Assignment 119: How does waitpid( ) prevent creation of Zombie or Orphan processes?

The `waitpid()` function plays a significant role in preventing the creation of zombie and orphan processes. When a parent process creates a child process using the `fork()` function, it is responsible for reaping the child process once it has finished executing. If the parent process exits before the child process, the child process becomes a zombie, as it is no longer associated with a parent process. A zombie process consumes system resources without performing any useful tasks and can cause issues in process management.

The `waitpid()` function allows a parent process to wait for the completion of a specific child process. It takes two arguments: the `pid_t pid` specifying the process ID of the child process to wait for, and an `int *statloc` pointer to store the exit status of the child process. If the child process with the specified `pid` has already exited, `waitpid()` returns immediately. If the child process is still running, `waitpid()` blocks the parent process until the child process terminates.

By using `waitpid()`, the parent process ensures that it waits for the correct child process to finish, thus preventing the creation of zombie processes. If a child process terminates before the parent process calls `waitpid()`, the kernel automatically places the child process in the zombie state. The parent process can then call `waitpid()` to retrieve the child's exit status and clean up the zombie process.

In the context of the 'Zombies and Orphans' example, the parent process uses `waitpid()` to wait for the child process to finish. This ensures that the child process is properly reaped and prevents it from becoming a zombie. If the parent process did not use `waitpid()` and instead relied on the default `wait()` function, it might miss the child process if it exits before the parent checks for completed child processes. In such cases, the child process would become a zombie, leading to potential issues in process management.