Universidad Rey Juan Carlos

LABORATORIO DE SISTEMAS (GIRS)

EXAMEN PRÁCTICO JUNIO - ANEXO: INFORMACIÓN DE UTILIDAD

21 DE JUNIO DE 2023

1 Códigos de salida en GNU/Linux

Código de salida	Descripción
0	Éxito
1	Operación no permitida
2	No existe el archivo o directorio
3	El proceso no existe
5	Error de entrada/salida
7	Lista de argumentos demasiado larga
13	Permiso denegado
17	El archivo ya existe
22	Argumento inválido
95	Operación no soportada

Tabla 1: Listado de algunos códigos de salida estándar para programas en GNU/Linux.

2 El módulo Python `argparse`

2.1 Creación de un nuevo *parser*

```
import argparse
...
parser = argparse.ArgumentParser(
                    prog='ProgramName',
                    description='What the program does',
                    epilog='Text at the bottom of help')
```

2.2 Añadir nuevos argumentos con `add_argument(...)`

```
parser.add_argument('filename')           # positional argument
parser.add_argument('-c', '--count')      # option that takes a value
parser.add_argument('-v', '--verbose',
                    action='store_true') # on/off flag
```

2.3 Programa de ejemplo

Ejemplo de un programa `prog.py`.

```
import argparse

parser = argparse.ArgumentParser(description='Process some integers.')
parser.add_argument('integers', metavar='N', type=int, nargs='+',
help='an integer for the accumulator')
parser.add_argument('--sum', dest='accumulate', action='store_const',
const=sum, default=max,
help='sum the integers (default: find the max)')

args = parser.parse_args()
print(args.accumulate(args.integers))
```

Que produce la siguiente información de ayuda:

```
$ python prog.py -h
usage: prog.py [-h] [--sum] N [N ...]

Process some integers.

positional arguments:
N           an integer for the accumulator

options:
-h, --help  show this help message and exit
--sum       sum the integers (default: find the max)
```

2.4 Posibles argumentos para `add_argument(...)`

A continuación, una lista resumen de argumentos aceptados por la función `ArgumentParser.add_argument()`.

Nombre	Descripción	Valores
`action`	Indica cómo manejar el argumento	`'store'`, `'store_const'`, `'store_true'`, `'append'`, `'append_const'`, `'count'`, `'help'`, `'version'`
`choices`	Limita los posibles valores recibidos a un conjunto predefinido	`['foo','bar']`, `range(1,10)`
`const`	Almacena un valor constante	
`default`	Valor por defecto asignado cuando no se introduce ese argumento	Por defecto es **`None`**
`dest`	Especifica el nombre de atributo empleado en el *namespace* resultante	
`help`	Mensaje de ayuda para un argumento	Cadena de caracteres con el mensaje

`metavar`	Nombre alternativo para el argumento cuando se muestra en el mensaje de ayuda	
`nargs`	Numero de veces que puede aparecer el argumento	`int`, `'?'`, `'*'`, `'+'` o `argparse.REMAINDER`
`required`	Booleano que indica si el argumento es obligatorio u opcional	**`True`** o **`False`**
`type`	Convierte de forma automática el argumento al tipo de dato indicado	Ejemplos: `int`, `float`

Tabla 2: Listado de algunos códigos de salida estándar para programas en GNU/Linux.