

P2

servcli2

```
1  import os, time, sys, datetime
2
3  rd, wd = os.pipe()
4  r, w = os.fdopen(rd, 'rb', 0), os.fdopen(wd, 'wb', 0)
5
6  pid = os.fork()
7
8  if pid: #padre (cliente)
9      w.close()
10
11      while True:
12          fecha_hora = r.readline()
13          if not fecha_hora:
14              break
15          print(fecha_hora.decode('utf8'))
16  else:
17      r.close()
18      fecha_hora_actual = datetime.datetime.now()
19      fecha_hora_actual_str = fecha_hora_actual.strftime("%Y-%m-%d %H:%M:%S")
20      w.write(fecha_hora_actual_str.encode('utf8'))
21      w.flush()
22      w.close()
```

### P3 - serv3 / cli3

```
1  import os, time, sys, datetime
2  fifo_cliente_servidor = '/tmp/cliservFIFOapp1'
3  fifo_servidor_cliente = '/tmp/servcliFIFOapp1'
4
5  if not os.path.exists(fifo_cliente_servidor):
6      os.mkfifo(fifo_cliente_servidor)
7
8  if not os.path.exists(fifo_servidor_cliente):
9      os.mkfifo(fifo_servidor_cliente)
10
11 #while True:
12 cliente_escribe = open(fifo_cliente_servidor, 'w')
13 path = input()
14 cliente_escribe.write(path)
15 cliente_escribe.close()
16
17 cliente_lee = open(fifo_servidor_cliente, 'r')
18
19 contenido = cliente_lee.read()
20 print(contenido)
21
22
23 cliente_lee.close()
```

```
1  import os, time, sys, datetime
2  fifo_cliente_servidor = '/tmp/cliservFIFOapp1'
3  fifo_servidor_cliente = '/tmp/servcliFIFOapp1'
4
5  if not os.path.exists(fifo_cliente_servidor):
6      os.mkfifo(fifo_cliente_servidor)
7
8  if not os.path.exists(fifo_servidor_cliente):
9      os.mkfifo(fifo_servidor_cliente)
10
11
12
13
14 while True:
15     servidor_lee = open(fifo_cliente_servidor, 'r')
16     path_fichero = servidor_lee.readline()
17     print(path_fichero)
18     servidor_lee.close()
19
20     servidor_escribe = open(fifo_servidor_cliente, 'w')
21     try:
22         #Abrimos el archivo y leemos su contenido
23         archivo = open(path_fichero, 'r')
24         contenido = archivo.read()
25         mensaje = contenido.encode('utf8')
26     except Exception as e:
27         # Si hay algún error, lo capturamos y lo enviamos al cliente
28         mensaje = str(e).encode('utf8')
29
30     servidor_escribe.write(contenido)
31     servidor_escribe.close()
32
33
```

## P4 - app1

### serv4

```
1  import socket
2  import os
3
4  s = socket.socket(socket.AF_UNIX, socket.SOCK_STREAM, 0) # creamos un socket
5
6  nombre_socket = input('Introduzca el nombre del socket: ')
7
8  if not nombre_socket:
9      nombre_socket = 'socket4'
10
11  # Eliminamos el archivo de socket si existe
12  socket_file = "/tmp/" + nombre_socket
13  if os.path.exists(socket_file):
14      os.remove(socket_file)
15
16
17  s.bind(socket_file) # asignamos una dirección al socket
18  s.listen(5)
19
20  while True:
21      ns, direccion = s.accept()
22      print('Nueva conexión establecida')
23
24      petición = ns.recv(1024) # hacer un bucle para que pueda leer cualquier tamaño
25      print(petición.decode('utf8'))
26
27      try:
28          #Abrimos el archivo y leemos su contenido
29          with open(petición, 'rb') as f:
30              while True:
31                  datos = f.read(1024)
32                  if not datos:
33                      break
34                  if isinstance(datos, str):
35                      contenido = datos.encode('utf8')
36                  elif isinstance(datos, bytes):
37                      contenido = datos
38                  ns.send(contenido)
39      except Exception as e:
40          contenido = str(e).encode('utf8')
41          ns.send(contenido)
42
43
44  ns.close()
```

### cli4

```
1  import socket
2
3  s = socket.socket(socket.AF_UNIX, socket.SOCK_STREAM, 0) # creamos un socket: socket(dominio, tipo, protocolo)
4
5  nombre_socket = input('Introduzca el nombre del socket: ')
6
7  if not nombre_socket:
8      nombre_socket = 'socket4'
9
10  socket_file = "/tmp/" + nombre_socket
11  s.connect(socket_file) # abrimos la conexión con el servidor
12
13  path = input('Introduzca el nombre del archivo: ')
14
15  s.send(path.encode('utf8'))
16
17  full_msg = ''
18  while True:
19      mensaje = s.recv(1024)
20      if not mensaje:
21          break
22      full_msg += mensaje.decode('latin-1')
23  print(full_msg)
24
25  s.close()
```

## P4 - app2

### serv4

```
1  import socket, datetime, os
2
3  s = socket.socket(socket.AF_UNIX, socket.SOCK_STREAM, 0)
4
5  nombre_socket = input('Introduzca el nombre del socket: ')
6
7  if not nombre_socket:
8      nombre_socket = 'socket4'
9
10 # Eliminamos el archivo de socket si existe
11 socket_file = "/tmp/" + nombre_socket
12 if os.path.exists(socket_file):
13     os.remove(socket_file)
14
15 s.bind(socket_file)
16 s.listen(1)
17
18 while True:
19     ns, dir = s.accept()
20     print('Nueva conexion establecida')
21     peticion = ns.recv(1024)
22     print(peticion.decode())
23     fecha_hora = datetime.datetime.now()
24     fecha_hora_str = fecha_hora.strftime("%Y-%m-%d %H:%M:%S")
25     ns.send(fecha_hora_str.encode('utf8'))
26     ns.close()
```

### cli4

```
1  import socket, os
2
3  s = socket.socket(socket.AF_UNIX, socket.SOCK_STREAM, 0)
4
5  nombre_socket = input('Introduzca el nombre del socket: ')
6
7  if not nombre_socket:
8      nombre_socket = 'socket4'
9
10 socket_file = "/tmp/" + nombre_socket
11 s.connect(socket_file)
12
13 peticion = '¿Qué hora es?'
14 s.send(peticion.encode('utf8'))
15
16 respuesta = s.recv(1024)
17 print(respuesta.decode())
18
19 s.close()
```

## P5 - app1

### serv5

```
1  import socket, os
2
3  socket_serv = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, 0)
4
5  dir_serv = input('Dirección del servidor: ')
6  puerto_serv = input('Puerto del servidor: ')
7
8
9  if not dir_serv:
10     dir_serv = '127.0.0.1'
11
12  if not puerto_serv:
13     puerto_serv = '5555'
14
15  socket_serv.bind((dir_serv, int(puerto_serv)))
16
17  while True:
18     path, dirCliente = socket_serv.recvfrom(1024)
19     print(path.decode())
20     print(dirCliente)
21
22     try:
23         #Abrimos el archivo y leemos su contenido
24         archivo = open(path, 'rb')
25         mensaje = "Empieza"
26         while mensaje:
27             mensaje = archivo.read(1024)
28             if isinstance(mensaje, str):
29                 contenido = mensaje.encode('utf8')
30             elif isinstance(mensaje, bytes):
31                 contenido = mensaje
32             socket_serv.sendto(contenido, dirCliente)
33
34     except Exception as e:
35         contenido = str(e).encode('utf8')
36
37
38
39
```

### cli5

```
1  import socket
2
3  socket_cli = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, 0)
4
5  dir_serv = input('Dirección del servidor: ')
6  puerto_serv = input('Puerto del servidor: ')
7  path = input('Path del fichero: ')
8
9
10 if not dir_serv:
11     dir_serv = '127.0.0.1'
12
13 if not puerto_serv:
14     puerto_serv = '5555'
15
16 socket_cli.sendto(path.encode('utf8'), (dir_serv, int(puerto_serv)))
17
18 full_msg = ''
19 datos = "Empieza"
20 while datos:
21     datos, dir = socket_cli.recvfrom(1024)
22     full_msg += datos.decode('latin-1')
23 print(full_msg)
24
25 socket_cli.close()
26
```

## P5 - app2

### serv5

```
1  import socket, datetime
2
3  socket_serv = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, 0)
4
5  dir_serv = input('Dirección del servidor: ')
6  puerto_serv = input('Puerto del servidor: ')
7
8  if not dir_serv:
9      dir_serv = '127.0.0.1'
10
11  if not puerto_serv:
12      puerto_serv = '5555'
13
14  socket_serv.bind( (dir_serv, int(puerto_serv)) )
15
16  print('El servidor está listo para recibir')
17
18  while True:
19      datos, dirCliente = socket_serv.recvfrom(1024)
20      print(datos.decode())
21      fecha_hora = datetime.datetime.now()
22      fecha_hora_str = fecha_hora.strftime("%Y-%m-%d %H:%M:%S")
23      socket_serv.sendto(fecha_hora_str.encode('utf8'), dirCliente)
```

### cli5

```
1  import socket
2
3  socket_cli = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, 0)
4
5  #s.bind( (socket.INADDR_ANY, 9999) )
6
7  dir_serv = input('Dirección del servidor: ')
8  puerto_serv = input('Puerto del servidor: ')
9
10  if not dir_serv:
11      dir_serv = '127.0.0.1'
12
13  if not puerto_serv:
14      puerto_serv = '5555'
15
16  mensaje = '¿Qué hora es?'
17
18  socket_cli.sendto(mensaje.encode('utf8'), (dir_serv, int(puerto_serv)))
19
20  respuesta, dir = socket_cli.recvfrom(1024)
21  print(respuesta.decode())
22
23  socket_cli.close()
```

## P6 - app1

### serv6

```
1  import socket
2
3  socket_serv = socket.socket(socket.AF_INET, socket.SOCK_STREAM, 0)
4
5  dir_serv = input('Dirección del servidor: ')
6  puerto_serv = input('Puerto del servidor: ')
7
8  if not dir_serv:
9      dir_serv = '127.0.0.1'
10
11  if not puerto_serv:
12      puerto_serv = '5555'
13
14  socket_serv.bind((dir_serv, int(puerto_serv)))
15
16  socket_serv.listen(1)
17
18  while True:
19      ns, dirCliente = socket_serv.accept()
20      print('Conexión establecida')
21
22      path_fichero = ns.recv(1024)
23
24      try:
25          #Abrimos el archivo y leemos su contenido
26          archivo = open(path_fichero, 'rb')
27          mensaje = "Empieza"
28          while mensaje:
29              mensaje = archivo.read(1024)
30              if isinstance(mensaje, str):
31                  contenido = mensaje.encode('utf8')
32              elif isinstance(mensaje, bytes):
33                  contenido = mensaje
34              ns.send(contenido)
35
36      except Exception as e:
37          contenido = str(e).encode('utf8')
38
39      ns.close()
40
```

### cli6

```
1  import socket
2
3  socket_cli = socket.socket(socket.AF_INET, socket.SOCK_STREAM, 0)
4
5  dir_serv = input('Dirección del servidor: ')
6  puerto_serv = input('Puerto del servidor: ')
7
8
9  if not dir_serv:
10     dir_serv = '127.0.0.1'
11
12  if not puerto_serv:
13     puerto_serv = '5555'
14
15  socket_cli.connect((dir_serv, int(puerto_serv)))
16
17  path = input('Introduzca el nombre del fichero: ')
18
19  socket_cli.send(path.encode('utf8'))
20
21
22  full_msg = ''
23  datos = "Empieza"
24  while datos:
25      datos = socket_cli.recv(1024)
26      full_msg += datos.decode('latin-1')
27  print(full_msg)
28
29  socket_cli.close()

```

P7

serv7

```
1  import socket, select, sys
2
3  socket_cli = socket.socket(socket.AF_INET, socket.SOCK_STREAM, 0)
4
5  dir_serv = input('Dirección del servidor: ')
6  puerto_serv = input('Puerto del servidor: ')
7  nombre_cliente = input('Nombre de usuario: ')
8
9  if not dir_serv:
10     dir_serv = '127.0.0.1'
11
12  if not puerto_serv:
13     puerto_serv = '8888'
14
15  if not nombre_cliente:
16     nombre_cliente = 'usr'
17
18  socket_cli.connect((dir_serv, int(puerto_serv)))
19
20  socket_cli.send(nombre_cliente.encode('utf8'))
21
22  rlist = [sys.stdin, socket_cli]
23
24  msj_recibido = socket_cli.recv(1024).decode('utf-8')
25  print(msj_recibido)
26  while True:
27
28     ready_rlist = select.select(rlist, [], [])
29     for s in ready_rlist:
30         if s == socket_cli:
31             data = s.recv(1024)
32             if not data:
33                 break
34             print(data.decode())
35             #respuesta = socket_cli.recv(1024)
36             #print(respuesta.decode('utf8'))
37         else:
38             mensaje = input('>> ')
39             socket_cli.sendall(mensaje.encode('utf8'))
40
41
```



cli7

```
1  import socket, os, select
2
3  lista_usuarios = []
4  lista_clientes = []
5
6  socket_serv = socket.socket(socket.AF_INET, socket.SOCK_STREAM, 0)
7
8  dir_serv = input('Dirección del servidor: ')
9  puerto_serv = input('Puerto del servidor: ')
10
11  if not dir_serv:
12      dir_serv = '127.0.0.1'
13
14  if not puerto_serv:
15      puerto_serv = '8888'
16
17  socket_serv.bind((dir_serv, int(puerto_serv)))
18
19  socket_serv.listen(1)
20
21  def interaccion_cliente(nuevo_socket, usuario):
22      while True:
23          try:
24              mensaje = nuevo_socket.recv(1024).decode()
25              for cliente in lista_clientes:
26                  if cliente != nuevo_socket:
27                      cliente.send(f'{usuario}: {mensaje}'.encode('utf8'))
28                      #nuevo_socket.send(f'{usuario}: {mensaje}'.encode('utf8'))
29
30          except Exception as e:
31              pass
32              #print(f'ERROR: {str(e)}')
33
34  while True:
35      ns, direccion = socket_serv.accept()
36      print(f'Conexión entrante desde {direccion[0]}:{direccion[1]}')
37
38      lista_clientes.append(ns)
39
40      usuario = ns.recv(1024).decode()
41      lista_usuarios.append(usuario)
42
43      pid = os.fork() #creamos proceso hijo para manejar la conexión con el nuevo cliente
44
45      if pid == 0: # hijo
46          socket_serv.close()
47          ns.send('Bienvenido al chat'.encode('utf8'))
48          interaccion_cliente(ns, usuario)
49
50
```