

Mask: diccionario de máscaras de cada fuente. Proporciona el estado de las fuentes de
sucesos.
Mask['A'] = estado de la fuente A (=1 activada o =0 desactivada). Id. para B, C
St: diccionario que proporciona el estado de la parada:
St['pax'] = # pasajeros en parada, St['cap'] = # plazas en el autobús. Si =0 es que no ha
llegado todavía bus.
(Mask, St) implementan el estado del sistema.
La lista de sucesos (LISTA) está formada por elementos tipo diccionario Lista['suc'], Lista['t']
Programa principal.
inicialización lista de sucesos con la llegada de un primer pasajero y la de un primer bus.

- tCK, Mask, St, LISTA deberían ser globales
- Generar al azar tA; Lista['suc'] = A; Lista['t'] = tA; "encolar" Lista en LISTA;
- Generar al azar tB; Lista['suc'] = B; Lista['t'] = tB; "encolar" Lista en LISTA;
- tCK = 0;
- St['pax'] = 0; St['cap'] = 0; # 0 pax, 0 capacidad
- Mask['A'] = 1; Mask['B'] = 1; Mask['C'] = 0;

Mientras tCK < T

tCK0 = tCK; St0 = St; Mask0 = Mask;
Lista ← Inicio.lista(LISTA);
tCK = Lista['t'];
si Lista['suc'] = A entonces llamar a succA;
si Lista['suc'] = B entonces llamar a succB;
si Lista['suc'] = C entonces llamar a succC;
se reporta en traza la iteración
Escribir: tCK0, St0, Mask0, Lista['suc'], tCK, St, Mask;

Funcion succA

#Generar siguiente pasajero:
Generar al azar tA; ELEM = ('A', tCK + tA); "encolar" ELEM en LISTA;
St['pax'] = St['pax'] + 1;

Funcion succB

#Generar siguiente llegada de Bus:
Generar al azar tB; ELEM = ('B', tCK + tB); "encolar" ELEM en LISTA;
Si St['pax'] > 0 & St['cap'] = 0 entonces
Generar c ; # (= capacidad del bus entrante)
St['cap'] = c;
Si c > 0 entonces
Mask['C'] = 1; # se activa fuente C
Generar al azar tC; ELEM = ('C', tCK + tC); "encolar" ELEM en LISTA;

Funcion succC

Si St['pax'] > 0 entonces
St['pax'] = St['pax'] - 1; St['cap'] = St['cap'] - 1;
Si St['pax'] = 0 entonces
St['pax'] = 0; Mask['C'] = 0;
De otro modo
Generar al azar tC; ELEM = ('C', tCK + tC); "encolar" ELEM en LISTA;
De otro modo
Reportar error: "subida con St['pax'] = 0" ;