5 - RESULTADOS DA SIMULAÇÃO.....	5
APENDICE - <u>PROGRAMAS UTILIZADOS</u>	
REFERÊNCIAS.....	18

1 - INTRODUÇÃO AOS SISTEMAS DE FILAS

Por sistemas de filas entende-se uma classe de sistemas dinâmicos que por conveniência são também chamados de fluxo. Assim sistema de fluxo ou filas é todo sistema em que determinada entidade se move através de canais para ir de um ponto a outro [Kleinrock, 1975]. Por exemplo o sistema de tráfego é um sistema de fluxo.

Tais sistemas têm a sua entidade definida pelos canais (que prestam serviço às entidades) e por filas (formadas pelas entidades). Na fig.1 ilustra-se a estrutura de um sistema simples e a notação utilizada nesse trabalho.

Todo sistema é caracterizado por uma entrada e por uma saída. No nosso exemplo as entradas são representadas por chegadas de entidades ao sistema enquanto as saídas são representadas por fins de serviço no canal. Na fig.2 ilustra-se o número de elementos no sistema tratado na fig.1

2 - TRATAMENTO TEÓRICO DO SISTEMA

O sistema aqui tratado para teste é o de um canal e uma fila simples como o que foi apresentado na seção anterior. Chamaremos μ a taxa média de serviço e λ a taxa média de chegada de eventos. Sendo todos os eventos perfeitamente aleatórios, as chegadas são distribuídas de acordo com Poisson e conseqüentemente os intervalos entre as chegadas são exponencialmente distribuídos bem como os tempos de serviço (com média $1/\mu$).

Dessa maneira as seguintes grandezas são previstas pela teoria:

$$\text{Seja } \rho = \frac{\lambda}{\mu} \quad e \quad P_0 = \frac{1}{\sum_{n=0}^{k-1} \frac{\rho^n}{n!} + \frac{\rho^k}{(k-1)!(k-\rho)}}$$

P_0 - probabilidade de que haja uma população nula no sistema

$$\text{então } \langle l \rangle, \text{ componente médio da fila é: } \frac{\rho^{k+1}}{(k-1)!(k-\rho)^2} P_0$$

$$\langle W_q \rangle \text{ tempo médio de espera na fila é: } \frac{\langle \rho \rangle}{\lambda}$$

1. FUNCIONAMENTO DOS SISTEMAS DE FILAS -2-

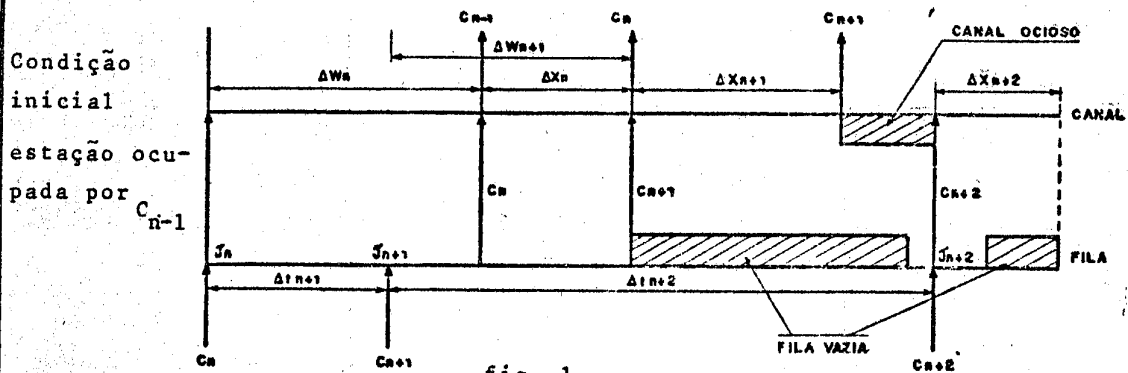


fig. 1

diagrama de tempo no intervalo $(\tau_n, \tau_{n+2} + \Delta X_{n+2})$

C_n → elemento de ordem n de chegada ao sistema

ΔX_n → intervalo de tempo de serviço para C_n

τ_n → hora de chegada no sistema

Δt_{n+1} → intervalo de tempo entre as chegadas de C_n e C_{n+1}

ΔW_n → intervalo de tempo de espera de C_n

Δs_n → intervalo de tempo de permanência no sistema (canal + fila)

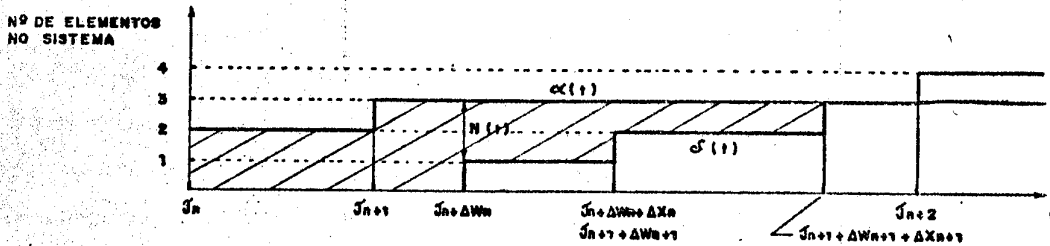


fig. 2

$\alpha(t)$ nº de chegadas no intervalo $(0, t)$

$\delta(t)$ nº de partidas no intervalo $(0, t)$

$N(t)$ nº de elementos no sistema no instante t

$\gamma(t) = \int_0^t [\alpha(t) - \tau(t)] dt$ tempo total que todos os elementos gastaram no sistema no intervalo $(0, t)$ em unidades elemento-segundo.

<s> tempo médio gasto no sistema \bar{e} : $\langle W_q \rangle + \frac{1}{\mu}$
<n> número médio de transações no sistema: $\langle l \rangle + \rho$

E claramente as soluções são válidas somente se $\frac{\lambda}{\mu} < 1$

3 - TRATAMENTO DO SISTEMA POR MEIO DE LINGUAGENS DE SIMULAÇÃO

As linguagens de simulação são linguagens de programação projetadas para auxiliar analistas a programar e analisar modelos de simulação. Para atingir tal fim elas necessitam pelo menos das seguintes características [Kiviat, 1969]

- 1 - Capacidade para representação de dados que permita modelagem direta e eficiente
- 2 - possibilidade de reprodução e representação da dinâmica de sistemas
- 3 - manipulação eficiente de rotinas de geração e análise de variáveis aleatórias e de séries temporais

Na verdade essas três características decorrem da definição tradicional de simulação como sendo o "uso de um modelo numérico para estudar o comportamento de um sistema que varia no tempo" [NAYLOR, 71]

Dentre as linguagens de simulação dedicadas ao tratamento de sistemas de filas três se destacam quanto à portabilidade

- a) GPSS [GORDON, M62]
- b) SIMSCRIPT [Markowitz et al., 63]
- c) GASP [Pritsker and Kiviat, 69]

Todas no entanto padecem de algum defeito grave sendo que um defeito comum a todas é a não generalidade. Todas essas linguagens se limitam a resolver problemas relativos a sistemas de filas não abrangendo problemas de natureza contínua ou dinâmica.

O GPSS apresenta além disso um nível baixo de inteligibilidade por ser uma linguagem de estrutura muito próxima da de um assembler. Sendo assim sua estrutura de parâmetros é muito pouco prática.

O GASP por sua vez, não é propriamente uma linguagem mas sim um conjunto de subrotinas com limitações muito grandes para uso geral. Sua estrutura é portanto de nível baixo também.

O SIMSCRIPT é de todas as que apresenta as melhores qualidades. Ainda assim, devido a sua estrutura de montagem se assemelhar ao COBOL [MARKOWITZ, 1963], a escrita de programas mesmo que curtos nesta linguagem torna-se tediosa.

Ilustramos nos apêndices programas em GPSS, GASP e SIMSCRIPT que modelam o sistema de canal e fila simples descritos nas seções 1 e 2 com os parâmetros $\lambda=10$ e $\mu=12$ (taxas horárias de chegadas e serviço)

Com estes valores os resultados teóricos são os seguintes:

tempo médio de serviço	$\langle t_s \rangle = 5 \text{ min}$
comprimento médio da fila	$\langle l \rangle \approx 4$
tempo médio de espera na fila	$\langle w_q \rangle \approx 25 \text{ min}$
tempo médio gasto no sistema	$\langle s \rangle \approx 30 \text{ min}$
número médio de transações	$\langle n \rangle \approx 5$

4 - LGS (UMA LINGUAGEM GERAL DE SIMULAÇÃO)

Em [C.J. Lucena, H.M. Quintella, D. Schwabé, 76],

Foi apresentado o conceito de cadeias que permite o tratamento de dados abstratos num nível mais alto do que os que são empregados nas linguagens de simulação correntes. A partir disso passou-se a orientar o projeto de uma linguagem geral de simulação (LGS) em andamento no departamento de Informática da PUC no sentido de utilizar esse conceito como uma linguagem de base para o compilador dessa mesma linguagem. As características da estrutura interna desse compilador são apresentadas em [H.M. Quintella, E. Silva e V. Silva 76] e não cabem no escopo desse trabalho.

As vantagens aparentes dessa linguagem são:

- a) Clareza - por ser uma linguagem em português onde o programador tem que escrever por extenso os comandos.
- b) estruturação - ela estimula a confecção de programas naturalmente estruturados.

- c) generalidade - a linguagem resolve os problemas de sistemas de filas como um sub conjunto dos problemas que ela pode atacar.

5 - RESULTADOS DA SIMULAÇÃO

Na tabela I são apresentados os resultados em cada uma das linguagens utilizadas.

Claramente o primeiro protótipo do conjunto da LGS que trata de sistemas de filas apresenta uma performance estatística semelhante às das demais linguagens especiais. Ela tem no entanto as vantagens já discutidas na seção 4 e que seguramente a recomendariam para uso em ensino.

Resta por fim adiantar que outros trabalhos estão em andamento no sentido estender a LGS anexando a ela outros comandos igualmente poderosos.

distribuição de probabilidade	tempo total de simulação T_{\sim}	nº de eventos	comp. médio fila $\langle l \rangle$	tempo médio de espera na fila $\langle w \rangle$	tempo médio gasto no sist. $\langle s \rangle$	nº médio e-ventos no sistema $\langle n \rangle$
TEORIA exponencial	independente	independente	4	25	30	5
GPSS exponencial	100h	1000	$\approx 4(3.68)$	$\sim 25(27.703)$ (21.608)	$\sim 30(32.400)$ (26.305)	$\sim 5(4.68)$
SIMSCRIPT exponencial	140h	≈ 1400	$\approx 5(5.24)$	$\approx 31(31.40)$	$\approx 36(36.68)$	$\approx 6(6.13)$
LGS exponencial	100h	≈ 1000	$\approx 4(4.20)$	$\approx 25(26.40)$	$\approx 30(31.20)$	$\approx 5(5.01)$
GASP exponencial	155h	≈ 1550	$\approx 4(4.19)$	$\approx 27(27.087)$	$\approx 32(31.948)$	$\approx 5(5.19)$

TABELA 1

Apêndice - UM EXEMPLO NAS QUATRO LINGUAGENS

Como exemplo do uso das linguagens, considera-se um sistema de bilheteria de cinema com um único guichê. As chegadas estão geradas no intervalo médio de 10 segundos (6 por hora) e um desvio padrão de 3 segundos, e o serviço de atendimento está executado com um intervalo médio de 12 segundos (5 por hora) e um desvio padrão de 4 segundos.

1 - O EXEMPLO EM GASP

```
0001      DIMENSION NSET(12,25)
0002      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,NOQ,
01NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,ISEED,TNOW,TBEG,TFIN,
02MXX,NPRNT,NCRDR,NEP,VNQ(100)
0003      COMMON ATRIB(10),ENQ(100),INN(100),JCELS(10,32),KRANK(100),
11MAXNQ(100),MFE(100),MLC(100),MLE(100),NCELS(10),NQ(100),PARAM(40,4
12),QTIME(100),SSUMA(30,5),SUMA(30,5),NAME(6),NPROJ,MON,NDAY,NYR,
13JCLR
0004      COMMON /RDC/ XISYS,BUS,XL,XMU
0005      NPRNT=6
0006      NCRDR=5
0007      XISYS=1.
0008      BUS=1.
0009      XL=6.
0010      XMU=5.
0011      CALL GASP(NSET)
0012      CALL EXIT
0013      END
```

```
0001      SUBROUTINE EVNTS(IX,NSET)
0002      DIMENSION NSET(12,1)
0003      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,NOQ,
01NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,ISEED,TNOW,TBEG,TFIN,
02MXX,NPRNT,NCRDR,NEP,VNQ(100)
0004      COMMON ATRIB(10),ENQ(100),INN(100),JCELS(10,32),KRANK(100),
11MAXNQ(100),MFE(100),MLC(100),MLE(100),NCELS(10),NQ(100),PARAM(40,4
12),QTIME(100),SSUMA(30,5),SUMA(30,5),NAME(6),NPROJ,MON,NDAY,NYR,
13JCLR
0005      COMMON /RDC/ XISYS,BUS,XL,XMU
0006      GO TO (1,2,3),IX
0007      1 CALL ARRVL(NSET)
0008      RETURN
0009      2 CALL ENDSV(NSET)
0010      RETURN
0011      3 CALL ENDSM(NSET)
0012      RETURN
0013      END
```

```
0001 SUBROUTINE OPUT(NSET)
0002 DIMENSION NSET(12,1)
0003 COMMON ID,IM,INIT,JEVNT,JINIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,NOQ,
01NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,ISEED,TNOW,TBEG,TFIN,
02MXX,NPRNT,NCRDR,NEP,VNQ(100)
0004 COMMON ATRIB(10),ENQ(100),INN(100),JCELS(10,32),KRANK(100),
11MAXNQ(100),MFE(100),MLC(100),MLE(100),NCELS(10),NQ(100),PARAM(40,4
12),QTIME(100),SSUMA(30,5),SUMA(30,5),NAME(6),NPROJ,MON,NDAY,NYR,
13JCLR
0005 COMMON /RDC/ XISYS ,BUS,XL,XMU
0006 ETISS=SUMA(1,1)/SUMA(1,3)
0007 EIDTS=(SSUMA(2,1)-SSUMA(2,2))/(SUMA(1,3)-1.)
0008 EWTS=SUMA(2,1)/SUMA(2,3)
0009 EIDTC=XL-XMU
0010 EWTC=(1./XL)/((1./XMU)*(1./XMU-1./XL))
0011 ETISC=1./(1./XMU-1./XL)
0012 YA=ETISS/ (SSUMA(1,2)/SSUMA(1,1))
0013 YS=ETISS-EWTS
0014 WRITE(NPRNT,85)
0015 85 FORMAT(/46X,'SIMULATED VALUE',4X,'THEORETICAL VALUE'//)
0016 WRITE(NPRNT,90)EIDTS,EIDTC
0017 WRITE(NPRNT,95)EWTS,FWTC
0018 WRITE(NPRNT,96)ETISS,ETISC
0019 WRITE(NPRNT,97)YA,XL
0020 WRITE(NPRNT,98)YS,XMU
0021 RO=XMU/XL
0022 TAM=RO**2/(1-RO)
0023 TRSYS=RO/(1-RO)
0024 WRITE(NPRNT,88)
0025 WRITE(NPRNT,91)TAM,TRSYS
0026 ROCA=YS/YA
0027 TAMCA=ROCA**2/(1-ROCA)
0028 TRSYC=ROCA/(1-ROCA)
0029 WRITE(NPRNT,89)
0030 WRITE(NPRNT,91)TAMCA,TRSYC
0031 88 FORMAT(25X,'THEORETICAL VALUE'//)
0032 89 FORMAT(25X,'SIMULATED VALUE'//)
0033 91 FORMAT(//,20X,'QUEUE AVERAGE LENGTH',8X,F8.3/,20X,'AVERAGE NUMBER
1OF TRANSACTION IN SYSTEM',8X,F8.3)
0034 90 FORMAT(20X,'EXPECTED IDLE TIME',11X,F8.3,12X,F8.3)
0035 95 FORMAT(20X,'EXPECTED WAITING TIME',8X,F8.3,12X,F8.3)
0036 96 FORMAT(20X,'EXPECTED TIME IN SYSTEM',6X,F8.3,12X,F8.3)
0037 97 FORMAT(20X,'EXPECTED ARRIVAL TIME',8X,F8.3,12X,F8.3)
0038 98 FORMAT(20X,'EXPECTED SERVICE TIME',8X,F8.3,12X,F8.3)
0039 RETURN
0040 END
```

```
0001 SUBROUTINE ENDSV(NSET)
0002 DIMENSION NSET(12,1)
0003 COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,NOQ,
01NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,ISEED,TNOW,TBEG,TFIN,
02MXX,NPRNT,NCRDR,NEP,VNQ(100)
0004 COMMON ATRIB(10),ENQ(100),INN(100),JCELS(10,32),KRANK(100),
11MAXNQ(100),MFE(100),MLC(100),MLE(100),NCELS(10),NQ(100),PARAM(40,4
12),QTIME(100),SSUMA(30,5),SUMA(30,5),NAME(6),NPROJ,MON,NDAY,NYR,
13JCLR
0005 COMMON /RDC/ XISYS, BUS,XL,XMU
0006 TISYS=TNOW-ATRIB(3)
0007 CALL COLCT(TISYS,1,NSET)
0008 CALL HISTO(TISYS,2.0,1.0,1)
0009 CALL TMST(XISYS,TNOW,1,NSET)
0010 XISYS=XISYS-1.
0011 IF(NQ(2))7,8,9
0012 7 CALL ERROR(41,NSET)
0013 RETURN
0014 8 CALL TMST(BUS,TNDW,2,NSFT)
0015 BUS=0.
0016 RETURN
0017 9 MTEMP=MFE(2)
0018 CALL RMOVE(MTEMP,2,NSET)
0019 WT=TNOW-ATRIB(3)
0020 CALL COLCT(WT,2,NSET)
0021 CALL DRAND(ISEED,RNUM)
0022 ATRIB(1)=TNOW-XMU*ALOG(RNUM)
0023 ATRIB(2)=2.
0024 CALL FILEM(1,NSET)
0025 RETURN
0026 END
```

1 - CONTINUAÇÃO

```
0001 SUBROUTINE ARRVL(NSET)
0002 DIMENSION NSET(12,1)
0003 COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,NOQ,
01NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,ISEED,TNOW,TBEG,TFIN,
02MXX,NPRNT,NCRDR,NEP,VNQ(100)
0004 COMMON ATRIB(10),ENQ(100),INN(100),JCELS(10,32),KRANK(100),
11MAXNQ(100),MFE(100),MLC(100),MLE(100),NCELS(10),NQ(100),PARAM(40,4
12),QTIME(100),SSUMA(30,5),SUMA(30,5),NAME(6),NPROJ,MON,NDAY,NYR,
13JCLR
0005 COMMON /RDC/ XISYS ,BUS,XL,XMU
0006 CALL DRAND(ISEED,RNUM)
0007 ATRIB(1)=TNOW-XL*ALOG(RNUM)
0008 ATRIB(2)=1.
0009 CALL FILEM(1,NSET)
0010 CALL TMST(XISYS,TNOW,1,NSET)
0011 IF (XISYS)7,8,9
0012 7 CALL ERROR(31,NSET)
0013 RETURN
0014 8 XISYS=XISYS+1.
0015 CALL TMST(BUS,TNOW,2,NSET)
0016 BUS=1.
0017 CALL COLCT(0.,2,NSET)
0018 CALL DRAND(ISEED,RNUM)
0019 ATRIB(1)=TNOW-XMU*ALOG(RNUM)
0020 ATRIB(2)=2.
0021 ATRIB(3)=TNOW
0022 CALL FILEM(1,NSET)
0023 RETURN
0024 9 XISYS=XISYS+1.
0025 ATRIB(3)=TNOW
0026 CALL FILEM(2,NSET)
0027 RETURN
0028 END
```

1 - CONTINUAÇÃO

```
0001 SUBROUTINE ENDSM(NSET)
0002 DIMENSION NSET(12,1)
0003 COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,NOQ,
01NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,ISEED,TNOW,TBEG,TFIN,
02MXX,NPRNT,NCRDR,NEP,VNQ(100)
COMMON ATRIB(10),ENQ(100),INN(100),JCELS(10,32),KRANK(100),
11MAXNQ(100),MFE(100),MLC(100),MLE(100),NCELS(10),NQ(100),PARAM(40,4
12),QTIME(100),SSUMA(30,5),SUMA(30,5),NAME(6),NPROJ,MON,NDAY,NYR,
13JCLR
0005 COMMON /RDC/ XISYS ,BUS,XL,XMU
0006 20 IF(NQ(1))7,8,9
0007 7 CALL ERROR(3,NSET)
0008 8 CALL TMST(XISYS,TNOW,1,NSET)
0009 CALL TMST(BUS,TNOW,2,NSET)
0010 MSTOP=-1
0011 NORPT=0
0012 RETURN
0013 9 MTEMP=MFE(1)
0014 CALL RMOVE(MTEMP,1,NSET)
0015 TNOW=ATRIB(1)
0016 IF(ATRIB(2)-2.)20,21,20
0017 21 CALL ENDSV(NSET)
0018 GO TO 20
0019 END
```


1 - CONTINUAÇÃO (parametros de controle)

SIMULATION PROJECT NO. 1 BY FILA SGS VU

SCALE = 10.0000

VALORES LIDOS

NRUNS =	1						
CARTÃO TIPO 2		0	1	2	2	25	
		4	2	32	10.00		
CARTÃO TIPO 3							
30							
CARTÃO TIPO 4							
1	3						
CARTÃO TIPO 5							
1	1						
CARTÃO TIPO 6 - NÃO HÁ							
CARTÃO TIPO 7		0	1	0	0	0.0	
		100.000	567				
CARTÃO TIPO 8 - VER NSET							

2 - O EXEMPLO EM GPSS

1
2
3
4
5
6
7
8

```
          SIMULATE
EXPON FUNCTION      RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69
.6,.915/.7,1.2/.75,1.38/.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/.9997,8
          GENERATE  6, FN$EXPON
          QUEUE     1
          SEIZE     1
          DEPART    1
          ADVANCE   5, FN$EXPON
          RELEASE   1
          TABULATE  TRT
          TERMINATE 1
TRT      TABLE    M1,10,10,20
          START     1000
          END
```

3 - 0 EXEMPLO EM SIMSCRIPT

```

1      PREAMBLE
2      NORMALLY, MODE IS INTEGER
3      EVENT NOTICES INCLUDE ARRIVAL
4          EVERY END.OF.SERVICE HAS A CUSTOMER
5      TEMPORARY ENTITIES
6      EVERY PERSON HAS AN ATIME,
7          AND MAY BELONG TO THE QUEUE
8      DEFINE ATIME AS A REAL VARIABLE
9      THE SYSTEM OWNS THE QUEUE
10     DEFINE QUEUE AS A FIFO SET
11     DEFINE K, NIDLE AS VARIABLES
12     DEFINE LAMBDA, MU, TMAX AS REAL VARIABLES
13     DEFINE EL.TIME AND QTIME AS REAL VARIABLES
14     DEFINE NCUST AS A VARIABLE
15     ACCUMULATE LQ AS THE AVG OF N.QUEUE
16     ACCUMULATE L AS THE AVG OF NCUST
17     TALLY WQ AS THE AVG OF QTIME
18     TALLY W AS THE AVG OF EL.TIME
19     END.

```

```

1      MAIN
2      'INPUT'
3          READ K, LAMBDA, MU, AND TMAX
4          LET TMAX = TMAX/HOURS.V
5          LET NIDLE = K
6          SCHEDULE AN ARRIVAL NOW
7          START SIMULATION
8      'PRINT STATISTICS
9          IF LINP.V = 1, PRINT 4 LINES WITH PAGE.V THUS
SPEEDLINE BANK QUEUE SYSTEM PAGE **
NUMBER HOURLY HOURLY TOTAL AVERAGE AVERAGE AVERAGE AVERAGE
OF ARRIVAL SERVICE HOURS QUEUE IN QUEUE SYSTEM
TELLERS RATE RATE TIME LENGTH SYSTEM MINUTES MINUTES
10     ALWAYS
11     PRINT 1 LINE WITH
12     K, LAMBDA, MU, TMAX*HOURS.V, LQ, L, WQ*HOURS.V*MINUTES.V, AND
13     W*HOURS.V*MINUTES.V THUS
*     **.**. **.**. **. **.**. **.**. **.**. **.**.
14     IF DATA IS ENDED, STOP
15     ELSE
16     LET TIME.V = 0
17     RESET TOTALS OF N.QUEUE, NCUST, QTIME, AND EL.TIME
18     GO TO INPUT
19     END

```

LOCAL VARIABLES USED IN ABOVE ROUTINE

I.1	RECUR @ 56	I.2	RECUR @ 60	I.3
J.1	RECUR @ 68	K.1	RECUR @ 76	K.2
K.3	RECUR @ 84	K.4	RECUR @ 88	N.1
R.1	RECUR @ 52			

3 - Continuação

```
1      UPON ARRIVAL
2      CREATE PERSON
3      LET NCUST = NCUST + 1
4      LET ATIME = TIME.V
5      IF NIDLE = 0, FILE PERSON IN QUEUE
6          GO SCHEDULE.NEXT.ARRIVAL
7      ELSE
8          'TELLER IS FREE, SCHEDULE END OF SERVICE
9          LET NIDLE = NIDLE - 1 LET QTIME = 0.0
10         SCHEDULE AN END.OF.SERVICE(PERSON)
11             IN EXPONENTIAL.F(1./MU,1) HOURS
12     'SCHEDULE.NEXT.ARRIVAL'
13     SCHEDULE AN ARRIVAL IN EXPONENTIAL.F(1./LAMBDA,1) HOURS
14     IF TIME.A(ARRIVAL) > TMAX, CANCEL THIS ARRIVAL
15         DESTROY THIS ARRIVAL
16     REGARDLESS
17     RETURN
18     END
```

```
1      UPON END.OF.SERVICE(CUST)
2      LET EL.TIME = TIME.V - ATIME(CUST)
3      DESTROY PERSON CALLED CUST
4      LET NCUST = NCUST - 1
5      IF QUEUE IS EMPTY, LET NIDLE = NIDLE + 1
6          RETURN
7      ELSE
8      REMOVE FIRST PERSON FROM QUEUE
9      LET QTIME = TIME.V - ATIME
10     SCHEDULE AN END.OF.SERVICE(PERSON)
11         IN EXPONENTIAL.F(1./MU,1) HOURS
12     RETURN
13     END
```

LOCAL VARIABLES USED IN ABOVE ROUTINE

CUST RECUR @ 52

4 - O EXEMPLO EM L.G.S.

COMEÇO;	# INICIO DO EXEMPLO #
GERA TRANSAÇÕES	# GERA A CHEGADA DE PESSOAS#
VALOR MEDIO = 10	
DISTRIBUIÇÃO=UNIFORME	
DESVIO = 3;	
ENTRA TRANSAÇÃO NA FILA (FILAI);	# A PESSOA ENTRA NA FILA#
OCUPA CANAL (GUICHE);	# A PESSOA CHEGA A BILHETERIA#
TIRA TRANSAÇÃO DA FILA(FILAI);	# SAI DA FILA #
INCREMENTA RELOGIO	# OCUPA A BILHETERIA #
DISTRIBUIÇÃO = UNIFORME	
VALOR MEDIO = 12	
DESVIO = 4;	
LIBERA CANAL (GUICHE);	# LIBERA A BILHETERIA #
DESTROI (1) TRANSAÇÃO;	# ENTRA NO CINEMA#
EXECUTA (500) TRANSAÇÕES	# SIMULA COM 500 PESSOAS #
FIM;	

REFERÊNCIAS

- GORDON, G. "A general purpose systems simulation" IBM Systems Journal 1, 1962.
- KIVIAT, P.J. "Digital Computer Simulation: Computer Programming Languages". Rand Comp. RM-5883-PR, Jan. 1969.
- KLEINROCK, L. Queuing Systems, Wiley, 1975. Vol I.
- LUCENA, C.J., QUINTELLA, H., Schwabe, D., "On the modelling and representation of abstraction in simulation languages". IFIP Working Conference on modelling of environmental systems, Tokyo, Japan, 1976.
- MARKOWITZ, HAUSNER, KARR, "SIMSCRIPT- A Simulation Programming Language", Prentice Hall, 1963.
- NAYLOR, T.H. "Computer Simulation Experiments with models of economic systems", Wiley, 1971.
- PRITSKER AAB & KIVIAT, P.J. "Simulation with GASP V", Prentice Hall, 1969.
- QUINTELLA, H., SILVA, F., SILVA, V., "Estudos Preliminares para especificação e implementação de uma linguagem geral de simulação (LGS). Monografias em Ciência de Computação nº 2/77. DI/PUC, 1977.