

Sistemas Operativos I - Parcial - 20/10/2015

- Nombre:
- DNI:
- Firma (al entregar y ante un docente):

Nota: Los alumnos que obtuvieron una calificación de *R (Regular)* o superior en el parcialito de scripting NO deben resolver el ejercicio 2).

1. Explique en detalle una ventaja y una desventaja del mecanismo de paginación respecto de los esquemas de asignación de memoria contigua.
2. Se tiene un archivo con información de ciudades del mundo, con una estructura tabular donde cada fila posee una columna con el nombre de la ciudad, otra con el país, y columnas que guardan la temperatura media histórica de cada mes (en grados Celsius). Escriba un script que dado como parámetro un país retorne el nombre y las temperaturas medias de sus ciudades expresadas en grados Fahrenheit. Considere que $F = 1,8 * C + 32$.
3. Se desea implementar una pequeña librería que ejecute trabajos en forma paralela. Un usuario de esta librería, desde su aplicación, debería ser capaz de enviar a ejecutar asincrónicamente un trabajo y seguir realizando otras tareas. Es decir, enviar un trabajo es una operación no bloqueante. Adicionalmente, se debe poder limitar la cantidad máxima de trabajos ejecutando en paralelo: si llegan más trabajos que hilos de ejecución los nuevos trabajos deben encolarse hasta que alguno de los hilos termine su trabajo actual. Características importantes:
 - * Los hilos que procesan trabajos ejecutan un trabajo y luego quedan a la espera de otro trabajo. Terminar un trabajo no implica la destrucción del hilo.
 - * Se puede asumir que todos los trabajos implementan la interfaz *ITrabajo* con el método *public Object ejecutar()*.
 - * Cada trabajo tiene asociada un objeto *Respuesta*. Esta respuesta permite recuperar el resultado de la ejecución, esperando en caso de que la misma no esté aún disponible.

Ejemplo de uso:

```
Paralelizador p = new Paralelizador(4);
                // Ejecuta hasta 4 trabajos a la vez.
Respuesta r = p.enviar(new TrabajoX());
                // Envío a ejecutar el trabajo x.
                // Llamada no bloqueante.
sentencial // Sigo ejecutando normalmente
...
sentenciaN
Object data = r.getRespuesta(); // Dos opciones:
                // El trabajo terminó -> el llamado a "getRespuesta" retorna
                // inmediatamente el resultado del trabajo;
                // o no se terminó de ejecutar: en cuyo caso se espera hasta
                // que el resultado esté disponible.
```