

Comunicaciones – Servicios – Seguridad

Aclaración: En este examen se evalúan contenidos específicos. Cada ejercicio **debe** ser hecho con las técnicas requeridas, en caso contrario el ejercicio será puntuado con **0 puntos**.

Ejemplo: Escribe un programa que cree un proceso hijo con fork, el hijo escribirá “Hola mundo” y el padre esperará a que finalice el proceso hijo.

Entrega: El alumno entrega un programa que escribe “Hola mundo” (Similar al que se hace el primer día de clase de programación).

Resultado: **0 puntos**.

Penalizaciones:

Cada línea de código con mala sangría o sangría inconsistente restará **0.2 punto**.

Cada número mágico en el código restará **0.2 puntos**. Los valores constante deben ser CONSTANTES (escritas con nombre mayúsculas)

Ejercicios

1. UDP-UDP generador de cuadrados 3 puntos
2. Cifrador y correo 3 puntos
3. Generador de primos HTTP 4 puntos

Ejercicio 1

Crea un programa que al ejecutarse reciba dos números enteros como parámetros. El primer entero es el puerto de escucha, el segundo es el puerto de envío.

En cada Datagrama UDP espera encontrar otros dos números y un carácter separados por espacios, el primer número representa la altura, el segundo el ancho y el carácter es el carácter utilizado a continuación.

Al recibir cada Datagrama el programa generará un cuadrado con el alto y el ancho recibidos y lo enviará en el segundo puerto a través de Broadcast UDP.

El programa estará leyendo hasta que se le envíe la cadena 'fin', 'Fin', o 'FIN', 'fiN', etc o cualquier combinación de mayúsculas y minúsculas que sean igual a 'fin'

Envío:

3 5 a

aaaaa

a a

aaaaa

Cuando lo tengas ejecuta dos comando nc para verificar que funciona. Haz capturas de pantalla de los comandos.

3 puntos puntos:

0.75 Escucha y recepción de datagramas

0.75 Generación de cuadrado

0.5 Envío de cuadrado

0.5 Finalización con fin y cierre de comunicaciones.

0.5 Captura de comandos nc funcionando (Se tiene que ver el comando)

NOTA:

No se producen errores en la entrada, el programa siempre manda números mayores que 1 y todos los datos están bien.

Ejercicio 2

Crea un programa que reciba como parámetro dos rutas, un número entero y dos cadenas de texto

La primera ruta es la ruta de un fichero con mensajes

La segunda ruta es la ruta de un fichero con direcciones de correo

El número entero representa el punto de partida para el cifrado Cesar incremental. (Similar al de los ejercicios propuestos). Tras cifrar el mensaje el cifrador se reinicia a este valor.

Las dos últimas cadenas representan el usuario de envío smtp y la contraseña.

Ejemplo de ejecución

```
java examen.E2 data/mensajes.txt data/dirs.txt 13 jorge.duenas@educa.org abc1234
```

El programa enviará el mensaje de la línea 1 del fichero mensajes a la dirección 1 del fichero dirs, usando el cifrado Cesar incremental con inicialización de 13, desde la cuenta jorge.duenas@educa.org y password abc1234.

Luego lo mismo con la línea dos, línea tres, etc.

3 puntos puntos:

0.5 Lectura de parámetros

0.5 Lectura de información de ficheros

0.5 Relacionar cada línea con la línea del otro fichero.

1.0 Cifrado del mensaje

0.5 Envío de correo

NOTA: Utiliza funciones auxiliares para hacer cada subtrabajo.

NOTA: Los ficheros tienen el mismo número de líneas. Mételes en un ArrayList y recórrelos.

Ejercicio 3

Crea un programa que reciba como parámetro un número de puerto por el que escuchar.

El programa será un servidor multithread web.

Cuando el programa reciba una petición en la URL habrá dos parámetros.

Ejemplo visito en el navegador

```
http://localhost:8000/30/6/
```

Petición:

```
GET /30/6/ HTTP/1.1
```

```
Host: localhost:8000
```

El programa extraerá de la petición get los dos números, el primero será n y el segundo m.

El primer número representa el entero inicial, el segundo número representa la cantidad de primos.

El programa devolverá una página web con ul-li con los m (6 en el ejemplo) siguientes primos a n (30 en el ejemplo).

Ejemplo:

```
<ul>
<li>31</li>
<li>37</li>
<li>41</li>
<li>43</li>
<li>47</li>
<li>53</li>
</ul>
```

Puntos

1 servidor multithread arranca en puerto pasado como parámetro

1 recibir petición GET y procesar los parámetros

1 generar cadena representado la web con los primero

1 construir petición HTTP de respuesta con el HTML

NOTA: Ejemplo de respuesta HTTP/1.1 200 OK