

PRÁCTICA 1 - PSP - CARRERA DE RELEVOS.

En este documento voy a escribir algunas de las conclusiones después de concluir la práctica 1 - Programación concurrente.

En primer lugar, HA SIDO CASI IMPOSIBLE HACER LAS DISTINTAS PISTAS CONCURRENTES.

Voy a intentar describir los problemas con los que me he encontrado:

1. El primer borrador del ejercicio seguía todas las normas de clases y métodos que estipulaba el profesor. Pero después de los primeros 30 errores de concurrencia, resolví arreglar de alguna manera el funcionamiento del programa para no volverme loco.
2. Lo más llamativo, SIN DUDA, es una tontería que choca con lo que estudiamos en primero. RUNNER NO TIENE POSICION! Si haces un programa, de carreras, y haces un Runner, lo más lógico es que guarde un número entero con los metros que ha recorrido. Para superar este problema, solo podía girar un poco el enunciado, y darle a la 'Pista' un atributo, de tipo int, que guardara una distancia recorrida virtualmente por el runner. En términos de concurrencia, esto simplifica las cosas, porque solo me preocupo de sumar 1 a la posición, realizando eventos en los runner cuando suceden ciertos 'turnPoints', que coinciden con los cambios de testigo.
3. A partir de aquí, todo parecía coser y cantar, pero han sucedido los problemas de concurrencia, porque desde el Main, con una lambda había que hacer concurrentes cuatro pistas, que con un executor había que hacer concurrentes 4 runners. Mientras escribo esto pienso que quizás no era necesario hacer concurrentes 4 runners, sino simplemente hacerlos secuenciales. Supongo que he tomado esta decisión para que fuera el objeto monitor de 'testigo' el que hiciera esperar o notificar a los runners. Tenía más gracia supongo. De cualquier modo, aunque el resultado final ya es bastante bueno, no he conseguido que las pistas sean 100% concurrentes, y esto se nota en las líneas que salen por consola, donde algunos cambios de testigo se pisan unos con otros.
4. Estoy a punto de tirar la toalla para depurar un programa de concurrencia. Supongo que será cuestión de práctica.

Además de estas conclusiones, quería hacer llegar una sugerencia sobre el formato de la asignatura. La última clase donde se debatía sobre el funcionamiento de este año, me dejó con algunas sugerencias que escribo en este documento PDF, simplemente a modo de feedback.

He pensado que quizás, sería posible reorganizar las clases teóricas y las clases prácticas de la siguiente forma:

LUNES: 1 HORA: Clase teórica.

No se hace pesado, es un buen momento para explicar teoría, y es una manera de ocupar toda la clase, dejar tiempo para dudas, y contar alguna anécdota.

VIERNES 1 HORA: Clase teórica.

Para terminar de reforzar conceptos, aclarar dudas y tratar temas generales de la asignatura.

DESCANSO: 25 MINUTOS

VIERNES 2 HORA: Examen.

Un examen, de 30-45 minutos. Cada viernes de cada semana toca un examen tipo distinto.

SEMANA 1: Tipo Test. 15-20 preguntas a,b,c

SEMANA 2: Tipo JAVA. 30-45 minutos examen práctico

SEMANA 3: Cualquier otra cosa de examen. Para practicar en la clase

SEMANA 4: Descanso

Estos exámenes son el 15-20% de la nota final y, aquí viene lo importante, sirven como PREPARACIÓN PARA EL EXAMEN FINAL, que no fue fácil, no teníamos práctica y no calculamos bien el tiempo.

Son cosas que se me ocurren, desde la parte que me toca de alumno, de cualquier manera, pienso al igual que mis compañeros, que la asignatura de PSP está menospreciada, dada su importancia y la escasa carga lectiva que tenemos en el curso.

Por lo demás, me parece muy profesional que el profesor nos pregunte nuestra opinión sobre la asignatura, y realmente pienso que el enfoque es muy acertado (la forma de dar teoría y los repositorios con ejemplos) y además se agradece que los contenidos no sean aburridos y que el ritmo de la clase sea ágil.