

## اطلاعات (الف)

PCB در ساختار داده‌های kernel ذخیره می‌شود جایی که در دسترس برنامه‌های سطح یوزر نباشد.

نقش PCB در Context switch

اطلاعات که همان Context است در PCB ذخیره می‌شود

وقتی CPU را از هم می‌گیریم می‌بینیم که  $P_1$  و اطلاعات مربوط به پردازش هم و

رجیسترها و وضعیت CPU را برای  $[PCB P_0]$  می‌بینیم و

$PCB - [P_1]$  را هم می‌بینیم

اطلاعات مربوط به رجیسترهای CPU که داشته‌اند هم را اجزای کردن باید در CPU PCB

state های پردازش ها که در حال اجرا هستند یا خیر را برای PCB باید ذخیره کنیم در زمان Context Switch

(ب) در interrupt ها از interrupt vector برای ذخیره آدرسی ها استفاده می‌کنند

هر interrupt باید آدرسی روی جدولی شود

وقتی که وقفه ایجاد می‌شود، CPU اجزای آن پردازش را متوقف می‌کند و می‌رود

این وقفه به CPU با رجیسترهایی که مربوط به برنامه‌ها یا (توابع)

در حال اجرا است را save می‌کند و می‌کشد. می‌گذارد روی stack تا آدرسی ها

را داشته باشد. وقتی اجزای پردازش روشن می‌شود می‌رود به stack و pop می‌کند

و ادامه حیات اجزای پردازش قبلی را