**OS: Error Handling (Exception)**

Error handling can be divided into two parts: response and recovery procedures. It anticipates, detects and solves different types of errors.

Although there could be various errors, we will only consider a subset of them due to time limit and our simulated OS is so simple that it might not generate many errors. Specifically, errors could occur, but not limted in, when:

* File System: open/write/delete file without permission; quit without saving; attempt to use duplicated file names.
* Task Scheduling & Memory allocation: memory usage exceeds limit; process breaks down; memory allocation overlapping.
* Terminal: use of invalid instructions; file path not exists.

……

Different errors should trigger different responses. If the error is fatal, such as memory usage exceeding limit, our OS will cease all processes and start again. Otherwise, if the error is minor, our OS simply throws out a warning message.

Exceptions could be classified as **Faults**, **Traps**, **Aborts**. **Faults** can be corrected, the program can continue if nothing happen. **Traps** will be reported immediately after the trapping instructions are executed. **Aborts** refer to some severe error.

The following table lists errors, their types and explanation that we indended to simulate.

|  |  |  |
| --- | --- | --- |
| Name | Type | Explanation |
| Divide-by-zero | Fault | Occurs when dividing any number by 0 using the DIV or IDIV instruction. Or the result is too large |
| Bound Range Exceeded | Fault | Occurs when the current instruction is invalid. For example, undefined instructions in terminal |
| Invalid Opcode | Fault | Occurs when the processor or user tries to execute an invalid opcode |
| Page Fault |  | Occurs on illegal memory access |

Exceptions are classified as **Faults**, **Traps**, **Aborts**. **Faults** can be corrected, the program can continue if nothing happen. **Traps** will be reported immediately after the trapping instructions are executed. **Aborts** refer to some severe error.

In our project, the error handling mainly deals with:

Error handling refers to the response and recovery procedures from error conditions present in a software application. In other words, it is the process comprised of anticipation, detection and resolution of application errors, programming errors or communication errors. Error handling helps in maintaining the normal flow of program execution. In fact, many applications face numerous design challenges when considering error-handling techniques.