-Assignment 4.3-
Various Possibilities on Creating and Terminating a Process

Suyash Mahar
ECE - 16116069

January 27, 2019

1 Creating a Process

On creating a process following possiblilities exists:

1. The parent can continue to execute concurently with its children.
2. The parent waits until some or all of its children have terminated.

Regarding address space of the child process again two possibilities exists for the process:

1. The child process has a duplicate address space from the parent. (**Unix/Linux** style.)
2. The child process has a new program loaded into it (**Windows** style).

2 Terminating a Process

When a process terminates it returns an exit code which represents the status of the process when it exited. When this occurs all the resources allocated to the process are deallocated by the operating system.

Various possibilities during the termination of a process are listed below:

1. **Cascading Termination**: Some systems terminate a process's children as soon as the parent process terminates. This is known as cascading termination. This is done regardless of normal or abnormal exit of the parent.
2. **Zombie Process**: (*Unix/Linux*) When a child process exits and it's parent process hasn't retrived it's exit status using ***wait()*** syscall the child processes becomes a zombie process. Such processes are already terminated but their exit status is still maintained by the process table. Such processes are however usually short lived, as soon as the parent calls ***wait()*** the exit status is returned and the entry from the process table is freed.
3. **Orphan Process**: When process terminates without waiting (using ***wait()*** syscall) for it's children to terminate, the children become orphan process. Linux/Unix in such case assigns the *init* process as the new parent of the child processes.