**Nama : Riski Agung Putro Laksono**

**NIM : L200150110**

**Makul : Organisasi Arsitektur Komputer**

**Fungsi Interrupt**

Fungsi interupsi adalah mekanisme penghentian atau pengalihan pengolahan instruksi dalam CPU kepada routine interupsi. Hampir semua modul (memori dan I/O) memiliki mekanisme yang dapat menginterupsi kerja CPU.

Tujuan interupsi secara umum untuk menejemen pengeksekusian routine instruksi agar efektif dan efisien antar CPU dan modul – modul I/O maupun memori. Setiap komponen komputer dapat menjalankan tugasnya secara bersamaan, tetapi kendali terletak pada CPU disamping itu kecepatan eksekusi masing – masing modul berbeda sehingga dengan adanya fungsi interupsi ini dapat sebagai sinkronisasi kerja antar modul. Macam – macam kelas sinyal interupsi :

-     *Program*, yaitu interupsi yang dibangkitkan dengan beberapa kondisi yang terjadi pada hasil eksekusi program. Contohnya: arimatika overflow, pembagian nol, oparasi ilegal.

-     *Timer*, adalah interupsi yang dibangkitkan pewaktuan dalam prosesor. Sinyal ini memungkinkan sistem operasi menjalankan fungsi tertentu secara reguler.

-     *I/O*, sinyal interupsi yang dibangkitkan oleh modul I/O sehubungan pemberitahuan kondisi error dan penyelesaian suatu operasi.

-     *Hardware failure,* adalah interupsi yang dibangkitkan oleh kegagalan daya atau kesalahan paritas memori.

Dengan adanya mekanisme interupsi, prosesor dapat digunakan untuk mengeksekusi instruksi – instruksi lain. Saat suatu modul telah selesai menjalankan tugasnya dan siap menerima tugas berikutnya maka modul ini akan mengirimkan permintaan interupsi ke prosesor. Kemudian prosesor akan menghentikan eksekusi yang dijalankannya untuk menghandel routine interupsi. Setelah program interupsi selesai maka prosesor akan melanjutkan eksekusi programnya kembali. Saat sinyal interupsi diterima prosesor ada dua kemungkinan tindakan, yaitu interupsi diterima/ditangguhkan dan interupsi ditolak. Apabila interupsi ditangguhkan, prosesor akan melakukan hal – hal seperti :

1. Prosesor menangguhkan eksekusi program yang dijalankan dan menyimpan konteksnya. Tindakan ini adalah menyimpan alamat instruksi berikutnya yang akan dieksekusi dan data lain yang relevan.

2. Prosesor menyetel program counter (PC) ke alamat awal routine *interrupt handler*.

Berikut menjelaskan siklus eksekusi oleh prosesor dengan adanya fungsi interupsi.

  Untuk sistem operasi yang kompleks sangat dimungkinkan adanya interupsi ganda (*multiple interrupt*). Misalnya suatu komputer akan menerima permintaan interupsi saat proses pencetakan dengan printer selesai, disamping itu dimungkinkan dari saluran komunikasi akan mengirimkan permintaan interupsi setiap kali data tiba. Dalam hal ini prosesor harus menangani interupsi ganda.

Dapat diambil dua buah pendekatan untuk menangani interupsi ganda ini.

      Pertama adalah menolak atau tidak mengizinkan interupsi lain saat suatu interupsi ditangani prosesor. Kemudian setelah prosesor selesai menangani suatu interupsi maka interupsi lain baru di tangani.

Pendekatan ini disebut *pengolahan interupsi berurutan / sekuensial*. Pendekatan ini cukup baik dan sederhana karena interupsi ditangani dalam ututan yang cukup ketat. Kelemahan pendekatan ini adalah metode ini tidak memperhitungkan prioritas interupsi. (Gambar a)

      Pendekatan kedua adalah dengan mendefinisikan prioritas bagi interupsi dan *interrupt handler* mengizinkan interupsi berprioritas lebih tinggi ditangani terlebih dahulu. Pedekatan ini disebut *pengolahan interupsi bersarang*. (Gambar b)

O   Sebagai contoh untuk mendekatan bersarang, misalnya suatu sistem memiliki tiga perangkat I/O: printer, disk, dan saluran komunikasi, masing – masing prioritasnya 2, 4 dan 5.

Pada awal sistem melakukan pencetakan dengan printer, saat itu terdapat pengiriman data padasaluran komunikasi sehingga modul komunikasi meminta interupsi. Proses selanjutnya adalah pengalihan eksekusi interupsi mudul komunikasi, sedangkan interupsi printer ditangguhkan. Saat pengeksekusian modul komunikasi terjadi interupsi disk, namun karena prioritasnya lebih rendah maka interupsi disk ditangguhkan. Setelah interupsi modul komunikasi selesai akan dilanjutkan interupsi yang memiliki prioritas lebih tinggi, yaitu disk. Bila interupsi disk selesai dilanjutkan eksekusi interupsi printer. Selanjutnya dilanjutkan eksekusi program utama.