# Lecture Summaries

## l6

parents process create children processes, which create other processes . process identified and managed via a process identifier (pid) Resource sharing options Parent and children share all resources Children share subset of parent’s resources . Process’ resources are deallocated by operating system Parent may terminate the execution of children processes using the abort() system call .

the parent process may wait for termination of a child process by using the wait()system call . the call returns status information and the pid of the terminated process pid = wait(&status); If no parent waiting (did not invoke wait()) process is a zombie . process is an orphan Context Switch .

limits OS support for Single foreground process- controlled via user interface Multiple background processes– in memory, running, but not on the display, and with limits Limits include single, short task, receiving notification of events, specific long-running tasks like audio playback Multi-process Application .

IPC – Message Passing Direct Communication Processes need Interprocess communication (IPC) Two models of IPC: Message passing Shared Memory An area of memory shared among the processes that wish to communicate . major issues is to provide mechanism that will allow the user processes to synchronize their actions when they access shared memory .