Universidad de Costa Rica

Escuela de Ciencias de la Computación e Informática

CI-1453

Investigación de Operaciones

Profesora: Ileana Alpízar

Proyecto 1 Simulación

Erick Guillén

Amanda Sagasti

Jose Víquez

30 de setiembre del 2015

**MANUAL DE INSTALACIÓN**

Para la instalación del programa lo que se necesita es una computadora con algún comilador, recomendamos gcc ya que es el que nosotros utilizamos. Una vez teniendo los archivos que componen el programa, se debe ejecutar una terminal e ir a la dirección donde se encuentra los archivos. Una vez en la dirección, se corre el comando *g++ -std=c++11 main.cpp Computadora.h Simulacion.h Util.h Computadora.cpp Simulacion.cpp Util.cpp –o sim* (Sim es solo el nombre del ejecutable, se puede escoger el que se prefiera). Habiendo compilado con éxito, se ejecuta *sim* en Windows y *./sim* en Linux y se contestan las preguntas referentes a como se desea ejecutar el programa.

**ENUNCIADO**

Suponga que se tiene una LAN (Local Area Network) compuesta por tres computadoras A, B, C. La capa de red de cada máquina recibe archivos para su envío hacia máquinas fuera de la red. Los archivos recibidos por cada una de estas computadoras deben ser enviados al "Servidor Antivirus" para que descarte aquellos que lleven virus, el resto de los archivos son enviados por este servidor al "Router" del sistema para que se encargue de enviarlos a su destino.

Hay una sola línea de transmisión que deben compartir las 3 computadoras para enviar sus archivos al Servidor Antivirus. La transferencia de un archivo desde el Antivirus al Router, se hace inmediata, o sea, en tiempo "0".

La manera de utilizar la única línea de transmisión por estas 3 máquinas para el envío de sus archivos al Antivirus sigue un protocolo de polling o Token Ring1

Este protocolo funciona de la siguiente manera:

1. En cualquier momento dado solamente una de las máquinas de la red puede transmitir información a través del medio (el cable).

2. El protocolo maneja un token que es asignado durante un período de tiempo T a cada máquina de manera secuencial.

3. Durante ese tiempo T, la máquina que tiene el token tiene acceso exclusivo al cable o línea de transmisión. Durante este período debe transmitir los archivos que le dé tiempo de transmitir, siempre respetando la etiqueta de prioridad del archivo y prefiriendo transmitir los archivos más grandes primero.

4. Si una máquina no tiene archivos que transmitir entonces se libera el token de forma inmediata y se asigna a la siguiente máquina. Si después de enviar al menos un archivo, el remanente de tiempo es menor al archivo de menor tamaño en cola, el token se libera y se asigna de inmediato a la siguiente máquina.

La máquina A recibe archivos con una distribución de su tiempo entre arribos exponencial con una media de 5 segundos.

La máquina B recibe archivos con la siguiente distribución para el tiempo entre arribos: f(x) = x/40 8 <= x <= 12 segundos

La máquina C recibe archivos con una distribución entre arribos normal, con una media de 4 segundos y una varianza de 0.01 segundos cuadrados o sea es n(μ=5, σ2=0.01).

Cada archivo recibido tiene una etiqueta de prioridad (1 ó 2), en donde los archivos con prioridad 1 son más importantes y deben ser transmitidos de forma prioritaria. Cada archivo tiene una probabilidad de 0.25 de ser de prioridad 1.

Suponga que cuando se recibe un archivo en una de las tres computadoras, éste se divide en paquetes de 1024 k- bits, si el último paquete tiene menos de 1024 k-bits, entonces se completa con 0 ́s para alcanzar dicha cantidad de bits.

El tamaño de cada archivo (M), medido en número de paquetes de 1024 kb (k-bits), es uniforme entre 1 y 64. Es decir, tiene distribución uniforme en {1, 2, 3, ..., 64}.

El proceso de transferir a la línea cada paquete tarda, en cualquiera de las 3 computadoras, 1/2 segundo. Una vez puesto en la línea el último paquete del archivo que está transmitiendo, el tiempo que tarda en llegar el último bit de ese paquete al ANTIVIRUS es de 1 segundo (tiempo de propagación) Es hasta ese momento que se considera que el archivo llegó a dicho servidor.

El servidor Antivirus tiene suficiente espacio para colocar en cola todos los paquetes que le vayan llegando antes de revisarlos y descartar el archivo o enviarlo al Router.

Cuando el proceso de revisión por virus de un archivo encuentra un virus, "lo elimina" y procede a realizar la revisión completa de nuevo. La probabilidad de encontrar un virus en una revisión es 0.05 (independiente del número de veces que se halla realizado la revisión). El tiempo que se tarda realizando la revisión número "I" del archivo, se ha estimado en M/8\*I segundos, donde M es el tamaño del archivo. Si a un archivo se le encuentra un virus en 3 revisiones, el archivo se descarta.

Finalmente, cuando un archivo pasa la revisión del antivirus, es enviado al Router de la red (vamos a asumir que el tiempo que dura es en ser enviado y llegar al router despreciable y se contabiliza como 0).

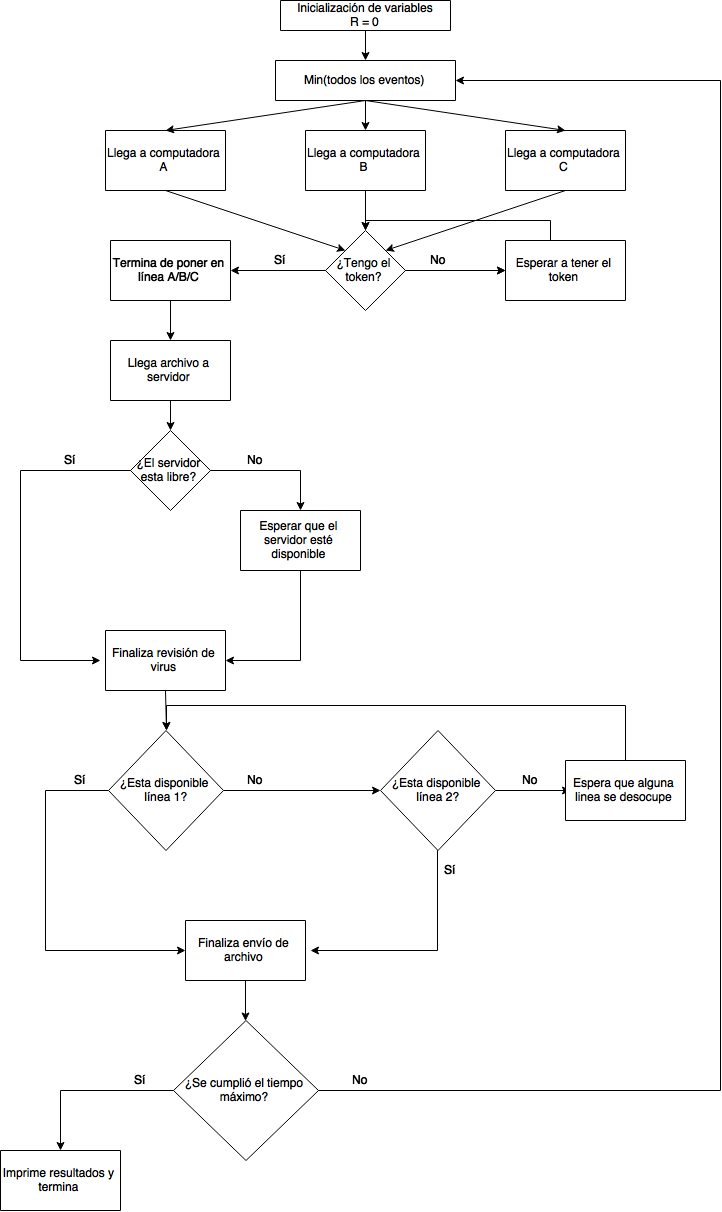
El Router tiene disponibles 2 líneas de transmisión los cuales son utilizados para poner los archivos en la red (uno por archivo). El tiempo que se dura en transferir a una línea un archivo es de M/64 segundos, donde M es el tamaño del archivo. Pasado ese tiempo, el archivo es considerado como enviado. El router sin embargo, es uno barato, y no tiene buffers para almacenar archivos en espera cuando todos los hilos están ocupados, por lo que cuando el antivirus termina un archivo, debe revisar el status del router, para saber si tiene hilos disponibles. En caso afirmativo, el archivo es pasado al router, en caso contrario, el archivo se mantiene en el antivirus hasta que haya uno libre.

**LISTA DE EVENTOS**

Unidad de tiempo: 1 segundo

|  |  |
| --- | --- |
| **Evento** | **Valor inicial** |
| Llega a computadora A | 0 |
| Llega a computadora B | 0 |
| Llega a computadora C | 0 |
| Libera token A | 1 |
| Libera token B | ∞ |
| Libera token C | ∞ |
| Termina de poner en línea archivo A | ∞ |
| Termina de poner en línea archivo B | ∞ |
| Termina de poner en línea archivo C | ∞ |
| Llega a servidor antivirus | ∞ |
| Finaliza revisión de virus | ∞ |
| Se libera línea 1 router | ∞ |
| Se libera línea 2 router | ∞ |

**DIAGRAMA DE FLUJO**

****

**GENERACIÓN DE VALORES ALEATORIOS**

* **Prioridad de archivo (Prioridad 1 o Prioridad 2)**

El 25% de los archivos que llegan a las computadoras son de Prioridad 1 y el 75% son de Tipo 2. Sea X la variable aleatoria que toma el valor de la prioridad de los archivos.

La función f(x) = P(X = x) está dada por P(x = 1) = 0.25 y P(x = 2) = 0.75. Para generar un valor aleatorio para X se genera un valor entero con distribución uniforme entre 0 y 99 y según el rango en el que esté el valor en la tabla siguiente, se determina el valor x de X.

x ε {1, 2}

|  |  |  |  |
| --- | --- | --- | --- |
| x | f(x) | F(x) | Rango |
| 1 | 0.25 | 0.25 | 0 – 24 |
| 2 | 0.75 | 1 | 25 – 99 |

* **Tasa de arribos a computadora A**

La máquina A recibe archivos con una distribución exponencial con una media de 5 segundos. Su función de distribución es de

con *λ* = 1/10

Para generar un valor aleatorio a la tasa de arribos en la computadora A se utiliza el Método de la Transformación Inversa. (Winston, 2004, p.1162)

Paso 1: Calcular F(x) como la integral de f(x)

Paso 2: Generar un valor aleatorio *r* con distribución uniforme entre [0,1]

Paso 3: F(x) = r para aislar x. La variable x es una variación aleatoria para la distribución exponencial.

* **Tasa de arribos a computadora B**

La máquina B recibe archivos con la distribución

Para generar un valor aleatorio para esta distribución se utiliza el Método de la Transformación Inversa.

Paso 1: Calcular F(x) como la integral de f(x)

Paso 2: Generar un valor aleatorio *r* con distribución uniforme entre [0,1]

Paso 3: F(x) = r para aislar x. La variable x es una variación aleatoria para la distribución.

* **Tasa de arribos a computadora C**

La máquina C recibe archivos con una distribución normal con una media de 4 segundos y una varianza de 0.01 segundos cuadrados n(*μ*=5, *σ2*=0.01).

Hay dos métodos para generar valores aleatorios para la distribución normal. Ambos métodos utilizan la relación donde *Z* es la normal estándar (Z es n(0,1)) y *X* es . De esta manera se genera un valor *z* para *Z* y el valor *x* para *X* se calcula como .

Método 1: Convolución

Este método utiliza el Teorema Central del Límite[[1]](#footnote-1) a 12 valores aleatorios *r* ε [0,1] con media y varianza . Esto resulta en

Donde Z es normal estándar.

Después de generar *z*, se puede aplicar la relación para aislar el valor correspondiente de *x.*

Método 2: Directo

En este método se generan dos valores aleatorios . Luego *r*1 y *r*2 se transforman en variables aleatorias normales estándar con las siguientes transformaciones directas:

Luego Z1 y Z2 se transforman en variables aleatorias normales con la relación

* **Tamaño de cada archivo**

El tamaño de cada archivo se mide en número de paquetes de 1024 kb (k-bits) y es uniforme entre 1 y 64. Sea X la variable aleatoria que toma el valor del número de virus que tiene un archivo.

x ε {1, 64}

|  |  |  |
| --- | --- | --- |
| x | f(x) | F(x) |
| 1 | 0.15 | 0.15 |
| … | … | … |
| 64 | 0.15 | 1 |

* **Probabilidad de que encuentre virus**

El 95% de los archivos que pasan por el servidor antivirus no tienen ningún virus, mientras el 5% tienen un virus. Sea X la variable aleatoria que toma el valor del número de virus que tiene un archivo.

La función f(x) = P(X = x) está dada por P(x = 0) = 0.95 y P(x = 1) = 0.05. Para generar un valor aleatorio para X se genera un valor entero con distribución uniforme entre 0 y 99 y según el rango en el que esté el valor en la tabla siguiente, se determina el valor x de X.

x ε {0, 1}

|  |  |  |  |
| --- | --- | --- | --- |
| x | f(x) | F(x) | Rango |
| 0 | 0.95 | 0.95 | 0 – 94 |
| 1 | 0.05 | 1 | 95 – 99 |

**ESTADÍSTICAS**

Al inicio, se le solicita al usuario los siguientes parámetros:

* Número de veces que va a correr la simulación: se inicializa una variable entera númeroDeVeces para que el usuario lo determine.
* Tiempo total que va a correr la simulación:
* Si desea ver la simulación correr en “modo lento” (delay de 1 segundo entre cada cambio de reloj): se inicializa una variable booleana modoLento = 0. Si se cambia el valor a 1, la simulación corre en modo lento.
* El tiempo T que cada máquina tiene el token: se inicializa una variable tiempoToken donde el usuario puede definir el tiempo que cada máquina tiene el token.

Durante la simulación, se despliegan en pantalla las siguientes estadísticas:

* Reloj del sistema: se utiliza una variable Reloj en la simulación para desplegarla en pantalla.
* Longitud de las colas de las 3 máquinas (dividido por prioridad)
* Longitud de la cola del antivirus
* La máquina que tiene el token
* El tiempo T durante el cual se le asigna el token
* La cantidad de hilos ocupados en el router
* El tipo de evento que se está procesando

Al final de la simulación, se muestran los siguientes resultados:

* Tamaño promedio de la cola en las 3 máquinas
* Tamaño promedio de la cola en el antivirus
* Tiempo promedio de “permanencia de un archivo en el sistema” (desde que llega a una computadora y sale del router) en general y por prioridad
* Promedio de archivos enviados por cada turno del token
* Número promedio de revisiones del antivirus por archivo

**ANÁLISIS DEL SISTEMA SIMULADO**

1. ¿Qué pasa si el tiempo de transmisión de los servidores A, B y C se incrementa?
2. ¿Qué pasa si el tamaño máximo de los archivos se duplica?
3. ¿Qué pasa si se añade memoria al router?
   1. Nada
4. ¿Cómo se podría mejorar la eficiencia? (reduciendo el tiempo promedio que pasa un archivo en el sistema)

**PROBLEMAS NO RESUELTOS**

**FUENTES**

Winston, W. (2004). *Operations Research: Applications and Algorithms* (4th ed.). Canada: Brooks/Cole.

1. La suma Y de *n* variables aleatorias independientes e idénticamente distribuidas es aproximadamente normal con media y varianza . [↑](#footnote-ref-1)