15005397. CSIFC02. MP0490. Programación de servizos e procesos. 2022-2023

Tarea para PSP01.

Tarea para PSP01.

Detalles de la tarea de esta unidad.

Enunciado.

Ejercicio 1) Se compone de 3 partes:

- Primera parte: implementa una aplicación que ordena un conjunto indeterminado de números que recibe a través de su entrada estándar; y muestra el resultado de la ordenación en su salida estándar. La aplicación se llamará 'ordenarNumeros'.
- Segunda parte: implementa una aplicación, llamada 'aleatorios', que genere al menos 40 números aleatorios (entre 0 y 100), y que los escriba en su salida estándar.
- Tercera parte: Realiza un pequeño manual (tipo "¿Cómo se hace?" o "HowTo"), utilizando un editor de textos (tipo word o writer) en el que indiques, con pequeñas explicaciones y capturas, cómo has probado la ejecución de las aplicaciones que has implementado en este ejercicio. Entre las pruebas que hayas realizado, debes incluir una prueba en la que utilizando el operador "|" (tubería) redirijas la salida de la aplicación 'aleatorios' a la entrada de la aplicación 'ordenarNumeros'.

Ejercicio 2) Se compone de 3 partes:

- Primera parte: implementa una aplicación que escriba en un fichero indicado por el usuario conjuntos de letras generadas de forma aleatoria (sin sentido real). Escribiendo cada conjunto de letras en una línea distinta. El número de conjuntos de letras a generar por el proceso, también será dado por el usuario en el momento de su ejecución. Esta aplicación se llamará "lenguaje" y como ejemplo, podrá ser invocada así:

  ```
  java -jar lenguaje 40 miFicheroDeLenguaje.txt
  ```

  Indicando que se generarán 40 palabras del lenguaje y serán guardadas en `miFicheroDeLenguaje.txt`

- Segunda parte: implementa una aplicación, llamada 'colaborar', que lance al menos 10 instancias de la aplicación "lenguaje". Haciendo que todas ellas, colaboren en generar un gran fichero de palabras. Cada instancia generará un número creciente de palabras de 10, 20, 30, ... Por supuesto, cada proceso seguirá escribiendo su palabra en una línea independiente de las otras. Es decir, si lanzamos 10 instancias de "lenguaje", al final, debemos tener en el fichero 10 + (Sumar.) 20 + 30 + ... + 100 = (Igual.) 550 líneas.
- Tercera parte: Realiza un pequeño manual (tipo "¿Cómo se hace?" o "HowTo"), utilizando un editor de textos (tipo word o writer) en el que indiques, con pequeñas explicaciones y capturas, cómo has probado la ejecución de las aplicaciones que has implementado en este ejercicio.

Criterios de puntuación. Total 10 puntos.

Actividad 1) 4 puntos, de los cuales:

- Aplicación "ordenarNumeros" aceptando un número indeterminado de valores → 1,5p (Puntos.).
- Aplicación "aleatorios"→ 0,75p.
- Documentación del código y Javadoc en ambas aplicaciones → 0,75p.
- Mini manual de uso y pruebas → 1p.

Actividad 2) 6 puntos, de los cuales:

- Aplicación "lenguaje" (correcta para su ejecución en modo aislado o secuencial) → 1,5p.
- Aplicación "colaborar" (lanzando aplicaciones simultáneas) → 1p.
- Aplicación "lenguaje" (correcta para su ejecución en un entorno concurrente de ejecución compartiendo un recurso) → 1,75p
- Mini manual de uso y pruebas → 1p.
- Documentación del código y Javadoc en ambas aplicaciones → 0,75p.

Se tendrá en cuenta que:

- La ejecución de los programas produce el resultado esperado.
- No se produce interbloqueo ni inanición.

Recursos necesarios para realizar la Tarea.

- IDE NetBeans.
- Contenidos de la unidad.
- Ejemplos expuestos en el contenido de la unidad.

Específico para el Ejercicio1: Puedes tomar como base el ejemplo2: comunicación utilizando tuberías, visto en el punto 5.1 Mecanismos básicos de comunicación de esta unidad.

Específico para el Ejercicio2: Puedes tomar como base el ejemplo visto en el apartado 6.1 Regiones críticas de esta unidad.

Si utilizas objetos de tipo `RandomAccessFile` para el acceso a ficheros, recuerda que puede ser necesario que te posiciones al final del fichero para ir añadiendo valores en el fichero uno detrás de otro.

Consejos y recomendaciones.

- Recuerda que para que un proceso tome como entrada los datos generados por otro, debes lanzar su ejecución desde el intérprete de comandos utilizando el operador tubería "|".
- Puedes programarte una aplicación que cuente el número de líneas de un fichero, para comprobar el correcto resultado del segundo ejercicio. En GNU/Linux, cuentas con el comando "wc -l nombreArchivo" que te devolverá el número de líneas que contiene nombreArchivo.

Indicaciones de entrega.

Los documentos deben tener tamaño de página A4, estilo de letra Times New Roman, tamaño 12 e interlineado normal.

Debes crear una carpeta para el ejercicio, y en ella una carpeta para cada actividad. En cada carpeta incluye los proyectos y documentos correspondientes. Comprímelo todo en un fichero.

Por ejemplo, tendremos: PSP01_Tarea con los directorios, Actividad1 y Actividad2; en Actividad1 tendremos dos proyectos y un documento; en Actividad2, dos proyectos y un documento.

Una vez realizada la tarea elaborarás un único documento donde figuren las respuestas correspondientes. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre_SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la primera unidad del MP de PSP**, debería nombrar esta tarea como...

sanchez_manas_begona_PSP01_Tarea

Estado da entrega

Estado da entrega	Sen intentos
Estado das cualificacións	Sen cualificar
Última modificación	-
Comentarios a entrega	▸ Comentarios (0)

Engadir entrega

Vostede aínda non fixo ningunha entrega

◀ Recursos complementarios UD01.

Ir a...

Inicio Preguntas Frecuentes Accesibilidade Aviso Privacidade Contacto