

# Universidad Politécnica de Cartagena



## Escuela Técnica Superior de Ingeniería de Telecomunicación

### PRÁCTICAS DE MODELADO Y SIMULACIÓN

### BOLETÍN DE ENTREGA

### Práctica 3: Simulador de colas G/G/k

#### INTEGRANTES DEL GRUPO:

| NOMBRE Y APELLIDOS            | CORREO ELECTRÓNICO          |
|-------------------------------|-----------------------------|
| Diego Ismael Antolinos García | diego.antolinos@edu.upct.es |
| Andrés Ruz Nieto              | andres.ruz@edu.upct.es      |

#### Información y metodología de evaluación:

1. Esta práctica tiene un peso total del **10% en la evaluación de la asignatura**
2. El boletín debe rellenarse y convertirse a **formato PDF** para su envío a través de aula virtual, en las fechas dispuestas en el calendario
3. La evaluación se basará en las **respuestas del boletín** y también se contempla la posibilidad de realizar **entrevistas online** individuales para verificar la autoría de las mismas.

## 1. Implementación de las funciones *GMLC* y *aleatorio*

### COMPLETE LOS SIGUIENTES CÓDIGOS

```
[1 pto]
function [Z, muestra] = aleatorio(Z, tipo, param1, param2)
    Z = GCLM(Z);
    m = 2^31;
    u = Z/m;
    switch tipo
        case 0 % -> VA uniforme [0,1]
            muestra = u;
        case 1 % -> VA uniforme [param1,param2]
            muestra = param1 + u*(param2-param1);
        case 2 % -> VA exponencial lambda = param1
            muestra = -log(u)/param1
        case 3 % -> Devuelve siempre param1 (VA "degenerada")
            muestra = param1;
        case 4 % -> VA Bernoulli (devuelve 1 con probabilidad dada
por param1, sino 0)
            if u <= param1
                muestra = 1;
            else
                muestra = 0;
            end
    end
end
```

```
[1 pto]
function nuevoZ = GCLM(Z)
% Usando Z como muestra previa del generador, crea la nueva muestra.
% El GCLM debe usar los parámetros de referencia de Kobayashi
m = uint64(2^31);
a = uint64(314159269);
c = 453806245;
nuevoZ = double(mod((a*Z+c),m));
end
```

- a. Indique el valor del GCL tras 10000 iteraciones partiendo de Z=1 [0.5 pto]

```
for i=1:10000
    Z=GCLM(Z)
end
```

1.3562e+09

- b. Establezca Z=1 e indique el valor de las 10 primeras muestras de un generador de tipo Bernoulli [0.5 pto]

```
for i=1:10
    [Z, muestra] = aleatorio(Z, 4, 0.47, 0);
end
```

Como valor de p hemos escogido 0.47

Obtenemos los siguientes valores: 1 0 1 1 1 0 0 0 1 1

## 2. Implementación del simulador

**COPIE EL CÓDIGO COMPLETO DE SU SIMULADOR EN TEXTO (NO SE ADMITEN IMÁGENES) AL FINAL DEL DOCUMENTO**

- a) Indique el tiempo medio de respuesta obtenido para la configuración M/M/4 con  $\lambda=10$  tareas/s y  $\mu=2.6$  tareas/s con 1 millón de pasos de simulación. [1 pto]

```
TIEMPO MEDIO
2.6361
```

También hemos realizado la simulación con Fishman-Moore y nos da el siguiente resultado:

```
TIEMPO MEDIO
2.6752
```

Valide el resultado usando: [1 pto]

$$I = \frac{\lambda}{\mu}$$

$$\rho = \frac{I}{k}$$

$$p_0 = \left[ \frac{I^k}{k!(1-\rho)} + \sum_{n=0}^{k-1} \frac{I^n}{n!} \right]^{-1}$$

$$T = \frac{1}{\mu} + \frac{I^k}{\mu k k! (1-\rho)^2} p_0$$

$$I = 50/13$$

$$\rho = 25/26$$

$$p_0 = 507/131207$$

$$T = 2.674742963$$

- b) Calcule T para la configuración M/D/1, es decir, con llegadas Poissonianas de tasa  $\lambda$  y tiempo de servicio constante (recuerde que  $\mu=1/s$ ), dada por  $\lambda=3$  tareas/s y  $s=0.25$  s/tarea. [0.5 pto]

```
TIEMPO MEDIO
0.6259
```

También hemos realizado la simulación con Fishman-Moore y nos da el siguiente resultado:

```
TIEMPO MEDIO
0.6236
```

Valide el resultado usando: [1 pto]

$$\mu = \frac{1}{S}$$

$$\rho = \frac{\lambda}{\mu}$$

$$T = \frac{1}{\mu} + \frac{\rho}{2\mu(1-\rho)}$$

$$\mu = 4$$

$$\rho = \frac{3}{4}$$

$$T = 0.625$$

- c) Calcule T para G/G/3, con  $X \sim \text{Bernoulli}(p=0.25)$ ,  $S \sim \text{Bernoulli}(p=0.4)$  [1 pto]

```
TIEMPO MEDIO
0.7500
```

```
NÚMERO MEDIO DE CLIENTES EN EL SISTEMA
2.2748
```

También hemos realizado la simulación con Fishman-Moore y nos da el siguiente resultado:

```
TIEMPO MEDIO
0.7494
```

- d) Si ha realizado el ejercicio opcional indique el valor de N obtenido mediante simulación [2 pto]

MM4

```
NÚMERO MEDIO DE CLIENTES EN EL SISTEMA
26.3369
```

$26.3369/10=2.63369$  cercano al valor de T

MD1

```
NÚMERO MEDIO DE CLIENTES EN EL SISTEMA
1.8762
```

$1.8762/3=0.6254$

Valide el resultado usando: [0.5 ptos]

$N = \lambda T$  (LEY DE LITTLE)

```

%% ESQUELETO DE SIMULACION (EL SIMULADOR HA SIDO MODIFICADO PARA PODER ELEGIR
LA FORMA EN LA QUE SE QUIEREN OBTENER LOS Z)

listaEV = [];           % Lista vacia al comienzo
t_simulacion = 0.0;     % Reloj de simulación
pasos = 10000000;       % Numero de iteraciones del simulador

% ACCIONES DE INCIO: p.ej. definir estado, generar primeros eventos

% Se proporciona ejemplo del
% Caso cola de trabajos

% TIPOS DE EVENTOS, CADA UNO UN NUMERO DIFERENTE
SALE = 0;
LLEGA = 1;
MUESTREO = 2;

% ESTADO
N = 0;
fifoTiempos = [];

% PARAMETROS DE SIMULACION
%1 -> Fishman-Moore
%2 -> Kobayashi
%3 -> Coveyou-McPherson
%4 -> glibc
%5 -> MMIX
generadorZ = 2;
Z = 1;

%PARAMETROS PARA M/M/4 | lambda = 10 | mu = 2.6
% k = 4; % Numero de recursos
% tipoX = 2;
% param1X = 10;
% param2X = 0;
%
% tipoS = 2;
% param1S = 2.6;
% param2S = 0;

%PARAMETROS PARA M/D/1 | lambda = 3 | mu = 1 | s = 0.25
k = 1;
tipoX = 2;
param1X = 3;
param2X = 0;

tipoS = 3;
param1S = 0.25;
param2S = 0;

%PARAMETROS PARA G/G/3 | X~Bernoulli(p = 0.25) | S~Bernoulli(p = 0.4)
% k = 3;
% tipoX = 4;
% param1X = 0.25;
% param2X = 0;
%
```

```

% tipoS = 4;
% param1S = 0.4;
% param2S = 0;
% VARIABLES PARA EL CALCULO DE LOS PROMEDIOS DE INTERES
summuestrasT = 0;
muestrasT = 0;
summuestrasN = 0;
muestrasN = 0;

% PRIMEROS EVENTOS
[Z,taux] = aleatorio(Z,tipoX,param1X,param2X,generadorZ);
listaEV = encolarEvento(listaEV, taux, LLEGA,0);
[Z,taux] = aleatorio(Z,tipoX,param1X,param2X,generadorZ);
listaEV = encolarEvento(listaEV, taux, MUESTREO,0);

for i=1:pasos

    [listaEV, tiempo, tipo, tllegadatarea] = sgteEvento(listaEV);

    % Actualizamos el tiempo
    t_simulacion = tiempo;

    switch tipo
        case LLEGA
            N = N+1;
            [Z,taux] = aleatorio(Z,tipoX,param1X,param2X,generadorZ);
            listaEV = encolarEvento(listaEV, t_simulacion + taux, LLEGA,0);
            if N<=k
                [Z,taux] = aleatorio(Z,tipoS,param1S,param2S,generadorZ); %
Tiempo en el recurso
                listaEV = encolarEvento(listaEV, t_simulacion + taux, SALE,
t_simulacion);
            else
                fifoTiempos = pushFIFO(fifoTiempos,t_simulacion);
            end

        case SALE
            N = N-1;
            if N>=k % Otro trabajo pasa a ocupar el "procesador"
                [fifoTiempos, tllegadacola] = popFIFO(fifoTiempos); %
Recuperamos el primer tiempo en cola
                [Z,taux] = aleatorio(Z,tipoS,param1S,param2S,generadorZ); %
Tiempo en el recurso
                listaEV = encolarEvento(listaEV, t_simulacion + taux, SALE,
tllegadacola);
            end

            summuestrasT = summuestrasT + (t_simulacion - tllegadatarea);
            muestrasT = muestrasT + 1;

        case MUESTREO
            muestrasN=muestrasN+1;
            summuestrasN=summuestrasN+N;
            [Z,taux] = aleatorio(Z,tipoX,param1X,param2X,generadorZ);
            listaEV = encolarEvento(listaEV, t_simulacion + taux,
MUESTREO,tllegadatarea);
        end
    end
end

```

```
% Mostramos los promedios calculador
display('FIN DE LA SIMULACION');
[i, summuestrasT, muestrasT]
display('TIEMPO MEDIO');
disp(summuestrasT / muestrasT);
display('NÚMERO MEDIO DE CLIENTES EN EL SISTEMA');
disp(summuestrasN/muestrasN);
```