LAB ASSIGNMENT

Anisha Bhandari

PAS078BEI005

Q) Describe and discuss the use case of following**:**

fork()

- **Description**: The fork() system call is used to create a new process by duplicating the calling process. The new process, known as the child process, runs concurrently with the parent process. After the fork() call, both the parent and child processes will execute the next instruction following the fork() call.
- **Use Case**: fork() is fundamental for process creation and multitasking. It's commonly used in scenarios where a process needs to spawn a new process to perform a separate task or to handle a new request, such as in web servers where each request might be handled by a separate process.

---

exec()

- **Description**: The exec() family of functions replaces the current process image with a new process image. It loads an executable file into the current process, terminating the current process and starting the new executable.
- **Use Case**: exec() is used to run a new program within the context of the current process. This is often used after a fork() in the child process to replace the process image with a different program. For instance, a shell uses exec() to execute a command typed by the user.

---

getpid()

- **Description**: The getpid() function returns the process ID (PID) of the calling process.
- **Use Case**: getpid() is useful for processes to identify themselves or to manage and coordinate with other processes. For instance, a parent process might use getpid() to log its own PID, and child processes might use it to determine their parent's PID.

---

wait()

- **Description**: The wait() function causes the calling process to pause execution until one of its child processes exits or a signal is received.

- **Use Case**: wait() is used by a parent process to ensure that it can collect status information from its child processes and avoid creating "zombie" processes. It's commonly used in scenarios where a parent needs to ensure that all of its child processes complete their execution.

---

stat()

- **Description**: The stat() function retrieves information about a file or directory, such as its size, permissions, owner, and timestamps. The information is stored in a structure with various fields representing different attributes.
- **Use Case**: stat() is used in file management and system programming to obtain detailed information about files or directories, which can be used for operations like checking file permissions, verifying file existence, or managing file metadata.

---

opendir()

- **Description**: The opendir() function opens a directory stream for reading the contents of a directory. The directory stream can be used with functions like readdir() to read directory entries and closedir() to close the directory stream.
- **Use Case**: opendir() is used in applications that need to access or analyze the contents of directories, such as file managers, backup tools, or scripts that process files in a directory.

---

readdir()

- **Description**: The readdir() function reads directory entries from a directory stream opened with opendir(). It allows iteration through the directory's contents to retrieve information about each file or subdirectory.
- **Use Case**: readdir() is used to traverse directory structures and perform operations on files or subdirectories within a directory. For example, it's used by file managers to list the contents of a directory.

---

close()

- **Description**: The close() function closes a file descriptor, which is an integer handle used to identify an open file, socket, or other I/O resource. Closing a file descriptor releases the associated resources and marks it as no longer in use.
- **Use Case**: close() is essential for resource management. It is used to release file descriptors and associated resources when they are no longer needed, preventing resource leaks and ensuring efficient use of system resources.