Solve it by writing a program in Python.

Task 1:
Implement a program that will launch a specified process and periodically (with a provided time interval) collect the following data about it:

- CPU usage (percent);
- Memory consumption: Working Set and Private Bytes (for Windows systems) or Resident Set Size and Virtual Memory Size (for Linux systems);
- Number of open handles (for Windows systems) or file descriptors (for Linux systems).

Data collection should be performed all the time the process is running. Path to the executable file for the process and time interval between data collection iterations should be provided by user. Collected data should be stored on the disk. Format of stored data should support automated parsing to potentially allow, for example, drawing of charts.

Task 2:
Implement a program that synchronizes two folders: source and replica. The program should maintain a full, identical copy of destination folder at replica folder.

Requirements:

- Synchronization must be one-way: after the synchronization content of the replica folder should be modified to exactly match content of the source folder;
- Synchronization should be performed periodically;
- File creation/copying/removal operations should be logged to a file and to the console output;
- Folder paths, synchronization interval and log file path should be provided using the command line arguments.

Task 3:
Implement a client-server application that follows the next algorithm:

1. Server keeps ports 8000 and 8001 open.
2. Each client generates a unique identifier for itself.
3. Client connects to server port 8000, provides its unique identifier and gets a unique code from the server.
4. Client connects to server port 8001, provides a text message, its identifier and code that it received on step 2.
5. If client code does not match client identifier, server returns an error to the client.
6. If client code is correct, server writes the provided text message to a log file.

Server should be able to simultaneously work with at least 50 clients.
It is acceptable (although not required) to use a high-level protocol (e. g. HTTP) for communication between client and server.