**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No.** | **Date** | **Practical Name** | **Sign** |
| 1. |  | Write a program to take backup of mysql database |  |
| 2. |  | Write a program to restore mysql database |  |
| 3. |  | Use DriveImage XML to image a hard drive |  |
| 4. |  | Write a program to create a log file |  |
| 5. |  | Write a program to find a file in a directory |  |
| 6. |  | Write a program to find a word in a file |  |
| 7. |  | Create forensic images of digital devices from volatile data such as memory using Imager for: (i) Computer System; (ii) Server; (iii) Mobile Device |  |
| 10. |  | Create a new investigation case using Forensic Tool:  (i) Computer System; (ii) Computer Network; (iii) Mobile Device ;(iv) Wireless Network. |  |

## PRACTICAL NO: 1

**Aim:** Write a program to take backup of mysql database

#### INPUT:-

import java.io.IOException;

public class backup {

public static void main(String[] args)

{

String host = "localhost";

String port = "3306";

String databaseName = "employee";

String username = "root";

String password = "12345";

String backupPath = "D:\\Folder/backup.sql";

backupMySQLDatabase(host, port, databaseName, username, password, backupPath);

}

public static void backupMySQLDatabase(String host, String port, String databaseName, String username, String password, String backupPath)

{

try {

String command = "mysqldump -h " + host + " -P " + port + " -u " + username + " -p" + password + " " + databaseName + " -r " + backupPath;

ProcessBuilder processBuilder = new ProcessBuilder("cmd.exe", "/c", command);

Process process = processBuilder.start();

int exitCode = process.waitFor();

if

(exitCode == 0) {

System.out.println("Backup completed successfully.");

} else {

System.out.println("Backup failed. Please check the error message.");

}

} catch (IOException | InterruptedException e) {}

}

}

Note :

Make sure to replace placeholders like "your\_database\_name", "your\_username", "your\_password", and "path\_to\_backup\_directory" with your actual MySQL database details.

#### OUTPUT:-

If the backup is successful:

Backup completed successfully.

If there is an error during the backup process: Backup failed. Please check the error message.

## PRACTICAL NO: 2

**Aim:** Write a program to restore mysql database

#### INPUT:-

import java.io.IOException;

public class MySQLRestore {

public static void main(String[] args)

{

String host = "localhost";

String port = "3306";

String databaseName = "employee";

String username = "root";

String password = "12345";

String backupPath = "D:\\Folder/backup.sql";

restoreMySQLDatabase(host, port, databaseName, username, password, backupPath);

}

public static void restoreMySQLDatabase(String host, String port, String databaseName, String username, String password, String backupPath)

{

try {

String command = "mysql -h " + host + " -P " + port + " -u " + username + " -p" + password + " " + databaseName + " < " + backupPath;

ProcessBuilder processBuilder = new ProcessBuilder("cmd.exe", "/c", command);

Process process = processBuilder.start();

int exitCode = process.waitFor(); if (exitCode == 0) {

System.out.println("Database restore completed successfully.");

} else {

System.out.println("Database restore failed. Please check the error message.");

}

} catch (IOException | InterruptedException e) {}

}

}

Note: Replace placeholders like "your\_database\_name", "your\_username", "your\_password", and "path\_to\_backup\_directory" with your actual MySQL database details.

#### OUTPUT:-

If the restore is successful:

Database restore completed successfully

If there is an error during the restore process:

Database restore failed. Please check the error message

## PRACTICAL NO: 3

**Aim:** Use DriveImage XML to image a hard drive

#### INPUT:-

Note : This program assumes you have already installed DriveImage XML on your system.

import java.io.BufferedReader; import java.io.IOException; import java.io.InputStreamReader;

public class DriveImageXMLExample {

public static void main(String[] args) {

// Replace these values with your actual DriveImage XML installation path and desired image file

path

String driveImageXMLPath = "C:\\Program Files\\Runtime Software\\DriveImage

XML\\dixml.exe";

String sourceDrive = "C:"; // Replace with the drive you want to image

String destinationImagePath = "D:\\backup\_image.xml"; // Replace with the desired image file

path

// Create a disk image

createDiskImage(driveImageXMLPath, sourceDrive, destinationImagePath);

// Restore from the disk image (optional)

// restoreFromDiskImage(driveImageXMLPath, destinationImagePath, "E:"); // Replace with the destination drive

}

public static void createDiskImage(String driveImageXMLPath, String sourceDrive, String destinationImagePath) {

try {

String command = driveImageXMLPath + " /b /v /c /s" + sourceDrive + " /f" + destinationImagePath;

ProcessBuilder processBuilder = new ProcessBuilder("cmd.exe", "/c", command); Process process = processBuilder.start();

int exitCode = process.waitFor(); if (exitCode == 0) {

System.out.println("Disk image created successfully.");

} else {

System.out.println("Disk image creation failed. Please check the error message.");

printErrorStream(process);

}

} catch (IOException | InterruptedException e) { e.printStackTrace();

}

}

public static void restoreFromDiskImage(String driveImageXMLPath, String sourceImagePath,

String destinationDrive) { try {

String command = driveImageXMLPath + " /r /f" + sourceImagePath + " /s" + destinationDrive;

ProcessBuilder processBuilder = new ProcessBuilder("cmd.exe", "/c", command); Process process = processBuilder.start();

int exitCode = process.waitFor(); if (exitCode == 0) {

System.out.println("Restoration from disk image completed successfully.");

} else {

System.out.println("Restoration from disk image failed. Please check the error message."); printErrorStream(process);

}

} catch (IOException | InterruptedException e) { e.printStackTrace();

}

}

private static void printErrorStream(Process process) throws IOException { try (BufferedReader reader = new BufferedReader(new

InputStreamReader(process.getErrorStream()))) { String line;

System.out.println("Error message:"); while ((line = reader.readLine()) != null) {

System.out.println(line);

}

}

}

}

#### OUTPUT:-

Disk image created successfully.

## PRACTICAL NO: 4

**Aim:** Write a program to create a log file.

#### INPUT:-

import java.io.IOException;

import java.util.logging.FileHandler;

import java.util.logging.Level;

import java.util.logging.Logger;

import java.util.logging.SimpleFormatter;

public class LogFileExample {

public static void main(String[] args) {

Logger logger = Logger.getLogger(LogFileExample.class.getName());

try {

// Create a file handler to write log messages to a file

FileHandler fileHandler = new FileHandler("example.log");

// Create a simple text formatter for the log records

SimpleFormatter simpleFormatter = new SimpleFormatter(); fileHandler.setFormatter(simpleFormatter);

// Add the file handler to the logger

logger.addHandler(fileHandler);

// Set the logging level (FINEST, FINER, FINE, INFO, WARNING, SEVERE)

logger.setLevel(Level.INFO);

// Log some sample messages

logger.info("This is an informational message.");

logger.warning("This is a warning message.");

logger.severe("This is a severe error message.");

}

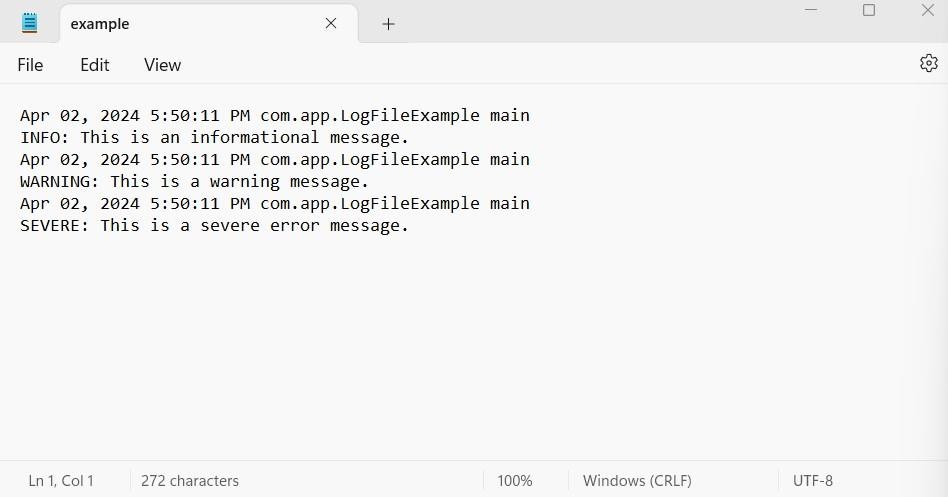
catch (IOException e) {}

}

}

|  |  |  |
| --- | --- | --- |
| Note : Compile and run this program, and you should see a new file ( | example.log | ) created in the |
| project directory with the logged messages. | | |

#### OUTPUT:-



## PRACTICAL NO: 5

**Aim:** Write a program to find a file in a directory

#### INPUT:-

a)

// Java Program to Search for a File in a Directory

import java.io.File;

public class Main {

public static void main(String[] args) {

// Create an object of the File class

File directory = new File("C:\\Users\\Aqsa Ulde\\OneDrive\\Documents\\NetBeansProjects\\MyFilename\\src");

// Store all file names in an array

String[] flist = directory.list();

boolean flag = false;

if (flist == null) {

System.out.println("Empty directory or directory not found.");

} else {

// Linear search in the array

for (String filename : flist) {

if (filename.equalsIgnoreCase("textfile.txt")) {

System.out.println(filename + " found");

flag = true;

break;

// Stop searching after finding the file

}

}

if (!flag) {

System.out.println("File Not Found");

}

}

}

}

b)

// Java Program to Search for a File in a Directory import java.io.File;

public class Main {

public static void main(String[] argv) {

// Create an object of the File class

File directory = new File("D:/New folder");

// Store all file names in an array

String[] flist = directory.list();

boolean flag = false;

if (flist == null) {

System.out.println("Empty directory or directory not found.");

} else {

// Linear search in the array

for (String filename : flist) {

if (filename.equalsIgnoreCase("Textfile.txt")) {

System.out.println(filename + " found");

flag = true;

break; // Stop searching after finding the file

}

}

if (!flag) {

System.out.println("File Not Found");

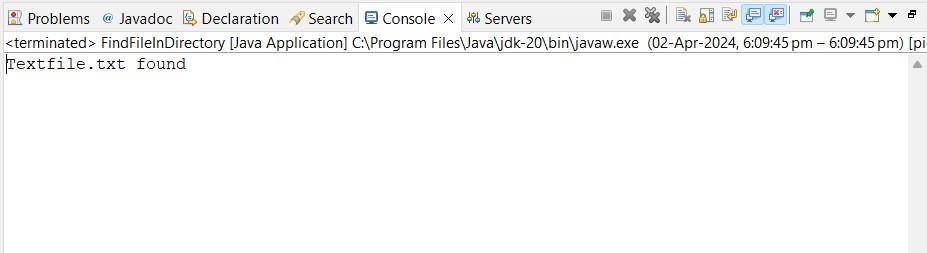
}

}

}

}

#### OUTPUT:-



## PRACTICAL NO: 6

**Aim:** Write a program to find a word in a file.

#### INPUT:-

#### import java.io.IOException;

#### import java.nio.file.Files;

#### import java.nio.file.Path;

#### import java.nio.file.Paths;

#### import java.util.Scanner;

#### public class FindWordInFile {

#### public static void main(String[] args) {

#### // Replace "path/to/your/file.txt" with the actual path to your text file

#### String filePath = "C:\\Users\\Aqsa Ulde\\OneDrive\\Documents\\NetBeansProjects\\FindWordInFile\\src\\example.log";

#### String targetWord = "WARNING";

#### 

#### try {

#### boolean wordFound = searchWordInFile(filePath, targetWord);

#### if (wordFound) {

#### System.out.println("The word '" + targetWord + "' was found in the file.");

#### } else {

#### System.out.println("The word '" + targetWord + "' was not found in the file.");

#### }

#### } catch (IOException e) {

#### }

#### }

#### public static boolean searchWordInFile(String filePath, String targetWord) throws IOException {

#### Path path = Paths.get(filePath);

#### 

#### // Check if the file exists

#### if (!Files.exists(path)) {

#### System.out.println("File not found: " + filePath);

#### return false;

#### }

#### 

#### try (Scanner scanner = new Scanner(path)) {

#### // Iterate through each line of the file

#### while (scanner.hasNextLine()) {

#### String line = scanner.nextLine();

#### // Check if the target word is present in the line

#### if (line.contains(targetWord)) {

#### return true; // Word found

#### }

#### }

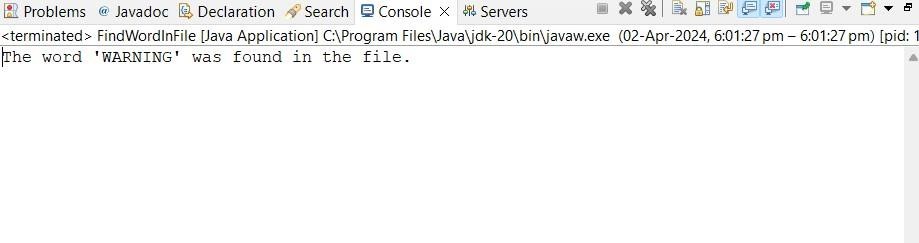
#### }

#### return false; // Word not found

#### }

#### }

#### OUTPUT:-

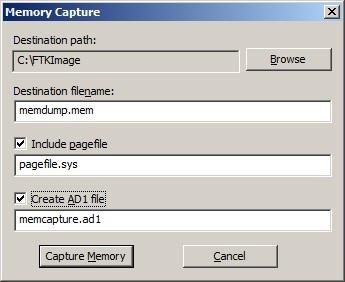


## PRACTICAL NO: 7

**Aim:** Create forensic images of digital devices from volatile data such as memory using Imager for: (i) Computer System; (ii) Server; (iii) Mobile Device.

Using FTK Imager:

FTK Imager is a graphical tool that provides a user-friendly interface for creating forensic images. It's available for Windows and Linux.

Download and install FTK Imager: AccessData FTK Imager. Open FTK Imager and go to File > Capture Memory.

Choose the target device, set the destination path, and click Start to create a memory image.



## PRACTICAL NO: 10

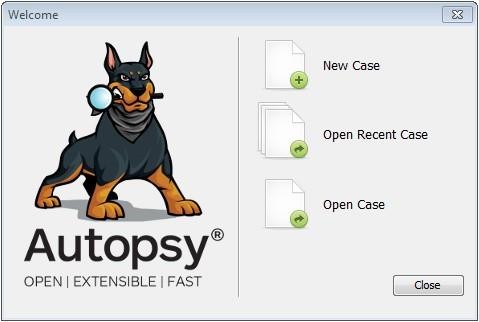
**Aim:** Create a new investigation case using Forensic Tool: (i) Computer System; (ii) Computer Network; (iii) Mobile Device; (iv) Wireless Network.

#### INPUT:-

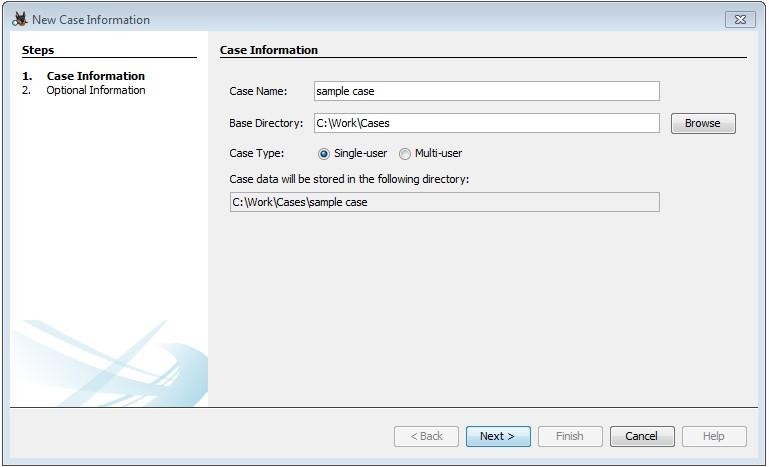
1. **Computer System:**

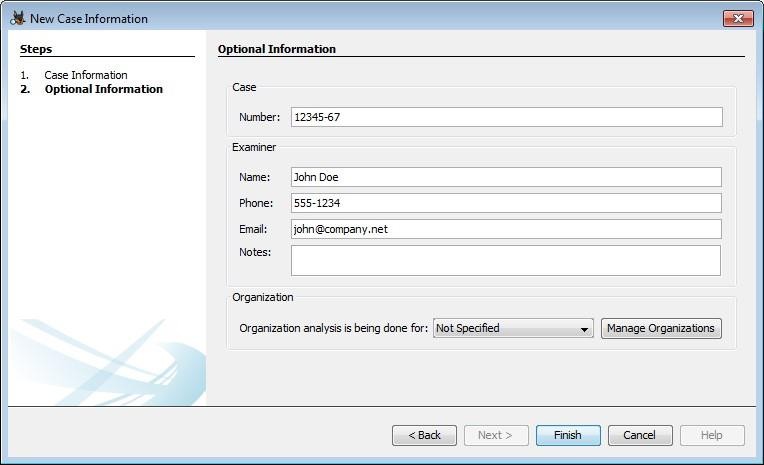
Forensic Tool: Autopsy

Download and Install Autopsy: Autopsy Open Autopsy and create a new case.

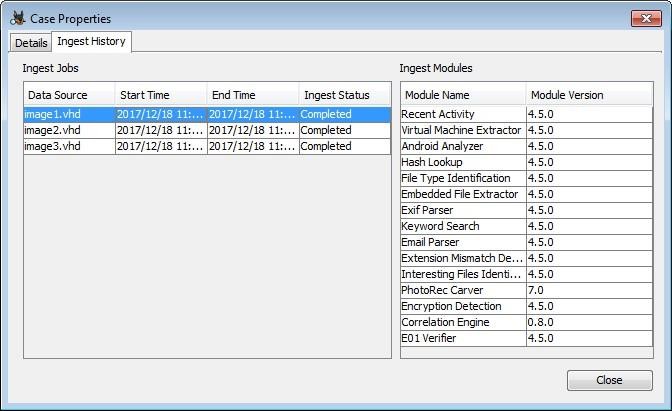


Enter case details, such as case name, case number, and investigator information.



Add a data source for the computer system (e.g., a disk image).

Start the analysis to examine file systems, recover deleted files, and explore system artifacts.



1. **Computer Network:**

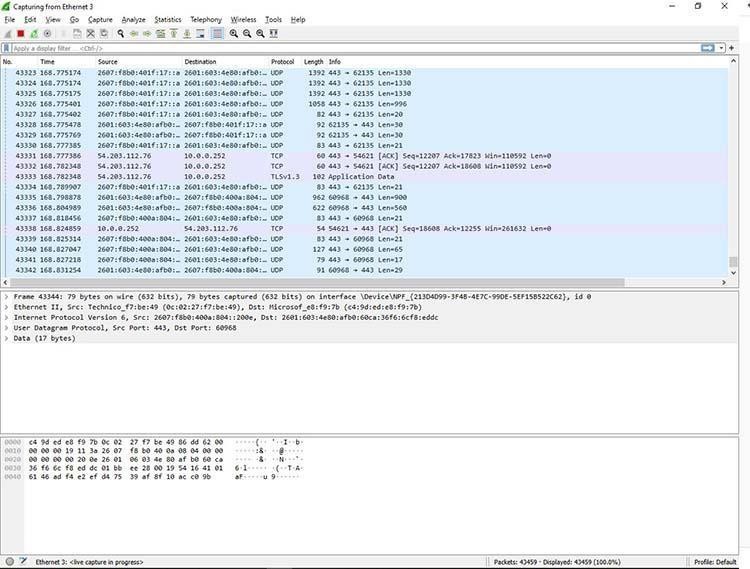
Forensic Tool: Wireshark

Download and Install Wireshark: Wireshark

Capture network traffic using Wireshark on the suspect system. Save the captured data to a file (PCAP format).

Open Wireshark and load the captured file.

Analyze the network traffic for any suspicious activities, such as unauthorized access or data exfiltration.



1. **Mobile Device:**

Forensic Tool: Cellebrite UFED (Universal Forensic Extraction Device) Acquire a Cellebrite UFED device or use the software version.

Connect the mobile device to the UFED device or software.

Follow the steps to create a new case in UFED.

Perform a forensic extraction of the mobile device's data.

Analyze the extracted data for evidence related to the investigation.

1. **Wireless Network:**

Forensic Tool: Aircrack-ng

Download and Install Aircrack-ng: Aircrack-ng

Capture wireless network traffic using Aircrack-ng tools like airodump-ng. Save the captured data to a file.

Use tools like aircrack-ng to analyze the captured data and crack WEP/WPA keys if necessary. Investigate any suspicious wireless network activities.