

İzmir University of Economics

2021-2022 Spring

SE311 Project

OS Modeling

Project #2

DEADLINE: 8 June 2022 18:00 PM (No late submissions)

1. Suppose you are designing a system that is capable of implementing many kinds of file systems. A Linux file system has Linux directories and Linux files, while a BSD file system and an NT file system have their own ways of representing directories and files. You want to be able to write reusable programs that do Input/Output to the file system. When you are creating directories and files, your system must create them for Linux, BSD, or NT as appropriate. To be more specific a Linux Directory cannot have an NT file in it. Every system will have only one filesystem.
2. Your compiler for the language DP you wrote must translate your programs I/O statements to the corresponding input/output API that the file system can understand. This is how your data is going to get written to a file in one of the three above Operating Systems. Your DP language print statement looks like the following: `int fprintf (File handle, String str);` where handle is the file object you have created. However, Linux and BSD filesystem use `int uprintf(String str, File handle);` and NT system use `int printf (byte[] charArrray, File handle);`
3. Your OS has devices and these devices provide data to applications because of an interrupt. When an interrupt is handled, the registered applications will be notified and the applications will take action. For instance, if a character arrives to the network port, the applications waiting for data will be notified and one of them will consume this data.
4. The OS will be using CPU, hard disk, and I/O devices. Before shutting down the OS each device must be properly "reset". Each device's reset behavior is different. Whenever we reset a device we must log a generic deviceReset event which does not change from device to device. In addition to having different reset behaviors, our devices have different reset interfaces. For example, for hard disk device a reset means first write whatever data in the buffer then close the files. For CPU it means terminate all processes. We would like to uniformly reset these devices. We want to reset the OS by issuing a shutdown command.

Notes:

- 1) No I/O is required for directories.
- 2) Remember you will not be writing an OS. You are just modeling it.

- Use at least 5 design patterns to solve this problem. Use only the ones that we have covered in the classes.
- You will not write a real-world application. You will simulate various functions of it.
- We will discuss some of the patterns in the coming weeks. So, either you can wait for the lecture, or you can proactively study the pattern and use it if applies to your problem.
- The whole design must be a coherent one. Do not just simply copy and paste things from your lecture examples. Adapt the examples. Use them as an inspiration or skeleton.
- **All the patterns must be connected to each other. The patterns must not be isolated from each other.**
- If you encounter any vagueness in the project description, and you cannot reach me feel free to make any assumption you want, if you state them very clearly. Be careful. Do not make unrealistic, farfetched assumptions. Do not add new functionality to make things unnecessarily complex.
- Please be creative. Choose meaningful names for your methods and classes.
- You will be graded proportional to the elegance of the solution.
- KEEP IT SIMPLE.

ACADEMIC INTEGRITY

This is a team project. All team projects must be completed by the members of the team without the aid of non-team members. **If a team member does not contribute to the implementation and the project report, his/her name should not appear on the work submitted for evaluation.** Plagiarism, copying, cheating, outsourcing the project to another person or organization for pay or without pay are considered as actions of academic dishonesty. Failure to maintain academic honesty may result in disciplinary action according to the Izmir University of Economics' disciplinary bylaw for students of institutions of higher education (<https://www.ieu.edu.tr/en/bylaws/type/read/id/13>).

Project Submission Guidelines

You will be providing

- a) a report that contains
 - i. a detailed account (in writing) of your thought process i.e. The patterns you choose and what made you to choose a pattern, participant mappings etc.
 - ii. Please pay attention to the format and appearance of the report. Use a spell checker, align the margins, use the same font and size, proper headings double spacing, use captions for the figures.
 - iii. UML diagram of the solution for the scenarios. You must use a software. The diagrams must be readable.
 - iv. An explanation in English of what each class does. Explanation is also needed for key methods in the classes.
- b) a Java implementation.

1. Members of the group are all expected to know all aspects of the solution intimately. I will schedule a conference meeting with group members and ask questions.
2. I must receive running programs. So, include a screen dump of your program.
3. **I will be paying attention to good programming style, i.e. indentation, comments, meaningful variable names etc.**
4. Submit a ZIP or JAR file.
5. Format of the submitted file:
<lastname1-lastname2-lastname3>PROJECT<number>.zip/.jar
(Example: **gunes-terim-vural-PROJECT1.zip**)
6. Do not send project meta files. Your zip/jar file must contain only java files. Please do not send the jdk/jre or IDE related jar files in the zip. ONLY YOUR source files.
No such file likes Project2(1).java, PROJETC1.txt, PROJECT.java, main.java, jre.jar etc.
7. Please limit the number of Java files. The number of Java files should not exceed the number of patterns used.
8. Write the names and last names of the group members on the first page of the source code listing in comments.
Example:
// ŞENOL GÜNEŞ
// FATİH TERİM
// YILMAZ VURAL
// Player Performance Monitoring System