**سیستم عامل جلسه 8**

**بن بست ها (Dead locks)**

در محیط چند برنامه ای ممکن است چندین فرایند برای تعداد محدودی از منابع با هم رقابت کنند. فرایند، منابعی را درخواست می کند و چنان چه این این منابع در آن زمان فراهم نباشد، فرایند به حالت انتظار می رود. ممکن است منابع درخواستی این فرایند دراختیار فرایند دیگری باشند که در حال انتظار هستند و این فرایند هرگز از حالت انتظار خارج نشود این وضعیت را بن بست می گویند

شاید بهترین مفهوم از بن بست، از قانونی استنباط شود که قانون گذازان به تصویب رساندند. بخشی از قانون می گوید "وقتی دو قطار در یک تقاطع به هم نزدیک می شوند و هر دو بایدکاملا باستند و هیچ کدام نباید حرکت کنند، مگر این که دیگری رفته باشد".

در این فصل، روش هایی را توصیف می کنیم که سیستم عامل می تواند برای جلوگیری از بن بست یا اداره کردن آن به کار گیرد. گرچه بعضی از برنامه های کاربری می توانند تشخیص دهند چه برنامه هایی دچار بن بست می شوند،ولی سیستم های عامل امکاناتی برای پیشگیری از بن بست ندارند و برنامه نویس باید تضمین کند که برنامه های آن ها فاقد بن بست باشد. با توجهبه گزایش های فعلی؛ مثل تعداد زیاد فرایند ها، برنامه های چند نخی(multi thread، منابع زیاد در یک سیستم و تاکیدبر سرور های پایگاه داده و فایل هایی با طول عمر زیاد به جای سیستم های دسته ای، مساله های بن بست در حال متداول شدن هستند.

**مدل سیستم**

هر سیستم متشکل از تعداد محدودی از منابع است که باید بین فرایند های متقاضی و رقیب توزیع شود. این منابع به چندین نوع تقسیم می شوند که هر کدام ممکن است شامل چند نمونه ی یکسان باشند

فضای حافظه، چرخه های پردازنده، فایل ها، دستگاه های IO (مثل چاپگر و گرداننده های DVD) از انواع منابع(Resource) هستند.

اگر فرایند نمونه ای از یک نوع منبع را درخواست کند، تخصیص هر نمونه از آن نوع، ان درخواست را برآورده می کند