**Nama : Muhammad Yulianto**

**NIM : L200150057**

**Prioritas Interupsi**

Interupsi dengan prioritas yang lebih tinggi akan didahulukan, artinya, program interupsinya akan diprioritaskan untuk dieksekusi terlebih dahulu.

Jika interupsi dengan prioritas yang sama terjadi bersamaan, maka program interupsi akan dijalankan secara berurutan dari nomor subrutin interupsinya.

Jika interupsi dengan prioritas yang lebih tinggi muncul ketika subrutin interupsi dengan prioritas yang lebih rendah sedang dieksekusi, maka program akan melompat dulu ke subrutin interupsi dengan prioritas yang lebih tinggi, dan setelah subrutinnya selesai dieksekusi, program akan kembali melanjutkan subrutin interupsi yang ditinggalkannya tadi.

**Vektor Interupsi dan Vector Chaining**

Bila ada sebuah sinyal interupsi pada interrupt request line, bagaimana sebuah interrupt handler mengetahui sumber dari interupsi itu? Apakah harus menelusuri semua sumber interupsi satu-persatu? Tidak perlu, karena setiap interrupt handler mempunyai alamat memori masing-masing. Alamat ini adalah offset pada sebuah tabel yang disebut dengan vektor interupsi.

**Tabel 10.1. Tabel Vector-Even pada Intel Pentium**

|  |  |
| --- | --- |
| Vector number | Description |
| 0 | Divide error |
| 1 | Debug Exception |
| 2 | Null Interrupt |
| 3 | Breakpoint |
| 4 | INTO-detected overflow |
| 5 | Bound range exception |
| 6 | Invalid opcode |
| 7 | Device not available |
| 8 | Double fault |
| 9 | Compressor segment overrun (reserved) |
| 10 | Invalid task state segment |
| 11 | Segment not present |
| 12 | Stack fault |
| 13 | General protection |
| 14 | Page fault |
| 15 | (Intel reserved, do not use) |
| 16 | Floating point error |
| 17 | Alignment check |
| 18 | Machine check |
| 19-31 | (Intel reserved, do not use) |
| 32-255 | Maskable interrupt |

Sesuai dengan perkembangan zaman, komputer mempunyai lebih banyak perangkat (dan lebih banyak interrupt handlers) daripada elemen alamat di vektor. Hal ini bisa diatasi dengan teknik vector chaining. Setiap elemen di vektor interupsi menunjuk ke kepala dari sebuah daftar interrupt handlers, sehingga bila ada interupsi, handler yang terdapat pada daftar yang ditunjuk akan dipanggil satu persatu sampai didapatkan handler yang bisa menangani interupsi yang bersangkutan.