

RA1 y se evalúa todos los CE de ese resultado de aprendizaje.

Cómo se entrega: Deberás seguir estas instrucciones para la entrega o no se corregirá el ejercicio.

- Descomprime la carpeta ‘ejercicio.zip’.
- Mete el código de cada ejercicio en su fichero correspondiente: ejer1.py, ejer2.py, etc....
- Antes de enviar la carpeta comprimida deberás ejecutar sobre cada archivo .py el siguiente comando: md5sum ejerX.py > chkX
- En la carpeta ‘ejercicio’ solo debe estar el pdf que estás leyendo, los archivos .py de los ejercicios que hayas realizado y sus respectivos chk, uno por cada archivo.
- Comprime la carpeta y súbela a la plataforma.

1.- (2.5 ptos) Crea un programa en python, ejer1.py, que:

- Pida al usuario un número N (cantidad de procesos hijos a crear).
- Crea ese número N de procesos hijos usando **os.fork()**
- Cada hijo debe imprimir:
 - Su número de proceso (PID).
 - El número de orden en el que se crea ($1 \leq \text{numero} \leq N$).
 - Una pausa aleatoria entre impresiones (para simular trabajo), entre 1 y 2 minutos.

```
introduce el número de hijos: 3
Hijo 1 creado con el número de proceso 28066.
Hijo 2 creado con el número de proceso 28067.
Hijo 3 creado con el número de proceso 28068.
miguel@miguel-GV62-8RD:~/Insync/instituto/Programación procesos y servicios/Marcombo/código$ Proceso 28066 finalizado
Proceso 28068 finalizado
Proceso 28067 finalizado
```

2.- (2.5 ptos) Crea un programa en python, ejer2.py, que haga lo siguiente:

- Pregunte por un directorio. Si se quiere listar el directorio actual no se introduce nada.
- Nos diga el número de archivos o directorios que son ocultos.
- El programa debe reportar cuando el directorio no es correcto por fallo del usuario.
- Para realizar este ejercicio debes usar **subprocess.Popen**

Ejemplo:

```
miguel@miguel-GV62-8RD:~/Insync/instituto/Programación procesos y servicios/Marco
puntuado/ejercicio2.py"
Introduce el directorio(intro para actual): /etc/miguel
ls: no se puede acceder a '/etc/miguel': No existe el archivo o el directorio

miguel@miguel-GV62-8RD:~/Insync/instituto/Programación procesos y servicios/Marco
puntuado/ejercicio2.py"
Introduce el directorio(intro para actual): /home/miguel
El número de fichero ocultos en ese directorio es: 42
miguel@miguel-GV62-8RD:~/Insync/instituto/Programación procesos y servicios/Marco
```

- No se debe contar ni '.' ni '..' que son el directorio actual y el padre.

3.- (2.5 ptos) Escribe un programa en Python, ejer3.py, que:

- Pida al usuario un número entero positivo N (por ejemplo, N = 100).
- Se crean 2 procesos hijo.
- El primer hijo buscará y contará los números primos en el rango [2, N//2].
- El segundo hijo buscará y contará los números primos en el rango [N//2 + 1, N].

- Cada hijo imprimirá cuántos primos encontró en su rango y la lista de los mismos.

Nota: X // Y en python es una división que devuelve un entero.

```
Introduce rango: 20
En el rango desde 2 hasta 10 el número de primos es 4
Lista [2, 3, 5, 7]
En el rango desde 11 hasta 20 el número de primos es 4
Lista [11, 13, 17, 19]
miguel@miguel-GV62-8RD:~/Insync/instituto/Programación pr
```

4.- (2.5 ptos) Escribe un programa en Python, **ejer4.py**, que:

- Usa el comando de linux **md5sum** que saca el hash de un archivo. El hash es un número único que si el archivo no cambia es siempre igual. Cualquier cambio provocará un cambio en el número.
- Pide una cadena de texto, le realiza el md5sum, usando **subprocess.Popen** y lo muestra por pantalla.
- El programa sigue pidiendo una nueva frase y realizando lo dicho en el segundo punto hasta que se para con la cadena ‘salir’.

```
Escribe la frase('salir' para acabar): hola
== SALIDA ==
4d186321c1a7f0f354b297e8914ab240 - 

Escribe la frase(salir para acabar): hola
== SALIDA ==
4d186321c1a7f0f354b297e8914ab240 - 

Escribe la frase(salir para acabar): holas
== SALIDA ==
149f00036617f2c10c9457af10540bf1 - 

Escribe la frase(salir para acabar): salir
```