

Programación Concurrente : Docencia Práctica

José Luis Herrero, Fabiola Lucio, David Domínguez, Fernando Sánchez

{Jherrero, FLucio,Fernando}@unex.es

Departamento de Informática

Universidad de Extremadura

Resumen

El objetivo principal de la asignatura es el de establecer los fundamentos de la programación concurrente. Para ello se estudian los problemas inherentes a este paradigma de programación.

El principal problema que tenemos actualmente es que las herramientas existentes para el desarrollo de las prácticas no son las más idóneas debido fundamentalmente a dos motivos: a) son difíciles de utilizar, ya que a menudo están formadas por programas diferentes que hay que manejar de forma separada y b) no ofrecen la posibilidad de realizar un seguimiento de la ejecución del programa concurrente. Debido a esto, los alumnos tardan en comprender los conceptos fundamentales de la asignatura. Con el propósito de solucionar estos problemas, nos hemos planteado la creación de una nueva herramienta integrada que permita la creación, ejecución y depuración de programas concurrentes. Además esta herramienta debe permitir un seguimiento exhaustivo del programa, y mostrar toda la información relativa a los elementos que intervienen en el mismo.

El presente artículo propone esta herramienta como solución a muchos de los problemas de la asignatura de programación concurrente.

1. Introducción

La asignatura de programación concurrente se imparte en el tercer curso de ingeniería informática. Siendo una asignatura troncal en las carreras de ingeniería informática e ingeniería técnica informática. El número de créditos asignados es seis, repartidos en 3 créditos teóricos y 3 prácticos. El objetivo principal de la asignatura es el de establecer los fundamentos de la programación concurrente. Para ello se estudian

los problemas inherentes a este paradigma de programación.

Uno de los problemas fundamentales con los que nos encontramos a la hora de impartir esta asignatura es la dificultad de encontrar herramientas software que permitan comprender el comportamiento de los programas concurrentes. Aunque existen diferentes herramientas que ofrecen la posibilidad de crear programas concurrentes, el problema es que no permiten realizar un seguimiento exhaustivo de la ejecución del mismo. Es por este motivo por el que el alumno tarda en comprender el concepto de concurrencia y los mecanismos de sincronización.

Con el propósito de solucionar este problema, hemos desarrollado una nueva herramienta que permite crear programas concurrentes de una forma sencilla, y que ofrece la posibilidad de mostrar el comportamiento de un programa concurrente.

El presente artículo aborda los problemas que han aparecido en la docencia de la asignatura, así como la propuesta de solución que ofrecemos. Las secciones en que se divide el artículo son las siguientes: La situación actual de la asignatura se explica en la sección segunda. Los problemas que nos ha impulsado a realizar este trabajo están definidos en la tercera sección. La solución que proponemos aparece en la sección cuarta. Por último, las conclusiones y trabajos futuros se muestran en la sección quinta.

2. Situación Actual

En las asignaturas que los alumnos han realizado en cursos anteriores han obtenido la base para desarrollar programas (Elementos de Programación), comprender su comportamiento y utilizar complejas estructuras de datos (Estructuras de Datos), y corregir errores de

programación utilizando la depuración de programas (Laboratorio I y II). Pero todos estos conocimientos han sido obtenidos a partir de lenguajes secuenciales, más concretamente Pascal y C. Debido a esto, aparecen dos problemas fundamentales en la asignatura:

Los alumnos tienen asumido que los programas se basan en tareas secuenciales. Por tanto, resulta muy difícil hacerles comprender la naturaleza concurrente de muchos problemas. Es necesario que el alumno cambie su mentalidad para que puedan entender la concurrencia entre procesos y sean capaces de utilizarla en la resolución de problemas. Las carencias que se han detectado en la enseñanza de la asignatura son las siguientes:

1. Las herramientas existentes para la enseñanza de la programación concurrente son complicadas e incompletas.
2. Las herramientas existentes no incorporan todas las primitivas de sincronización que se estudian, impidiendo así realizar algunas prácticas.

Es en esta asignatura donde los alumnos estudian el concepto de proceso, la ejecución en paralelo de tareas y la sincronización. Es importante que el alumno comprenda los problemas existentes en la programación concurrente tales como la exclusión mutua, los interbloqueos o la inanición, problemas que no aparecían en la programación secuencial. Para solucionar estos problemas han aparecido mecanismos de sincronización entre procesos que permiten establecer zonas de exclusión mutua sobre los recursos compartidos. Entre estos mecanismos de sincronización están los semáforos, regiones críticas, monitores, canales, etc.

La parte teórica de la asignatura se ocupa de explicar cada uno de estos conceptos y que además debe estar apoyada por clases prácticas que ayuden al alumno a comprenderlos mejor. Es en este punto donde se hace necesario la utilización de un conjunto de herramientas que permitan visualizar el comportamiento de los programas concurrentes. Actualmente existe una gran variedad de herramientas que permiten la creación de programas concurrentes en diferentes

lenguajes de programación, tales como ADA, Pascal o incluso C. El problema fundamental de todas estas herramientas es que, o bien es muy difícil realizar un seguimiento del programa, o bien es simplemente imposible.

Hasta la fecha, las prácticas se han desarrollado utilizando una herramienta de programación denominada Pascal-FC creada por Alan Burns y Geoff Davis [Ala93] de la Universidad de Bradford. Esta herramienta, que ha sido diseñada específicamente para la docencia, permite la creación de programas concurrentes usando para ello una extensión del lenguaje Pascal. Los orígenes de esta herramienta se remontan al Pascal-S Desarrollado por Wirth y posteriormente al Pascal Concurrente desarrollado por Ben Ari [Ben90].

La herramienta Pascal-FC permite la creación de programas concurrentes extendiendo el lenguaje Pascal con varias primitivas de concurrencia y sincronización. Es posible crear y lanzar procesos, así como sincronizarlos utilizando diferentes mecanismos como semáforos, monitores, canales, etc.

En la figura 1 se muestran los distintos elementos que intervienen en la creación de un programa concurrente usando Pascal-FC.

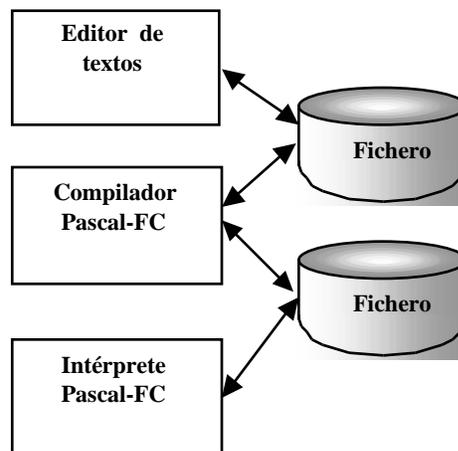


Figura 1. Entorno de programación del Pascal-FC

Como puede comprobarse, la herramienta está formada por tres componentes básicos:

- Un editor de texto, que permite la creación y manipulación de programas concurrentes.
- Un compilador, que realiza un análisis léxico y sintáctico del programa.
- Un intérprete, que ejecuta el programa intercalando cada uno de los procesos.

3. Problemática

Aunque la herramienta Pascal_FC está ampliamente difundida y se utiliza en muchas facultades para la enseñanza práctica de la programación concurrente, presenta varios inconvenientes, entre los que podemos destacar:

La creación, compilación y ejecución se realizan de forma separada. Haciendo muy complicado la creación de programas concurrentes.

La información que proporciona sobre la ejecución del programa es insuficiente, limitándose a mostrar el estado de los procesos al finalizar el programa.

Es posible realizar un seguimiento del programa y comprobar los valores que van tomando las variables a lo largo de la ejecución. Es por tanto difícil de encontrar errores de programación.

Todos estos problemas hacen que a los alumnos les resulte muy complicado desarrollar programas concurrentes y entender el resultado final del mismo. Más aún, se ven incapaces de corregir los errores que puedan aparecer durante la ejecución del programa.

4. Solución propuesta

Debido a los problemas enumerados en la sección anterior, nos hemos planteado desarrollar una nueva herramienta que permita comprender mejor el funcionamiento de los programas concurrentes. Con este objetivo hemos desarrollado una herramienta integrada que permite no sólo desarrollar programas concurrentes, si no lo que es más importante, realizar un seguimiento

exhaustivo de la ejecución del programa, mostrando la suficiente información para que el alumno sea capaz de entender qué es lo que está sucediendo en cada momento.

Esta herramienta ha sido desarrollada por alumnos de la propia Universidad de Extremadura a través de proyectos fin de carrera. Los motivos de habernos decidido por el desarrollo de esta nueva herramienta son fundamentalmente tres:

1. Definir una extensión de Pascal, lenguaje conocido por nuestros alumnos ya que ha sido ampliamente utilizado en prácticas de cursos anteriores. Hay que tener en cuenta que el objetivo de la asignatura no es el de explicar un nuevo lenguaje de programación.
2. Introducir todas las primitivas de creación de procesos y sincronización que se estudian a lo largo de la asignatura.
3. Dotar al alumno de mecanismos suficientes para realizar una el seguimiento a cualquier programa concurrente y así comprender mejor su funcionamiento.

Para que en esta nueva herramienta no estuvieran presentes los mismos problemas que se han mencionado anteriormente, hemos exigido que el producto final cumpla las siguientes requisitos:

- **Integrada** : Es una herramienta integrada que se compone de un editor, compilador, intérprete y depurador. Así el alumno puede realizar todas las operaciones desde la misma herramienta.
- **Completa** : Muestra la información del estado de cada proceso y el valor de las variables en cualquier momento de la ejecución del programa. Además es posible conocer el estado de cada una de las primitivas de sincronización ya sean semáforos, monitores, regiones críticas, canales, etc.
- **Depurable** : El depurador permite realizar un seguimiento exhaustivo de la ejecución del programa. La ejecución paso a paso y los

puntos de ruptura han sido incluidos para facilitar esta tarea.

- **Amigable** : Para que resulte más atractivo para el alumno, hemos cambiado el interface de usuario para que se pueda visualizar en formato gráfico de ventanas, ya que la mayoría de las herramientas sólo permitían un interface básico de comandos.

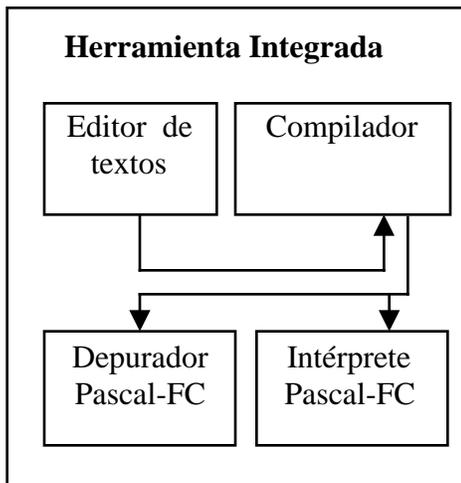


Figura 2. Entorno de programación integrado del Pascal-FC

El proceso de depuración es el más innovador, ya que permite visualizar en todo momento el estado de cada uno de los procesos e información referente a las variables del programa. Para ello, los programas originales son traducidos a código intermedio P_Code que permite definir un lenguaje basado en una máquina de pila. Es este código intermedio el que se interpretará. Gracias a este lenguaje intermedio es posible ejecutar paso a paso cada una de las instrucciones de un programa y consultar el estado de los procesos y los valores de las variables.

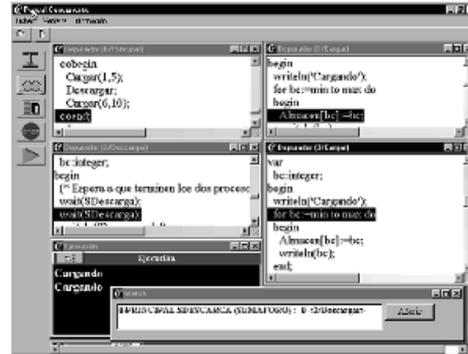


Figura 3. Entorno de la herramienta desarrollada

La figura 3 muestra el funcionamiento de la herramienta. Como puede observarse, cada uno de los procesos activos están contenidos en una ventana independiente, quedando resaltada la instrucción que se está ejecutando en cada momento. Además la ventana de ejecución muestra la salida en pantalla del programa. Por último, la ventana de variables permite seleccionar tanto las variables del programa, como las primitivas de sincronización para comprobar su valor actual.

La evaluación de esta herramienta no ha sido efectuada todavía ya que ha sido terminada recientemente. Esta herramienta se implantará en el segundo cuatrimestre del presente curso para realizar las prácticas de la asignatura.

5. Conclusiones y trabajos futuros

Las herramientas existentes en la actualidad para la práctica de la programación concurrente son complicadas y en muchos casos incompletas. Debido a esto los alumnos tardan en comprender los conceptos básicos de la asignatura. Para solucionar este problema, nos hemos planteado desarrollar una nueva herramienta integrada que soporte todas las necesidades de la asignatura. Entre las características que debe cumplir la herramienta están las de ser integrada, completa, que permita la depuración y ofrezca un entorno amigable. Aunque ha sido terminada recientemente, esperamos ofrecer los resultados en poco tiempo, ya que será implantada en el segundo cuatrimestre del presente curso académico

Como trabajo futuro nos estamos planteando la posibilidad de incluir la comunicación entre procesos remotos. De este modo, procesos que se encuentran en diferentes máquinas podrían estar sincronizados de la misma manera que los procesos locales. La nueva versión incluirá el estándar CORBA como mecanismo de comunicación entre procesos.

REFERENCIAS

[Ben90] Ben Ari. *Principles of concurrent and distributed programming*. Ed. Prentice Hall, 1990.

[Ala93] Alan Burns & Geoff Davies. *Concurrent Programming*. Ed. Addison-Wesley. 1993.