

Experto A

El experimento y la prueba experimental

Un experimento se describe en un fichero ASCII con extensión *.exp*. Un experimento consiste de varias pruebas experimentales que serán comparadas y analizadas. Una prueba experimental indica un algoritmo de planificación y un conjunto de procesos. Se describe en un fichero ASCII con extensión *.run*.

La prueba experimental

En su forma más simple, el formato del fichero que describe una prueba experimental es el siguiente:

name nombre_de_la_prueba_experimental
comment descripción_de_la_prueba_experimental
algorithm descripción_del_algoritmo
numprocs número_de_procesos
firstarrival tiempo_de_la_primera_llegada
interarrival distribución_entre_llegadas
duration distribución_para_la_duración
cpuburst distribución_de_las_ráfagas_de_cpu
ioburst distribución_de_las_ráfagas_de_IO
basepriority prioridad

A continuación se muestra el contenido de un fichero ejemplo que contiene la especificación para una prueba experimental:

```
name myrun
comment Esto es un ejemplo de especificación de una prueba experimental
algorithm SJF
numprocs 20
firstarrival 0.0
interarrival constant 0.0
duration uniform 500.0 1000.0
cpuburst constant 50.0
ioburst constant 1.0
basepriority 1.0
```

El fichero especifica que se ejecutarán un total de 20 procesos planificados según el algoritmo SJF. El primer proceso llega en el instante 0,0. Los tiempos entre llegadas son todos 0,0 lo que indica que todos llegan al mismo tiempo. Cada proceso tiene una duración (tiempo total de uso de CPU) elegida desde una distribución uniforme en el intervalo de 500 a 1000. Todos los procesos tienen un tiempo de ráfaga de CPU constante e igual a 50 y un pequeño tiempo de ráfaga de I/O constante e igual a 1. La prioridad especificada en el fichero solo se utiliza cuando se configura el simulador para que utilice prioridades.

El experimento

El formato del fichero que describe un experimento contiene, además del nombre y una línea para comentarios, otro conjunto de líneas especificando las pruebas experimentales a realizar. Dichas pruebas se indican con el nombre del fichero que la contiene y, posiblemente, algunos parámetros. Por ejemplo, quizás esté interesado en

ejecutar el mismo conjunto de procesos con el algoritmo Round Robin pero variando el quantum.

El formato de un fichero que especifica un experimento es el siguiente:

```
name nombre_del_experimento
comment descripción_del_experimento
run nombre_de_la_prueba_experimental lista_opcional_de_parámetros
...
run nombre_de_la_prueba_experimental lista_opcional_de_parámetros
```

A continuación se muestra el contenido de un fichero ejemplo que contiene la especificación para un experimento:

```
name myexp
comment este experimento especifica 3 pruebas experimentales
run myrun
run myrun cpuburst uniform 10 90
run myrun cpuburst exponential 50
```

Este experimento describe tres pruebas experimentales. Todas las pruebas están basadas en la descripción contenida en el fichero *myrun.run*. En la segunda prueba se modifica la distribución de las ráfagas de CPU a una distribución uniforme en el intervalo 10..90. En la tercera prueba se modifica a una distribución exponencial con valor medio de 50.

Las posibles modificaciones que se pueden indicar sobre una prueba experimental y que pueden estar descritas en un fichero experimento son las siguientes:

- **numprocs**: el parámetro es el número de procesos a crear.
- **firstarrival**: el parámetro, un número en coma flotante, indica el tiempo de llegada del primer proceso.
- **basepriority**: el parámetro es la prioridad base del proceso creado. La prioridad base no es útil a menos que el simulador esté utilizando prioridades.
- **interarrival**: el parámetro es un string que representa la distribución entre los tiempos de llegada de los procesos.
- **duration**: el parámetro es un string que representa la distribución del tiempo total de CPU utilizado por los procesos.
- **cpuburst**: el parámetro es un string que representa la distribución de los tiempos de las ráfagas de CPU de los procesos.
- **ioburst**: el parámetro es un string que representa la distribución de los tiempos de las ráfagas de I/O de los procesos.
- **algorithm**: el parámetro es un string que representa el algoritmo de planificación.
- **seed**: el parámetro es un entero que representa la semilla a utilizar para inicializar el generador de números aleatorios.
- **priorityoff**: indica al simulador que no utilice prioridades.
- **priorityon**: indica al simulador que utilice una política de prioridades no expulsora.
- **prioritypreempt**: indica al simulador que utilice una política de prioridades expulsora.
- **key**: el parámetro indica la clave a utilizar en un gráfico de tiempos de espera. Si el parámetro contiene más de una palabra, éstas deben estar encerradas entre dobles comillas.

Experto B

1.1. Números aleatorios

Java tiene un generador de número pseudoaleatorios que se utiliza cuando hay que seleccionar valores desde una distribución de probabilidad. Uno de los problemas de utilizar este generador es que a menos que todas las distribuciones sean constantes, diferentes ejecuciones sobre los mismos datos producirán resultados ligeramente distintos. Pero a veces es interesante repetir el mismo experimento para analizarlo en más profundidad. Poniendo la opción portable a true permite hacer esto. El simulador utiliza un generador portable de números pseudoaleatorios con una semilla fija tal que si se repite el mismo experimento con los mismos datos, producirá los mismos resultados. La semilla se inicializa al comienzo de cada prueba experimental, tal que una prueba no se vea afectada por otras anteriores.

El simulador utiliza un generador de números aleatorios para calcular todas las distribuciones. Las ráfagas de CPU y de I/O de los procesos se calculan sobre la marcha para cada proceso. Los tiempos de ráfagas de CPU se calculan cuando el proceso entra en la cola de preparados. Puesto que el algoritmo de planificación puede afectar al orden en el que los procesos entran en la cola de preparados, el mismo conjunto de procesos ejecutándose bajo diferentes algoritmos de planificación tendrán ráfagas de duración diferente.

1.2. Procesos

Un proceso se especifica describiendo la siguiente información:

- Tiempo de llegada.
- Tiempo total de CPU.
- Distribución del tiempo de ráfagas de CPU.
- Distribución del tiempo de ráfagas de I/O.

1.3. Distribuciones de probabilidad

El simulador soporta actualmente tres tipos de distribuciones de probabilidad. Cuando se especifica una distribución de probabilidad en un fichero ASCII se representa en una única línea de texto. La línea comienza con una palabra indicando el tipo de distribución y va seguida por o bien un número en coma flotante representando la media de la distribución o, en el caso de la distribución uniforme, dos números en coma flotante representando las cotas inferior y superior del intervalo. A continuación se indica cómo especificar estas distribuciones:

- La distribución constante. Ejemplo:
constant 23.45
- La distribución exponencial. Ejemplo:
exponential 23.45
- La distribución uniforme. Ejemplo:
uniform 23.45 47.89

1.3.1. Distribución constante

En una distribución constante el valor de la variable aleatoria siempre es el mismo.

1.3.2. Distribución exponencial

La función de densidad tiene la forma:

$$f_x(x) = \begin{cases} \lambda e^{-\lambda x}, & \text{para } x \geq 0; \\ 0, & \text{en otro caso} \end{cases} \quad (1)$$

donde λ ($\lambda > 0$) es el parámetro de la distribución. La función de distribución tiene la forma siguiente:

$$F_x(x) = \begin{cases} 1 - e^{-\lambda x}, & \text{para } x \geq 0; \\ 0, & \text{en otro caso} \end{cases} \quad (2)$$

La media, m_x , y la varianza, σ_x^2 , son:

$$m_x = \frac{1}{\lambda}, \quad \sigma_x^2 = \frac{1}{\lambda^2} \quad (3)$$

1.3.3. Distribución uniforme

Una variable aleatoria continua X tiene una distribución uniforme sobre un intervalo de a..b ($b > a$) si la probabilidad de que tome cualquier valor de dicho intervalo es la misma. La función de densidad tiene la forma:

$$f_x(x) = \begin{cases} \frac{1}{b-a}, & \text{para } a \leq x \leq b; \\ 0, & \text{en otro caso.} \end{cases} \quad (4)$$

La función de distribución se obtiene integrando la fórmula (4) anterior, obteniéndose la expresión siguiente (5):

$$F_x(x) = \begin{cases} 0, & \text{para } x < a; \\ \frac{x-a}{b-a}, & \text{para } a \leq x \leq b; \\ 1, & \text{para } x > b; \end{cases} \quad (5)$$

La representación gráfica de ambas funciones se indica en la Figura 1.

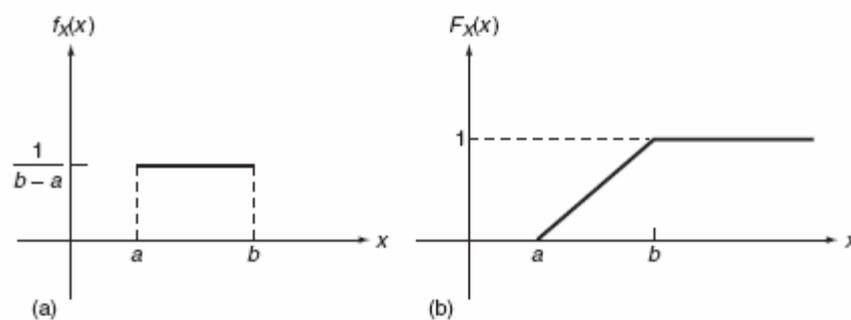


Figura 1. (a) Función de densidad y (b) función de distribución

La media, m_x , y la varianza, σ_x^2 , de X se hallan como:

$$m_x = \int_a^b x f_x(x) dx = \frac{1}{b-a} \int_a^b x dx = \frac{a+b}{2}; \quad (6)$$

$$\sigma_x^2 = \frac{1}{b-a} \int_a^b \left(x - \frac{a+b}{2} \right)^2 dx = \frac{(b-a)^2}{12}$$

Experto C

La interfaz del simulador

La ventana principal del simulador se muestra en la Figura 1. En la parte superior hay dos áreas de texto etiquetadas con History y con Event List. Cada una de estas dos áreas tiene asociados 5 botones. El del medio (el más grande) visualiza (on) o no (off) el contenido del área de texto. El botón a su izquierda (Clr) borra el contenido del área de texto mientras que el botón de la derecha (Log) almacenada en el fichero log la información visualizada. Los dos botones pequeños en los extremos (etiquetados con < y >) disminuyen y aumentan respectivamente el tamaño del font del texto visualizado en la ventana.

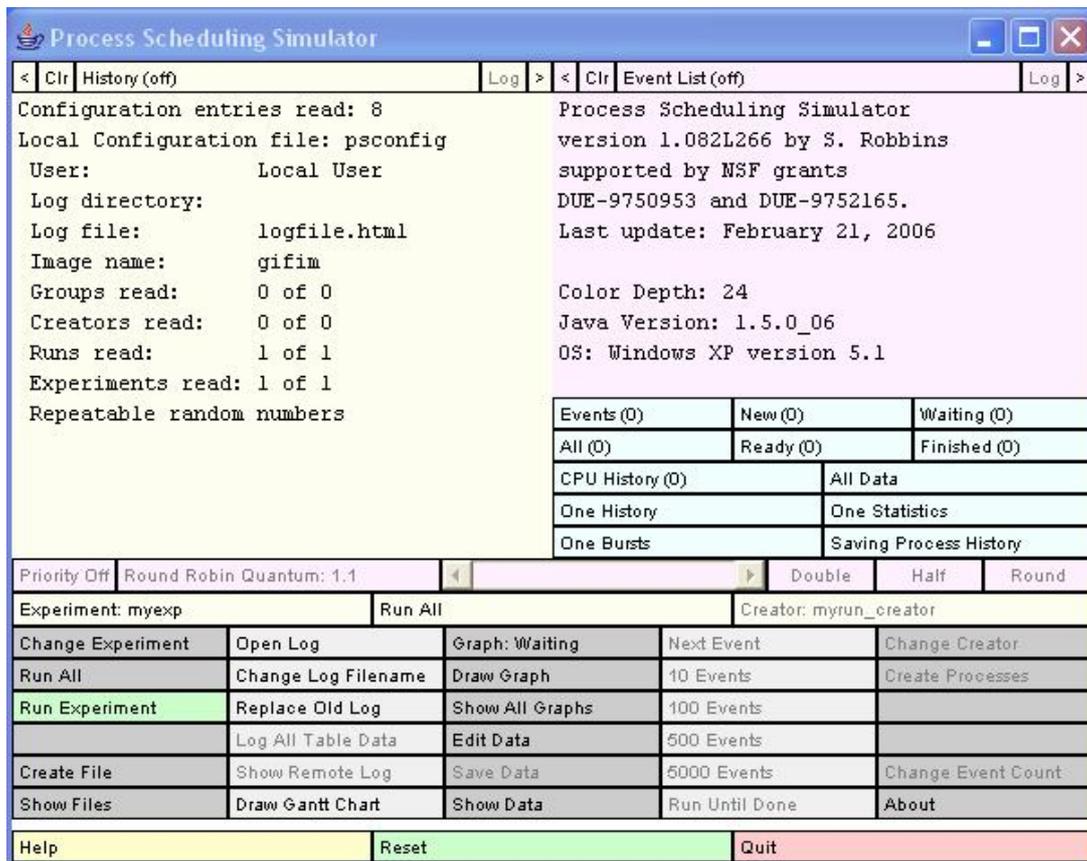


Figura 1. Ventana principal del simulador

El área de texto del ángulo superior izquierdo etiquetada History muestra la configuración inicial leída del fichero de configuración así como información variada acerca del progreso de la simulación. El área de texto de más a la derecha etiquetada Event List muestra los eventos según van ocurriendo.

Información sobre procesos y eventos

Debajo de la ventana Event List hay 12 botones para visualizar información sobre procesos y eventos relativos a la última ejecución. La función de cada botón no es de interés para este ejercicio.

Controles sobre el experimento y el algoritmo de planificación

Debajo de esos botones se especifica el algoritmo de planificación actual así como el experimento que se está llevando a cabo (ver Figura 2). El conjunto de botones que aparecen en la Figura 2 están inhibidos. Solo están habilitados cuando el experimento se ejecuta de manera interactiva (Interactive Run), escenario éste que no es de interés para este ejercicio.



Figura 2. Botones para especificar el algoritmo y el experimento a ejecutar

Controles para la ejecución de experimentos

Más abajo existen 5 columnas de botones. La primera columna (la de más a la izquierda) contiene botones relacionados con la ejecución de experimentos y creación/visualización de ficheros de configuración. La funcionalidad que nos interesa para este ejercicio es la asociada a los botones:

- Run Experiment. Este botón arranca la ejecución del experimento.
- Show Files. Visualiza el menú pop-up indicado en la Figura 3.

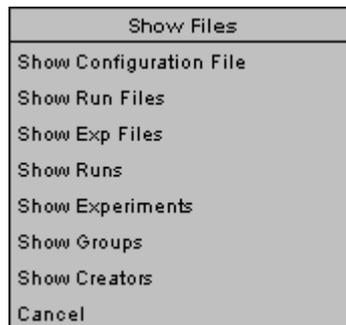


Figura 3. Menú pop-up asociado al botón Show Files

La funcionalidad asociada con las opciones del menú anterior son las siguientes:

- Show Configuration File. Visualiza en el área de texto History el contenido del fichero de configuración.
- Show Run Files. Visualiza en el área de texto History el contenido de los ficheros que describen las pruebas experimentales.
- Show Exp Files. Visualiza en el área de texto History el contenido de los ficheros que describen los experimentos.
- Show Runs. Visualiza en el área de texto History el número de pruebas experimentales y algunas propiedades de éstas.
- Show Experiments. Visualiza en el área de texto History el número de experimentos y algunas propiedades de éstos.
- Show Groups y Show Creators. Permiten gestionar los parámetros del simulador a más bajo nivel y no son de interés para el ejercicio actual.
- Cancel. Cierra el menú.

Experto D

Controles para el fichero log

La segunda columna de botones de la Figura 1 (experto C) controla el fichero log sobre el que se registran los resultados del experimento y los eventos del propio experimento. La funcionalidad que nos interesa para este ejercicio es la asociada a los botones:

- Open Log. Abre el fichero log y habilita el resto de botones. Después de pulsarlo su etiqueta cambia a Close Log. Si cierra el fichero log y se vuelve a abrir, se elimina el contenido anterior del fichero.
- La funcionalidad del tercer botón depende de si el fichero log está abierto o cerrado. Cuando está abierto el fichero log, la etiqueta de este botón es Log Comment. Cuando se pulsa sobre el botón se visualiza una ventana que permite al usuario introducir un comentario en el fichero log.
- Log All Table Data. Si el fichero log está cerrado, este botón está deshabilitado. Crea dos tablas en el fichero log que contienen estadísticas sobre los datos salvados. Los datos se salvan de manera automática cuando acaba un experimento o cuando se pulsa sobre el botón Save Data, pero no aparecen en el fichero log hasta que se pulsa Log All Table Data. Las tablas contienen una entrada para cada prueba experimental ejecutada. Un ejemplo de tales tablas se ilustra en la Figura 1.

| | | | | | | | Entries | | Average Time | |
|-------------|-----------|--------|-----------|----------|-----------------|------------|---------|-----|--------------|-------|
| Description | Algorithm | Time | Processes | Finished | CPU Utilization | Throughput | CPU | I/O | CPU | I/O |
| myrun_1 | FCFS | 257.05 | 30 | 30 | .96104 | .116711 | 90 | 60 | 2.74 | 15.20 |
| myrun_2 | SJF | 256.90 | 30 | 30 | .96158 | .116777 | 90 | 60 | 2.74 | 15.20 |

| | | Turnaround Time | | | | Waiting Time | | | |
|-------------|-----------|-----------------|---------|---------|-------|--------------|---------|---------|------|
| Description | Algorithm | Average | Minimum | Maximum | SD | Average | Minimum | Maximum | SD |
| myrun_1 | FCFS | 215.05 | 169.60 | 257.05 | 31.45 | 176.41 | 138.82 | 202.24 | .66 |
| myrun_2 | SJF | 124.87 | 62.00 | 256.90 | 62.17 | 86.23 | 16.69 | 229.82 | 2.37 |

Figura 1. Dos tablas creadas por el simulador con el botón Log All Table Data

La primera tabla contiene información sobre la utilización de la CPU y el throughput para cada prueba experimental (myrun_1 y myrun_2). La columna etiquetada Entries especifica dos valores: el número de veces que los procesos entraron a la CPU y el número de veces que pidieron una I/O. La columna Average Time especifica dos valores: el tiempo medio gastado en la CPU y el tiempo medio por ráfaga de I/O. La segunda tabla contiene información más detallada sobre los tiempos de turnaround y de espera.

- Draw Gantt Chart. Permite visualizar un diagrama de Gantt sobre el experimento ejecutado. Antes de eso, el usuario puede elegir sobre cuál de los algoritmos ejecutados quiere el diagrama (ver Figura 2).

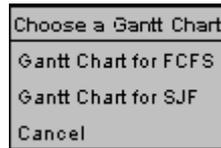


Figura 2. Selección del tipo de diagrama de Gantt

Una vez seleccionado se visualiza en una ventana el citado diagrama (Figura 3).

En el eje de abscisas se representa el tiempo mientras que en el de ordenadas se utiliza una fila para representar a cada uno de los procesos que han tomado parte en la simulación. Justo debajo del gráfico existen una serie de botones de los cuales nos interesan los siguientes:

- Hide. Cierra la ventana.
- Log. Permite insertar una copia del diagrama en el fichero log.

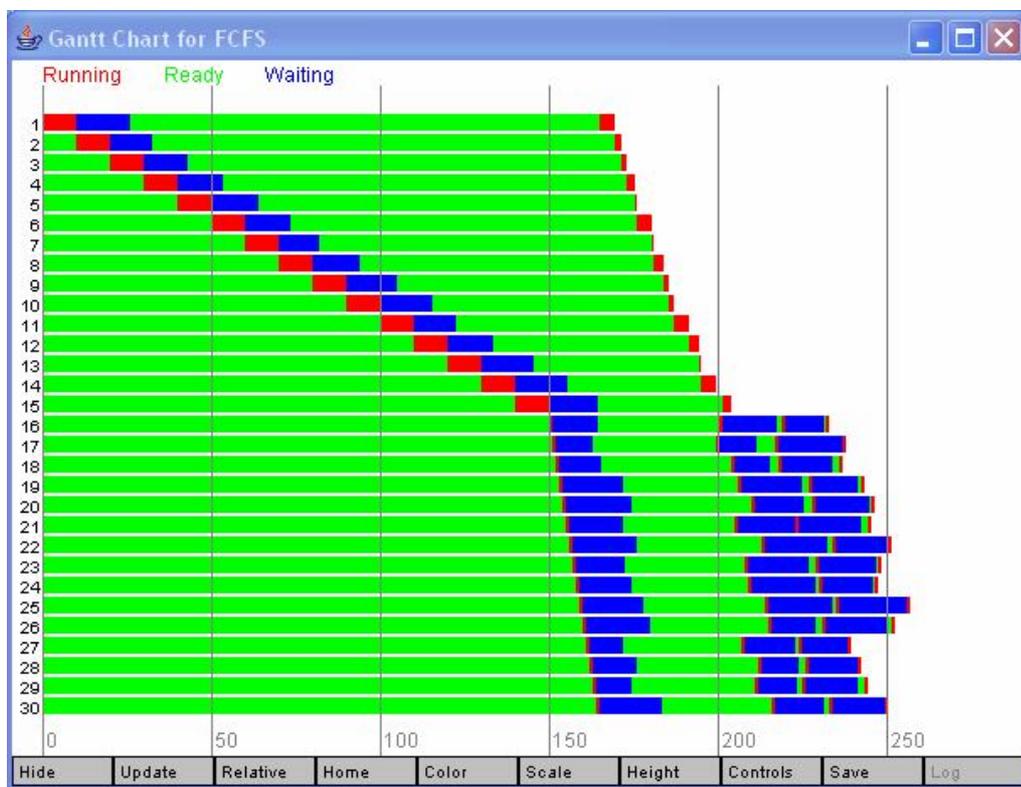


Figura 3. Un ejemplo de diagrama de Gantt