Daemon Processes

A **daemon process** is a background process that runs independently of any user control and performs specific tasks for the system. Daemons are usually started when the system starts, and they run until the system stops. A daemon process typically performs system services and is available at all times to more than one task or user. Daemon processes are started by the root user or root shell and can be stopped only by the root user.

- A daemon process runs in the background without direct user interaction.
- It is often started automatically when the system boots.
- Daemons usually detach from their parent processes and run independently.
- example - the "qdaemon" process, "sendmail" daemon etc.

How Daemon Processes are created

A daemon process is created by detaching a process from its parent and associating it with the `init` process. This is done in several steps: first, the process calls `fork()` to create a child process. Then, the child process calls `setsid()` to create a new session and detach from any terminal. Afterward, it changes its working directory to root (`/`), resets file permissions, and redirects standard input, output, and error to `/dev/null`. Finally, it runs in the background to perform its tasks.
Below figure explains the creation of Daemon Process:

- A daemon is usually created either by a process forking a child process and then immediately exiting, thus causing init to adopt the child process.
- Daemon can also be created by the init process directly launching the daemon.

Example

let's see an example of Daemon Process:

```
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <fcntl.h>

int daemon_initialise() {
   pid_t pid;

   // Step 1: Fork the parent process
   if ((pid = fork()) < 0) {
```

```
        return -1; // Fork failed
    } else if (pid != 0) {
        exit(0); // Parent process exits
    }

    // Step 2: Create a new session
    if (setsid() < 0) {
        return -1; // Failed to create a new session
    }

    // Step 3: Change working directory to root
    if (chdir("/") < 0) {
        return -1; // Failed to change directory
    }

    // Step 4: Reset file permissions
    umask(0);

    // Step 5: Close standard file descriptors
    close(STDIN_FILENO);
    close(STDOUT_FILENO);
    close(STDERR_FILENO);

    // Step 6: Redirect standard input, output, and error to /dev/null
    open("/dev/null", O_RDONLY); // Redirect standard input
    open("/dev/null", O_WRONLY); // Redirect standard output
    open("/dev/null", O_WRONLY); // Redirect standard error

    return 0; // Daemon initialized successfully
}

int main() {
    if (daemon_initialise() < 0) {
        // Failed to initialize daemon
        exit(EXIT_FAILURE);
    }

    // Daemon logic (e.g., perform tasks in the background)
    while (1) {
        // Example: Sleep for 10 seconds
        sleep(10);
    }

    return 0;
}
```