UNIX AND POSIX APIs

- **APIs – a set of application programming interface functions that can be called by user's programs to perform system specific functions**

Common functions

- **Determine system configuration and user information**
- **File manipulation**
- **Process creation and control**
- **Interprocess communication**
- **Network communication**

Common functions

- Determine system configuration and user information
  - **Examples**: `ioctl()` etc.
- Files Manipulation
  - **Examples**: `read()`, `write()`, `stat()` etc.
- Processes creation and control
  - **Examples**: `fork()`, `vfork()` etc.
- Interprocess communication
  - **Examples**: `msgget()`, `msgctl()`, `semget()`, `shmget()` etc.
- Network Communication
  - **Examples**: `socket()`, `connect()` etc.

API COMMON CHARACTERISTICS

- **API return -1 to indicate the execution has failed**

- **Global variable errno is set with an error code**

- **Perror function prints diagnostic message of the error to the standard output or calls streeror with errno as argument**

C++ program to print error msg using sterror

```
#include <string.h>
char *strerror(int errval);
```

Returns : pointer to message string

Program 2.1

```cpp
/* 2.1.cpp */
#include <errno.h>
#include <iostream>
#include <cstdio>
using namespace std;
int main()

{
   cerr << "EACCES : " << strerror(EACCES) << endl;
   return 0;
}
```

Output:

```
EACCES : Permission denied // message associated with the
                           // EACCES errno value
```

C++ program to print error msg using perror

```
#include <stdio.h>
void perror( const char *message );
```

Program 2.2

```cpp
/* 2.2.cpp */
#include <iostream>
#include <stdio.h>
#include <errno.h>
using namespace std;
int main()
{
    errno = EACCES;  // no need to declare as it is already
                     // done in errno.h

    perror("File");
    return 0;
}
```

Output:
```
File: Permission denied
```

ERROR STATUS CODE :: MEANING

- **EACCESS** :: No access to perform an operation via a API
- **EPERM** :: a API was aborted because the calling process doesnot have superuser privilege
- **ENOENT** :: an invalid filename was specified to an API
- **BADF** :: a API was called with an invalid file descriptor
- **EINTR** :: a API excecution was aborted due to signal interruption

- **EAGAIN** :: a API was aborted because system resource it requested was temporarily unavailable

- **ENOMEM** :: a API was aborted because it

    could not allocate dynamic memory

- **EIO** :: I/O error occurred in a API excecution

- **EPIPE** :: a API attempted to write to a

    pipe which has no reader

- **EFAULT** :: a API was passed an invalid address in one of its arguments

- **ENOEXEC** :: a API could not execute a program via one of the exec API

- **ECHILD** :: a process doesnot have any child process which it can wait on