

# Sistem Operasi

Rudi Rosadi, S.Si., M.Kom /  
Rahmatullah Arrizal P, S.Kom, M.T

# Aturan Perkuliahan

Nama MK : Sistem Operasi

Kelas : TI/3/A dan TI/3/B

Prodi : S1 Teknik Informatika

SKS : 3 SKS

Waktu :

Kelas A: Jum'at , 08:30 -10:30

Kelas B : Jum'at, 13:30 - 15:30

# Aturan Perkuliahan

- Dilaksanakan secara Hybrid
- Mengisi Absensi
- Menggunakan LMS Live Unpad, Aplikasi Meeting, Video Pembelajaran
- Total Pertemuan 16 (Termasuk Quis, UTS, UAS)
- Tidak Ada Susulan Quiz/Tugas
- Tidak Ada Susulan Ujian (UTS/UAS) Tanpa ada Konfirmasi Sebelumnya

# Aturan Penilaian

- Tugas : 10 %
- Quiz : 10 %
- Presentasi : 20%
- UTS : 20 %
- UAS : 40 %

# Overview Materi

- Konsep perangkat keras,
- Pengertian dasar sistem operasi ,
- Konsep proses berikut penjadwalan,
- Masalah kongkurensi seperti mutual exclusion,
- Sinkronisasi, deadlock dan starvation,
- Manajemen memori meliputi real memori dan virtual memori, Manajemen I/O, Manajemen File dan Keamanan Sistem.
- Studi kasus yang diambil mengenai Linux dan Windows, OSX, dll

## Referensi

- Hariyanto,B, Sistem Operasi.. Bandung: Penerbit Informatika
- Tanenbaum, A.S. Modern Operating System. New Jersey: Prentice Hall
- Silberschatz, A dan Galvin,P. Operating System Concept. MA : Addison Wesley
- Sri Kusumadewi, Sistem Operasi: Penerbit Graha Ilmu
- Stalling, W. Operating System. New Jersey:Prentice Hall

# Trivia

20% dari apa yang kita dengar

30% dari apa yang kita baca

40% dari apa yang kita lihat prosesnya

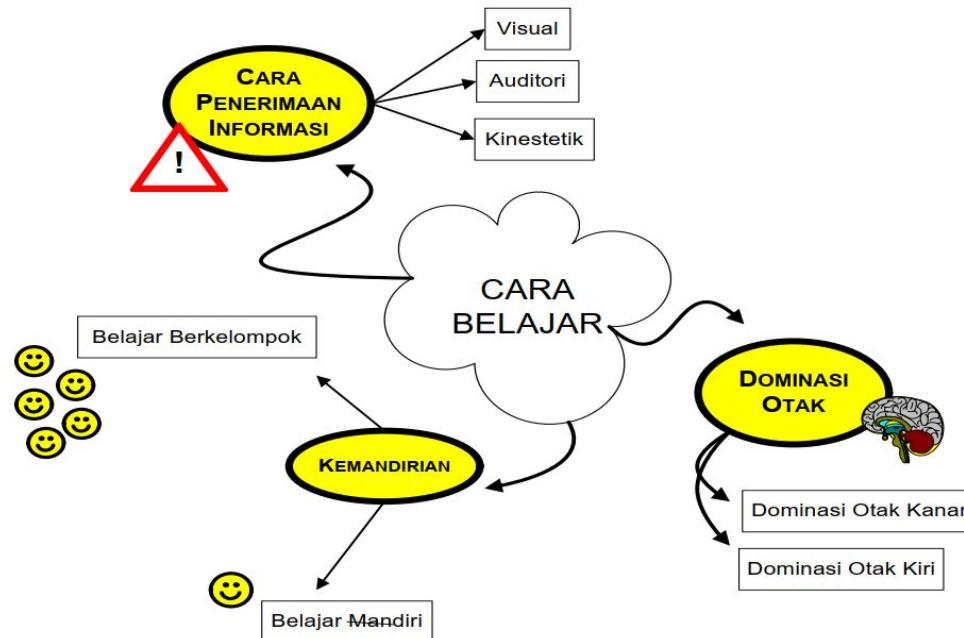
50% dari apa yang kita katakan

60% dari apa yang kita kerjakan

90% dari apa yang kita lihat, dengar, katakan dan kerjakan

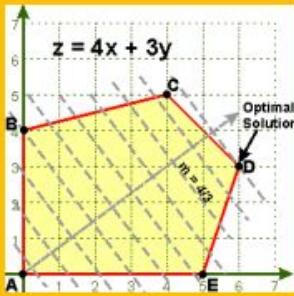
\*Rose, Colin dan Malcolm J. Nicholl, Accelerated Learning for the 21st Century

# Cara Belajar



\*Ichsan S Putra, Ariyanti Pratiwi, 2004, Strategi Sukses di Kampus, Penerbit ITB

# Strategi Belajar Visual



Gunakan gambar, peta, grafik

Visualisasikan materi dalam benak



## STRATEGI BELAJAR VISUAL



Pahami materi lewat bacaan atau gambar

Beri warna & gambar pada catatan

long desire to gain for job and study. To for success spend a lot of labor. Learn,

# Strategi Belajar Auditori

## STRATEGI BELAJAR AUDITORI

Dengar baik-baik penjelasan dosen di kelas

Terangkan secara lisan suatu pelajaran pada teman

Gunakan nada-nada lagu untuk mengingat informasi

Bericara dengan keras untuk menerangkan informasi baru

Lakukan Pengulangan dengan suara keras dan cukup sering

Bacalah dengan suara keras

Bentuk Kelompok Belajar

# Strategi Belajar Kinestetik



Libatkan sebanyak mungkin anggota tubuh:

1. Eksperimen di Laboratorium
2. Menggunakan komputer untuk mengetik informasi yang kita pelajari



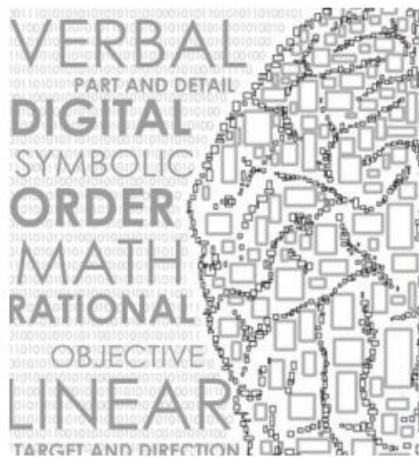
Perbanyak latihan soal



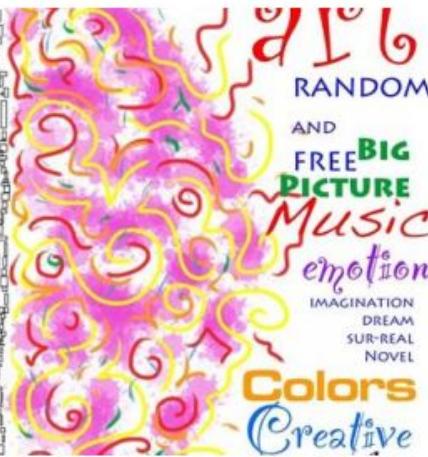
Seusai belajar, coba tuliskan poin-poin pentingnya



# Cara Belajar Berdasarkan Dominasi Otak



Otak  
Kiri



Otak  
Kanan



You can do any one way

Is your right thumb or the left on top?

Dominasi  
otak  
kanan

Dominasi  
otak kiri

# Cara Belajar Berdasarkan Dominasi Otak

| No. | Strategi Belajar Otak Kiri                         | Strategi Belajar Otak Kanan             |
|-----|--|---|
| 1   | Pelajari dahulu prinsip-prinsip dasar              | Cari gambaran global dulu               |
| 2   | Lihat keterkaitan antar prinsip dasar              | Buat peta pikiran                       |
| 3   | Pelajari bahan secara berurutan                    | Bagi waktu belajar menjadi sesi singkat |
| 4   | Tandai bagian yang tidak dimengerti lalu lanjutkan | Buat daftar hal yang harus dipelajari   |

# Cara Belajar Berdasarkan Kemandirian

| No. | Dominan Belajar Berkelompok                    | Dominan Belajar Sendiri   |
|-----|--|---|
| 1   | Pilih teman belajar yang cocok                 | Perbanyak literatur & latihan soal yang digunakan   |
| 2   | Saling bandingkan catatan                      | Pilih tempat belajar yang tenang untuk mencoba memahami terlebih dahulu materi sebelum berdiskusi dengan orang lain |
| 3   | Tanya jalan penggerjaan, bukan sekadar jawaban | Setelah memahami materi, cobalah berdiskusi dengan orang lain   |
| 4   | Rutinkan Kegiatan Belajar Bersama              |   |

# Manajemen Waktu

- Manajemen waktu yang baik penting untuk kesuksesan studi.
- Dengan merencanakan penggunaan waktu memudahkan anda mendistribusi tugas dan pekerjaan sepanjang semester, menghindari ‘kemacetan’ pekerjaan, dan menangani stress.
- Banyak pekerjaan/tugas di kampus yang deadline pada waktu bersamaan. Jika tidak “diatur” dengan baik, tidak mungkin dapat dikerjakan semua secara baik.
- Untuk dapat mengerjakan semua pekerjaan/tugas secara maksimal, anda harus menyebar beban kerja sepanjang semester (tidak ada sistem kebut semalam (SKS) yang memberikan hasil yang baik)
- Dengan manajemen waktu anda menentukan apa yang harus dikerjakan dan kapan mengerjakannya
- Dengan manajemen waktu yang baik anda dapat meraih hidup yang lebih “berwarna” dengan alokasi waktu yang cukup untuk belajar, bermain, bersosialisasi, beribadah, dsb
- Manajemen waktu tidak hanya penting selama anda kuliah, tapi juga dalam bekerja dan dalam hidup
- Bagaimana memanfaatkan waktu secara efektif

# Tingkat-tingkat Abstraksi Organisasi Komputer

# Software

## Application S/W

- Bahasa / Compilers sbg translator (source to executable program)
- System S/W
  - Operating Systems Program / perangkat lunak yang mengontrol eksekusi program-program aplikasi dan berfungsi sebagai perantara (interface) antara pemakai komputer (user) dan komputer (hardware)

# Hardware

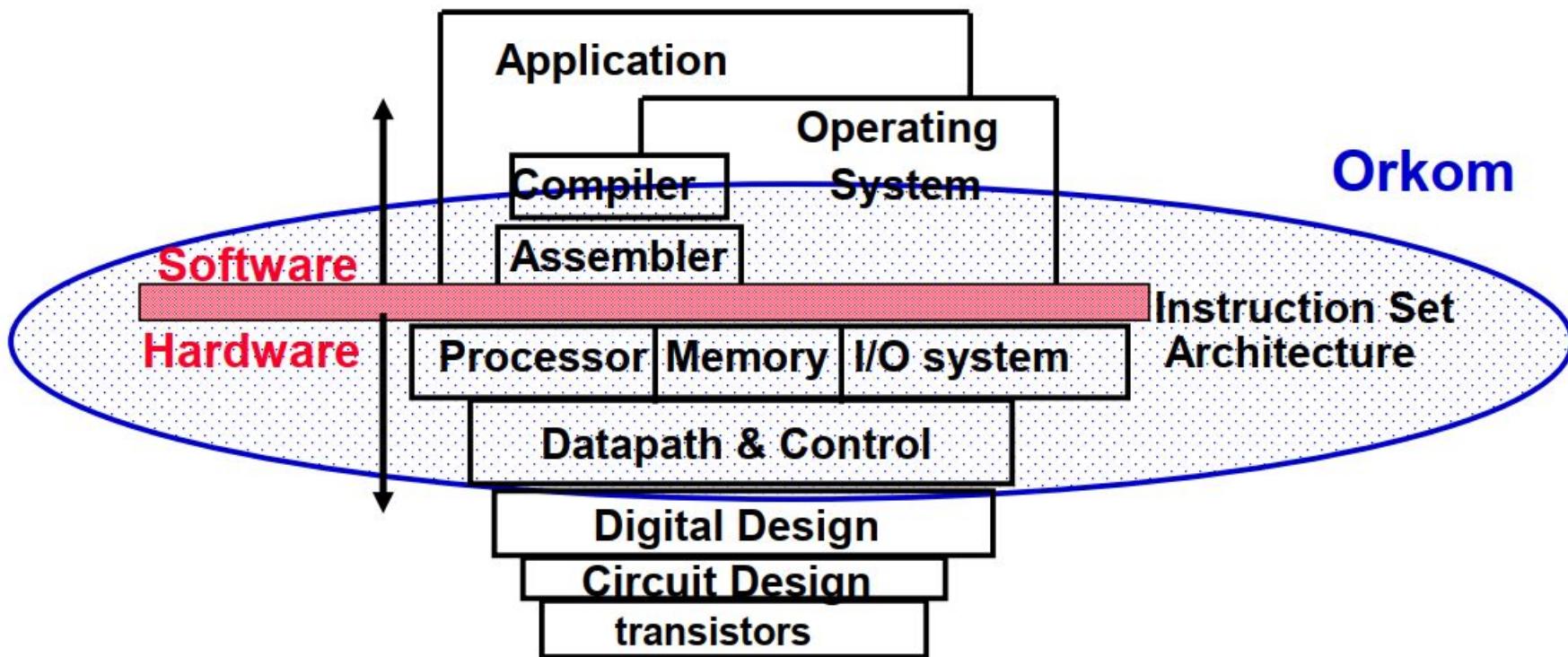
3 components: Processor (ALU, Register, Control), Memory, dan  
Input/Output

- Instruction Set
  - What basic operations can be carried out
  - What, where, and how data can be stored & retrieved in/from memory
  - How can data be exchanged to the outside “world

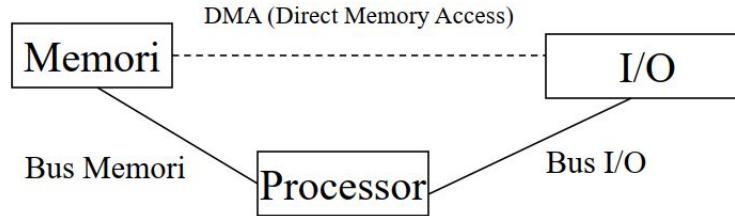
# Brainware

- Seorang manusia yang mampu menggunakan dan menjalankan perangkat lunak atau keras pada sebuah komputer.
- Tugas utama dari orang ini adalah untuk merancang bagaimana suatu perangkat akan bekerja sesuai dengan hasil yang diinginkan.

# Organisasi Sistem Komputer



# H/W Secara Fungsional



## ° Processor

- Arithmetics & Logic Unit (ALU)
- Register
- Control Unit (CU)



## ° Memori

- Internal (RAM, ROM, Cache)
- External (H/D, F/D, CD, FlashDisk, dll)

## ° Input / Output

## ° Bus

- Bus Alamat, Bus Data dan Bus Control



# Processor

- Pemroses disebut CPU, berfungsi mengendalikan operasi komputer dan melakukan pengolahan data.
- Pemroses melakukan kerja dengan langkah sbb:
  - Mengambil instruksi yang dikodekan secara biner dari memori utama
  - Men-dekode instruksi menjadi proses-proses sederhana
  - Melaksanakan proses-proses tersebut
- Operasi-operasi pada pemroses dikategorikan menjadi:
  - Operasi aritmetika
    - Penambahan, pengurangan, perkalian, pembagian dsb
  - Operasi logika
    - OR, AND, X-OR, inversi dsb
  - Operasi pengendalian
    - Operasi percabangan, lompat dsb

# Komponen Processor

Pemroses terdiri dari tiga komponen, yaitu:

- CU (Control Unit)
  - Berfungsi mengendalikan operasi yang dilaksanakan sistem komputer
- ALU (Arithmetic Logic Unit)
  - Berfungsi melakukan operasi aritmatika dan logika
- Register
  - Merupakan memori yang sangat cepat berfungsi sebagai tempat transaksi dari operasi yang akan dilakukan oleh pemroses

# Satuan Waktu kecepatan Proses Komputer

| <b>Satuan waktu</b> |      | <b>Kecepatan</b>                               |
|---------------------|------|--|
| Millisecond         | (ms) | Ribu operasi perdetik (1/1000)                 |
| Microsecond         | (us) | Juta operasi perdetik (1/1000.000)             |
| Nanosecond          | (ns) | Milyard operasi per detik (1/1000.000.000)     |
| Picosecond          | (ps) | Triliun operasi per detik (1/1000.000.000.000) |

# Memori

- Memori berfungsi untuk menyimpan data dan program.
- Hierarki memori berdasarkan kecepatan akses, seperti tabel berikut ini:

satuan memory:

- 1 byte = 8 bit atau 1 karakter
- 1 kb (kilobyte) = 1024 byte
- 1 mb (megabyte) = 1024 kb atau 1.048.576 byte
- 1 gb (gigabyte) = 1024 mb atau 1.048.576 kb atau 1.073.741.824 byte
- 1 tb (terabyte) = 1.009.511.627.776 bit atau 137.438.953.472 byte

|           |                          |
|-----------|--------------------------|
| Tercepat  | Register                 |
|           | Cache memory             |
|           | Main memory              |
|           | Disk memory (Flash Disk) |
|           | Magnetic Disk            |
|           | Magnetic Tape            |
|           | Optical disk             |
| Terlambat |                          |

# Satuan Memori Diatas Terabyte

**1024 Gigabytes = 1 Terabyte**

**1024 Terabytes = 1 Petabyte**

**1024 Petabytes = 1 Exabyte**

**1024 Exabytes = 1 Zettabyte**

**1024 Zettabytes = 1 Yottabyte**

1024 Yottabytes = 1 Brontobyte

1024 Brontobytes = 1 Geopbyte

1024 Geopbyte = 1 Saganbyte

1024 Saganbyte = 1 Pijabyte

1024 Pijabyte = 1 Alphabyte

1024 Alphabyte = 1 Kryatbyte

1024 Kryatbyte = 1 Amosbyte

◦ 1024 Amosbyte = 1 Pectrolbyte

1024 Pectrolbyte = 1 Bolgerbyte

1024 Bolgerbyte = 1 SambobYTE

1024 SambobYTE = 1 Quesabyte

1024 Quesabyte = 1 Kinsabyte

1024 Kinsabyte = 1 Rutherbyte

1024 Rutherbyte = 1 Dubnibyte

1024 Dubnibyte = 1 Seaborgbyte

1024 Seaborgbyte = 1 Bohrbyte

1024 Bohrbyte = 1 Hassiubyte

1024 Hassiubyte = 1 Meitnerbyte

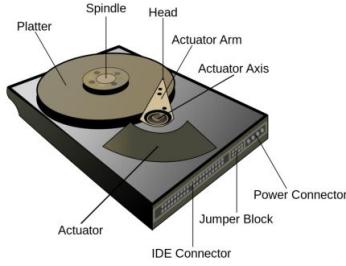
1024 Meitnerbyte = 1 Darmstadbyte

1024 Darmstadbyte = 1 Roentbyte

1024 Roentbyte = 1 Coperbyte

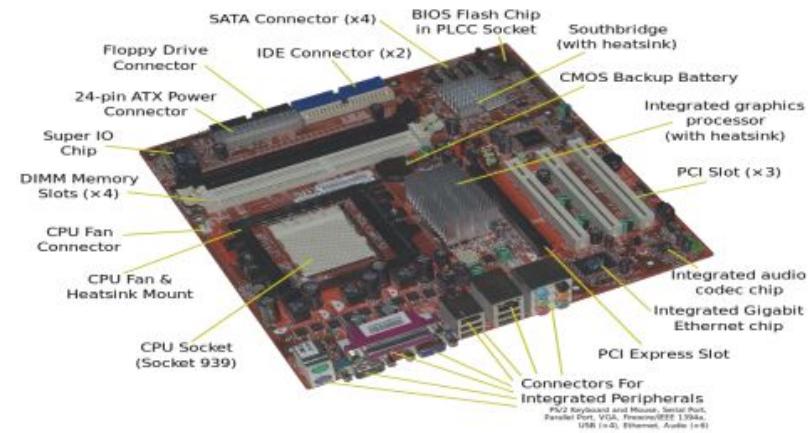
# I/O

- Perangkat masukan/keluaran digunakan sistem komputer untuk berinteraksi dengan lingkungan luar, baik ke pemakai ataupun lingkungan secara umum
- Perangkat masukan/keluaran terdiri dari dua bagian, yaitu:
  - Komponen mekanis, yaitu perangkat itu sendiri
  - Komponen elektronis, yaitu pengendali perangkat berupa chip controller

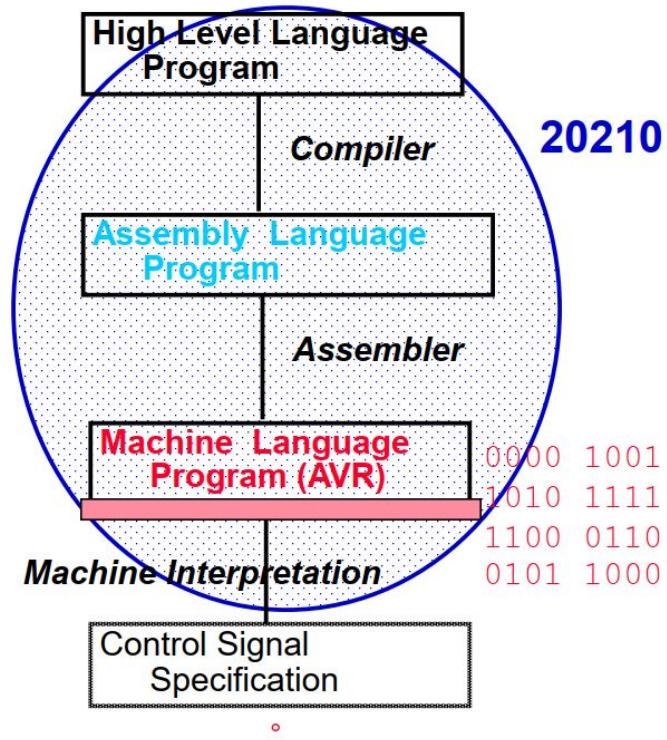


# Bus / Inter-Connection

- Disebut BUS dan interkoneksi ini berkaitan dengan tatacara hubungan antarkomponen-komponen sistem komputer.
- Bus terdiri dari tiga macam, yaitu:
  - Bus alamat (address bus)
    - Untuk memberikan alamat dari memori atau port yang hendak diakses. Bus alamat berisi 16, 20, 24 ... 64 jalur sinyal paralel atau lebih
  - Bus data (data bus)
    - Untuk membaca dan mengirim data dari/ke memori atau port. Bus data berisi 8, 16, 32 ... 64 jalur sinyal paralel atau lebih.
  - Bus kendali (control bus)
    - Sinyal bus kendali antara lain:
      - Memory Read
      - Memory Write
      - I/O read
      - I/O Write
- Sistem interkoneksi antarkomponen (Bus) yang populer antara lain: ISA, EISA, MCA, SCSI, VESA, PCI, AGP dan USB



# Tingkat-tingkat Bahasa Pemrograman



**A = 25;**

**B = 8;**

**C = A + B;**

**lds r1, 0x100**

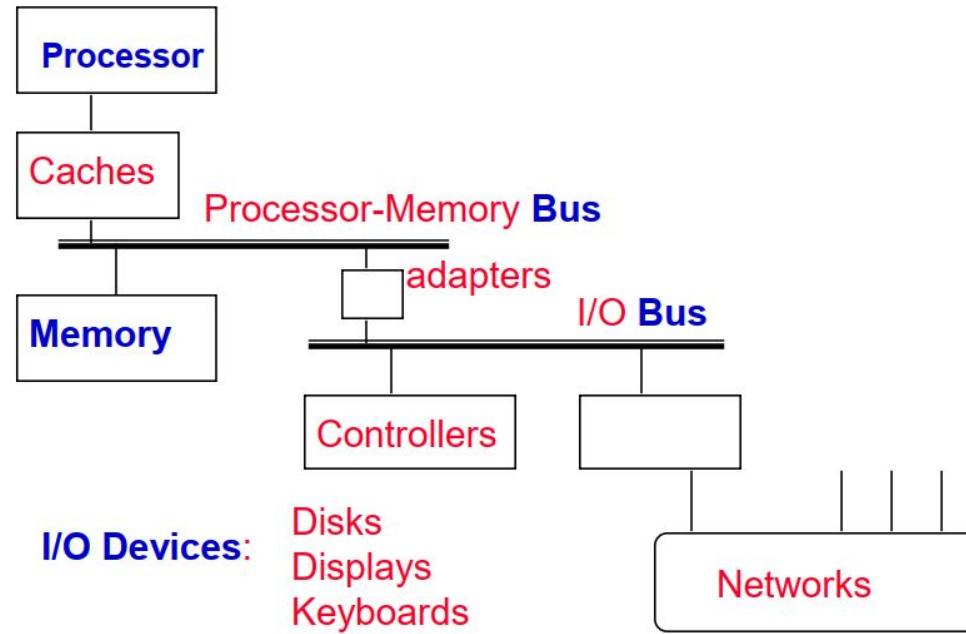
**lds r2, 0x102**

**add r1, r2**

**sts 0x104, r1**

0000 1001 1100 0110 1010 1111 0101 1000  
1010 1111 0101 1000 0000 1001 1100 0110  
1100 0110 1010 1111 0101 1000 0000 1001  
0101 1000 0000 1001 1100 0110 1010 1111

# Struktur (Umum) Interkoneksi Antar-Komponen

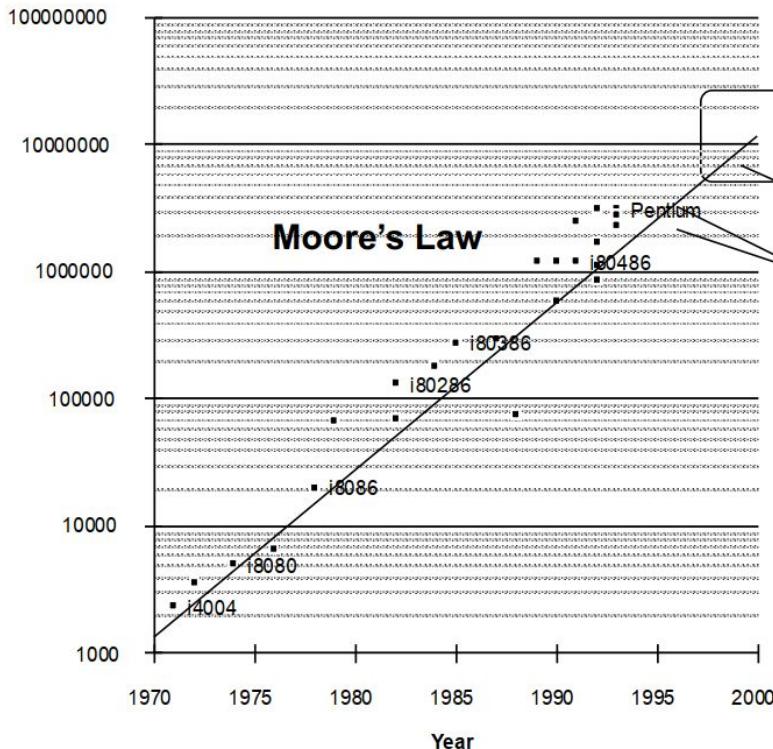


Semua komponen memiliki organisasi & antar-muka

# Hukum Moore

- Hukum Moore adalah salah satu hukum yang terkenal dalam industri mikroprosesor yang menjelaskan tingkat pertumbuhan kecepatan mikroprosesor.
- Diperkenalkan oleh Gordon E. Moore salah satu pendiri Intel. Ia mengatakan bahwa pertumbuhan kecepatan perhitungan mikroprosesor mengikuti rumusan eksponensial.
- Namun, hukum ini semakin tidak relevan. Perkembangan mikroprosesor saat ini semain tidak sesuai dengan hukum Moore.
- Justru hukum moore sekarang digunakan sebagai acuan dan tujuan yang ingin dicapai oleh para ilmuwan dalam pembuatan mikropsosessor.
- Sekarang ini, Hukum Moore menjadi pengendali laju peningkatan jumlah transistor pada keping IC
- Intel secara resmi memulai arsitektur prosesornya dengan code [Nehalem](#). Prosesor ini akan mulai menerapkan teknik teknologi nano dalam pembuatan prosesor, sehingga tidak membutuhkan waktu selama 18 bulan untuk melihat peningkatan kompleksitas tapi akan lebih singkat.

# Trend Teknologi: Kapasitas Mikroprosesor

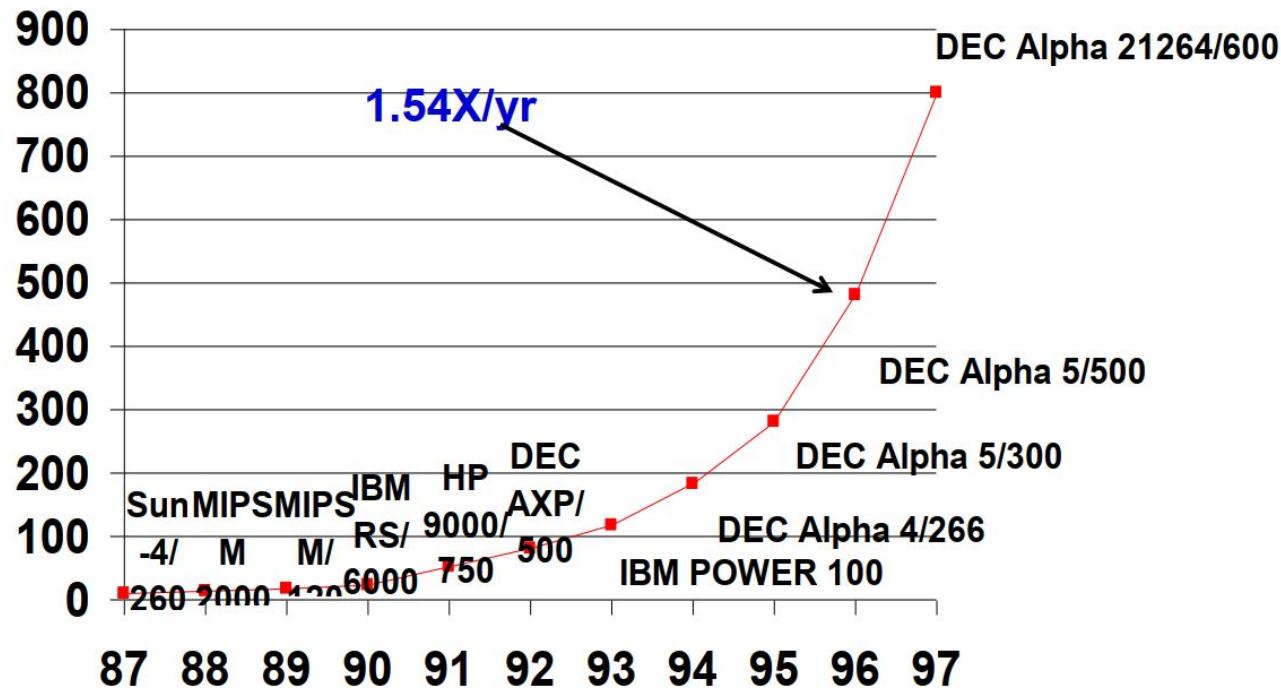


Alpha 21264: 15 million  
Pentium Pro: 5.5 million  
PowerPC 620: 6.9 million  
Alpha 21164: 9.3 million  
Sparc Ultra: 5.2 million

**2X transistors/Chip  
Every 1.5 years**

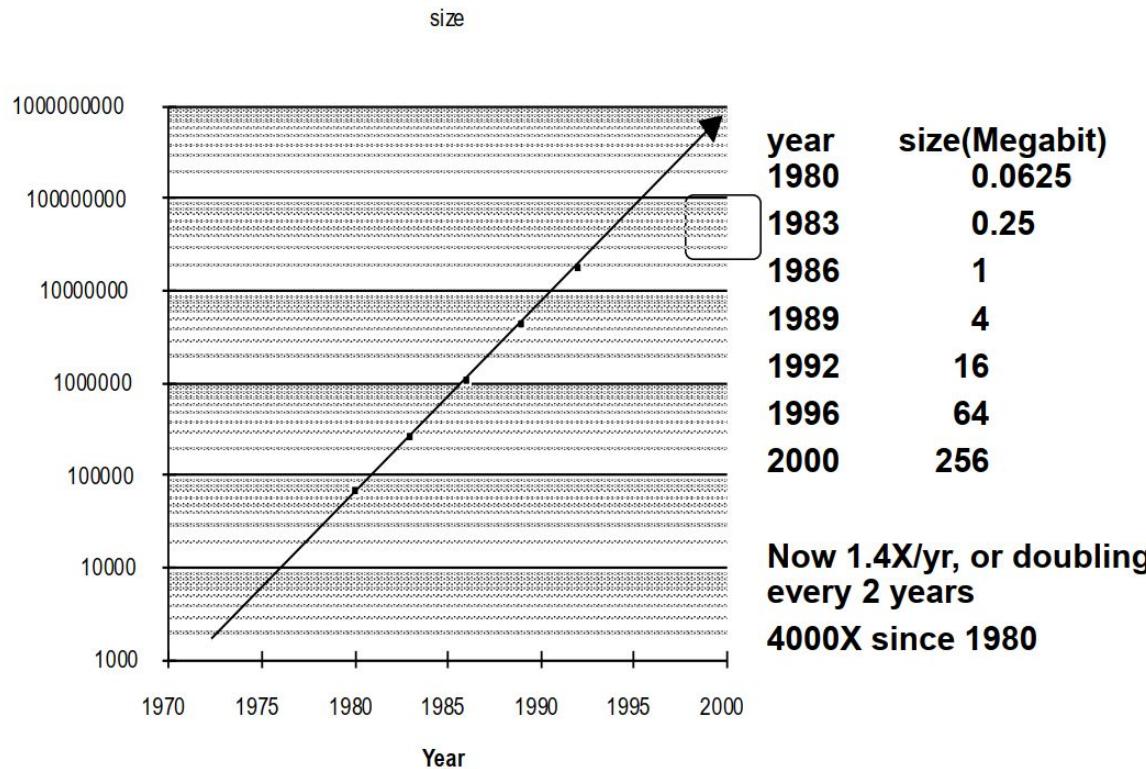
Called  
**“Moore’s Law”**

# Trend Teknologi: Kinerja Prosesor



Processor performance increase/year, mistakenly referred to as Moore's Law (transistors/chip)

# Trend Teknologi: Kapasitas Memori (1 Chip DRAM)



# Teknologi Komputer Perubahan Dramatis

## Processor

- 2X lebih cepat setiap 1,5 tahun
- 100X lebih cepat dalam dekade terakhir

## Memori

- Kapasitas DRAM: 2x / 2 years
- Kecepatan Memori: meningkat 10% per tahun
- Biaya per bit: membaik 25% per tahun
- Kapasitas meningkat 64X dalam dekade terakhir

## Disk

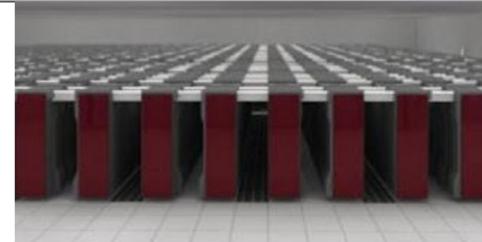
- Kapasitas disk: > 2X setiap 1,0 tahun
- Biaya per bit: membaik 100% per tahun
- Kapasitas meningkat 120X dalam dekade terakhir

# Komputer Tercepat dan Tercanggih di Dunia

1. Sequoia



2. K computer



3. Mira



4. SuperMUC



5. Tianhe 1A



6. Jaguar



# Komputer Tercepat dan Tercanggih di Dunia

## 5. Connection Machine 5

Digunakan oleh National Security Agency (NSA) pada tahun 1991 dan tetap beroperasi sampai 1997, komputer "Frostburg" pada awalnya dikirimkan dengan 256 node pemrosesan tetapi kemudian ditingkatkan pada tahun 1993 menjadi (kemudian) 512 node pemrosesan besar dan 2TB RAM.

CM-5 memiliki kinerja puncak 65,5 gigaflops \* dan lampu yang tampak futuristik di sisinya tidak hanya untuk alasan estetika. Panel cahaya akan menunjukkan pemrosesan penggunaan node dan juga digunakan untuk tujuan diagnostik.



# Komputer Tercepat dan Tercanggih di Dunia

## 4. SpiNNaker

Dibangun oleh University of Manchester, mesin SpiNNaker terdiri dari satu juta prosesor yang mampu melakukan 200 triliun tindakan per detik. Artinya dapat memodelkan lebih banyak neuron biologis secara real time daripada mesin lain yang pernah dibuat.

SpiNNaker adalah upaya nyata pertama untuk menciptakan model nyata dari otak manusia yang bekerja, dengan 1 juta core, masih jauh dari 1 miliar neuron biologis secara real time yang disediakan oleh otak manusia. Meski begitu, mesin akan dapat memberikan wawasan yang tak tertandingi tentang cara kerja otak.



# Komputer Tercepat dan Tercanggih di Dunia

## 3. The Summit

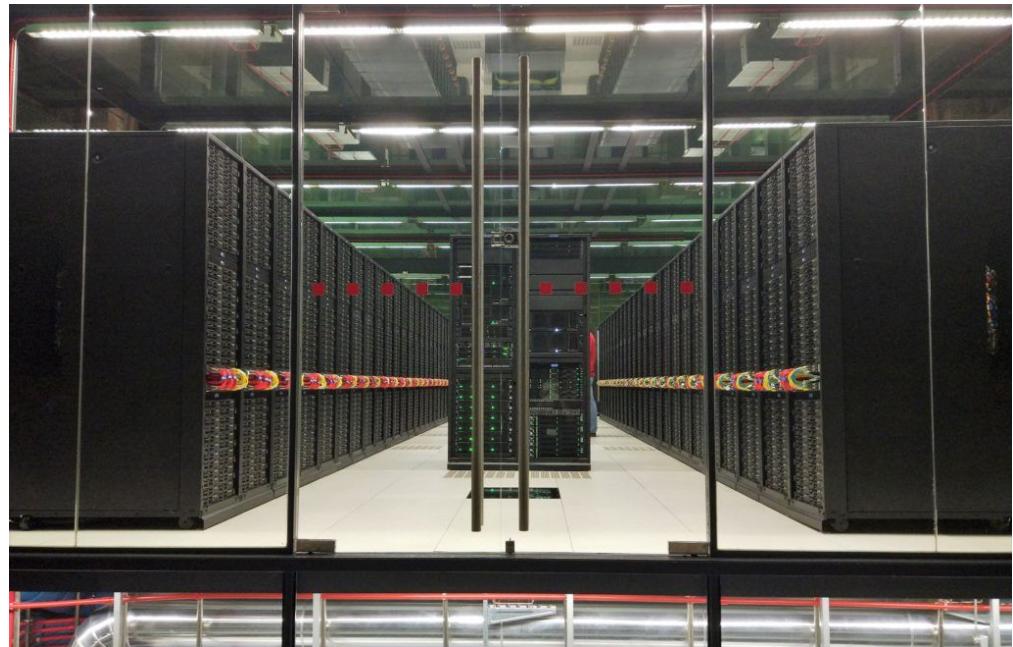
Superkomputer Summit, pada Juni 2018, adalah superkomputer paling kuat di planet ini. Itu membuatnya menjadi nomor tiga dalam daftar ini hanya karena kekuatannya yang belum pernah terjadi sebelumnya – tercepat di dunia dengan 200 exaflops \*. Tidak hanya itu yang paling kuat, itu juga superkomputer paling hemat energi kelima di sekitar.



# Komputer Tercepat dan Tercanggih di Dunia

## 2. MareNostrum 4

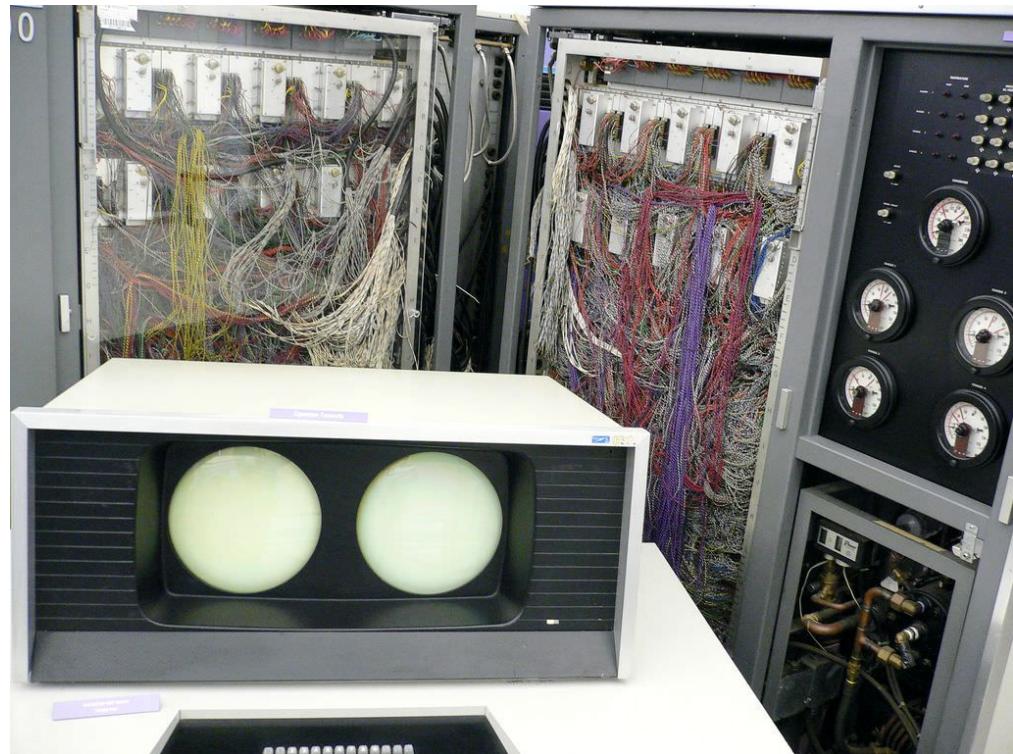
Beroperasi sejak 2017, kekuatan puncaknya adalah 11,15 petaflops, setara dengan mampu melakukan lebih dari sebelas ribu triliun operasi per detik dan sepuluh kali lebih banyak dari MareNostrum 3, yang dipasang antara 2012 dan 2013. Meskipun keuatannya sepuluh kali lebih besar daripada pendahulunya, itu hanya meningkatkan konsumsi energi hingga 30%.



# Komputer Tercepat dan Tercanggih di Dunia

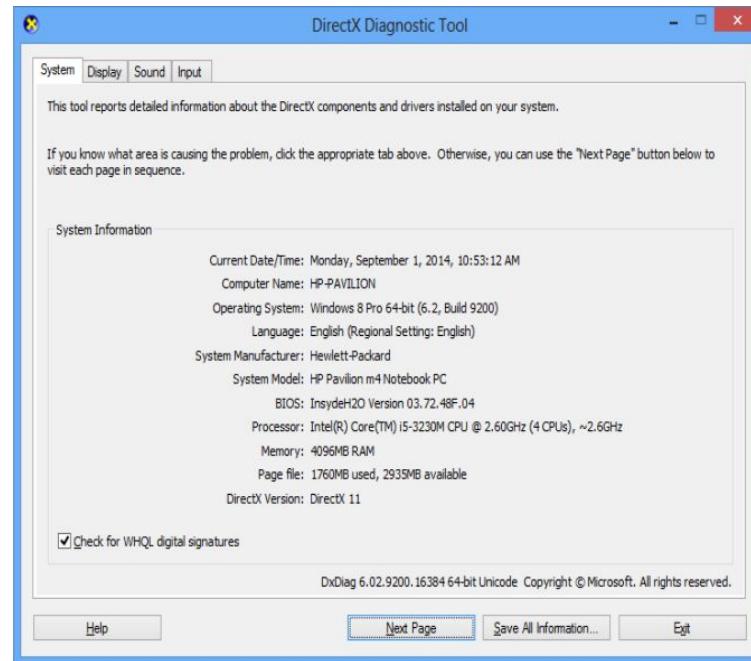
## 1. CDC 6600

CDC 6600 adalah komputer tercanggih unggulan dari 6000 seri komputer mainframe, diproduksi oleh Control Data Corporation (CDC). 6600 dianggap sebagai superkomputer pertama dan asli yang sukses, mengungguli pemegang rekor industri sebelumnya dengan kinerja hingga tiga megaflop.



# Tugas

- Kelompok 3 Orang
- Buat makalah tentang spesifikasi komputer tercanggih yang ada di pasaran Indonesia beserta. Contoh produknya (Low, middle, premium)
  - Desktop
  - Laptop
  - Tablet
- Kata Pengantar
- Daftar Isi
- I. Pendahuluan
  - Latar Belakang
  - Rumusan Masalah
  - Tujuan Penulisan
- II. Isi
- III. Penutup
  - Kesimpulan
- Daftar Pustaka



# Sistem Operasi - 2

Rudi Rosadi, S.Si., M.Kom /  
Rahmatullah Arrizal P, S.Kom, M.T

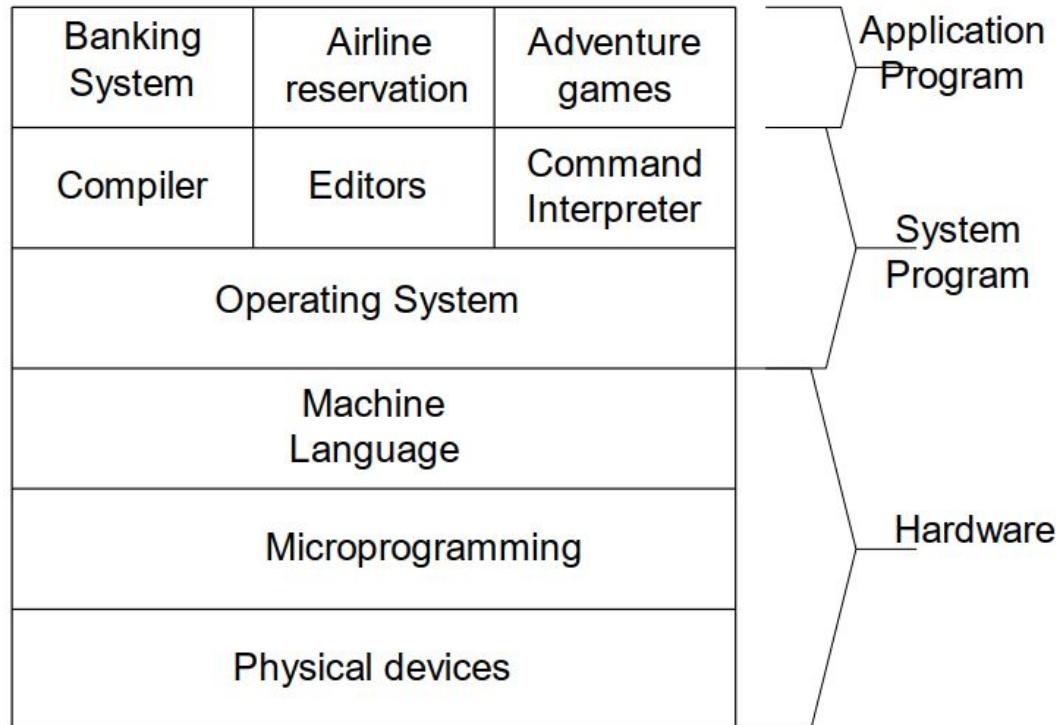
# Sistem Operasi (Operating System) (1)

- Program / perangkat lunak yang mengontrol eksekusi program-program aplikasi dan berfungsi sebagai perantara (interface) antara pemakai komputer (user) dan komputer (hardware)
- Perangkat Keras ----- S.O. ----- User

# Sistem Operasi (Operating System) (2)

- S.O. menyediakan sarana (beserta lingkungannya) sehingga pemakai dapat menjalankan program.
- **Tujuan Sistem Operasi :**
  - Nyaman : Sistem operasi akan membuat sistem komputer lebih mudah digunakan.
  - Efisiensi : Sistem Operasi memungkinkan sumber daya sistem komputer dapat digunakan dengan cara yang efisien

Sebuah sistem komputer terdiri dari hardware, program sistem dan program aplikasi, terlihat dari gambar berikut :



# Service yang disediakan SO

- Menyediakan user interface
- Menyediakan program execution
- Menyediakan I/O operations
- Menyediakan file-system manipulation
- Menyediakan system communications / networking
- Mampu melakukan error detection
- Mampu melakukan resource Sharing
- Ada fasilitas security
- Fasilitas accounting system

# Fungsi dasar Sistem Operasi (1)

## 1. S.O. sebagai ‘Extended machine’ atau ‘virtual machine’

Contoh: Untuk membaca atau menulis sesuatu ke disk, beberapa parameter harus diketahui :

misalnya :

- alamat block yang akan dibaca
- jumlah sektor per track
- mode perekaman
- jarak antar sektor
- apa yang akan dilakukan bila ada tanda terhapus

S.O. menyembunyikan semua masalah kerumitan ini dengan penamaan file yang bisa dibaca dan ditulis yang sederhana. Begitu juga dengan masalah interrupt, timers, memori manajemen, dan semua feature low level lainnya.

# Fungsi dasar Sistem Operasi (2)

## 2. S.O. sebagai ‘Resources manager’ atau pengelola sumber daya

Komputer modern terdiri dari :

prosesor, memori, timer, disk, terminal, tape drive, network interface , laser printer dll

Perangkat tsb memiliki keterbatasan kapasitas, sedangkan program yang akan menggunakannya banyak.

- Tugas O.S. mengatur agar semua peralatan komputer dapat bekerja dengan baik tanpa ada saling berebut atau berkompetisi

# SEJARAH SISTEM OPERASI (1)

## Generasi I (1945 - 1955)

- Tabung hampa (vacuum tube)
- Plugboard
- Orang yang berhasil membangun Mesin Hitung menggunakan tabung hampa: Howard Aiken , John Van Neumann , J. Presper Eckert dan William Mauchley, Konrad Zuse.
- Sangat besar dengan 20.000 tabung
- Bahasa mesin
- Tanpa S.O.

## Generasi II (1955 - 1965)

- Transistor
- Sistem batch
- Mahal
- Menggunakan punch card
- Bahasanya FORTRAN dan Assembly
- Menggunakan tape drive
- Berisi sistem operasi (FMS - Fortran Monitor System, dan IBSYS)
- Off line printing (tidak tersambung ke komputer utama)

# SEJARAH SISTEM OPERASI (2)

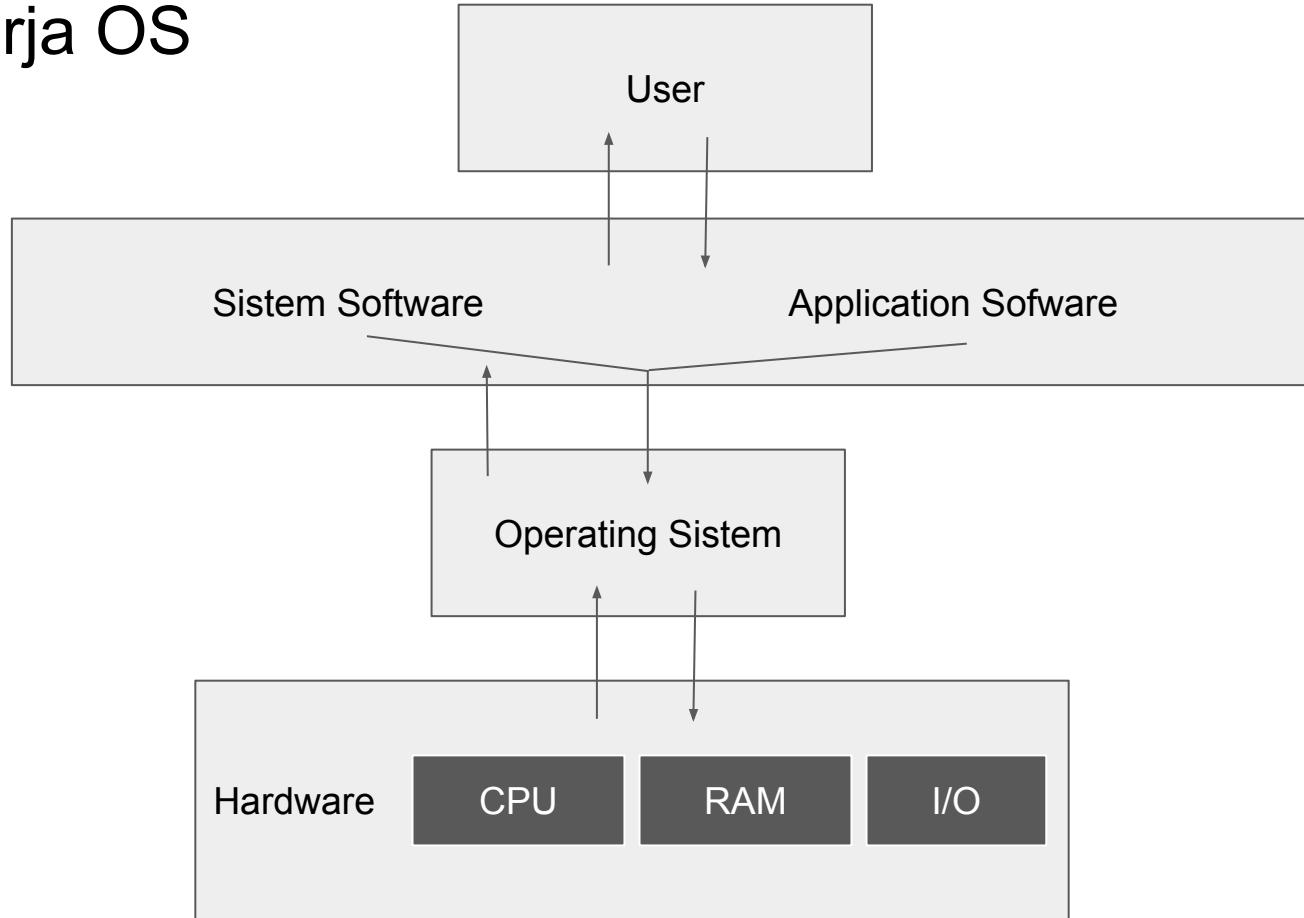
## Generasi III (1965 - 1980)

- IC (Integrated Circuit)
- multi programming
- misal : IBM System / 360
- OS / 360
- Menggunakan disk (hard disk)
- Teknik Spooling (dari : Simultaneous Peripheral Operation On line)
- MULTICS (MULTIplexed Information and Computer Service) --> PDP 1 s.d. Mini PDP 11
- Cikal bakal UNIX oleh Ken Thomson (PDP-7)

## Generasi IV (1980 – 1990 , 1990- ....)

- PC
- LSI (Large Scale Integration) --> VLSI
- MS DOS
- UNIX
- IBM PC
- CISC Intel x86, pentium, n core, xeon
- RISC Computer Mac , Power PC
- Network Operating System
- Distributed Operating System
- Uni processor, **multi processor**
- Keluarga Microsoft Windows
- Keluarga UNIX
- Keluarga Mac OS
- Client Server Computing
- SO untuk Smart Phone

# Cara Kerja OS



# Sistem Operasi pada Komputer

**Terbagi menjadi 3 kelompok besar:**

1. Keluarga **Microsoft Windows** - yang antara lain terdiri dari Windows Desktop Environment (versi 1.x hingga versi 3.x), Windows 9x (Windows 95, 98, dan Windows ME), dan Windows NT (Windows NT 3.x, Windows NT 4.0, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7 (Seven) yang dirilis pada tahun 2009, dan Windows 8)
2. Keluarga **Unix** yang menggunakan antarmuka sistem operasi POSIX, seperti SCO UNIX, keluarga BSD (Berkeley Software Distribution), GNU/Linux, MacOS/X (berbasis kernel BSD yang dimodifikasi, dan dikenal dengan nama Darwin) dan GNU/Hurd.
3. Mac OS, adalah sistem operasi untuk komputer keluaran **Apple** yang biasa disebut Mac atau Macintosh. Sistem operasi yang terbaru adalah **Mac OS X** versi 10.6 (Snow Leopard). Musim panas 2011 ada peluncuran versi 10.7 (Lion).

Sedangkan komputer **Mainframe**, dan **Super komputer** menggunakan banyak sekali sistem operasi yang berbeda-beda, umumnya merupakan turunan dari sistem operasi UNIX yang dikembangkan oleh vendor seperti IBM AIX, HP/UX, dll.

# Sistem Operasi pada Komputer (Contd)

| Icon             | Windows icon.   | Apple icon.  | Different icons depending on what software interface you are using e.g. Redhat has a redhat symbol. Ubuntu has the Circle of Friends symbol.   |
|------------------|---|--|--|
|                  |  This is the Start Menu and is located in the bottom left hand corner of the screen it allows you to access your ' <b>Control Panel</b> ', ' <b>Computer</b> ', programs, folders and more.<br><br>You can also shut down your computer using this menu. |  It is located in the top left hand corner of the screen and it is where you can access your ' <b>System Preferences</b> ', ' <b>Software Update</b> ', ' <b>About This Mac</b> ' and more.<br><br>You can also shut down your computer using this menu. |  These symbols are located in the top left hand corner of the screen and it is where you can access all your computer applications. |
| Search Tool      | Magnifying glass located in the Start Menu.   | Magnifying glass located in the top right hand corner.   | If using Ubuntu, <b>click on the Circle of Friends icon</b> .  |
| Task Bar         | Located at the bottom of the screen.<br><br>It contains shortcuts to applications, the date and time, and more.   | Located at the bottom of the screen.<br><br>It contains shortcuts to applications, files and folders. Referred to as a ' <b>Dock</b> '.  | Located on the left hand side of the screen with applications running in a vertical manner.<br><br>It contains shortcuts to applications, files and folders.   |
| Finding Programs | Start Menu.<br><br>An application where you can view and organise files and folders.  | Finder.<br><br>An application where you can view and organise files and folders.   | Dash.<br><br>The dashboard of Ubuntu where you can view all applications and files.  |
| Default Browser  | Internet Explorer.<br><br>(Now known as Microsoft Edge for Microsoft Windows 10 version and later)  | Safari.  | Firefox.   |

# Jenis Platform & OS

- Microcomputers: Unix, Windows, Linux
- Mainframe: IBM, Unix
- Supercomputer: IRIX, Linux
- Workstation, server: Linux, Windows
- Networking: Linux, Netware, Windows
- PDA: Blackberry, Linux, Palm, Windows Mobile
- Smartphone: Symbian, Linux, Windows Mobile, Android, RIM

# Berbagai versi Unix

| Nama     | Vendor                         |
|----------|--------------------------------|
| AIX      | IBM                            |
| A/UX     | Apple (Macintosh)              |
| BSD      | University of California       |
| DG/UX    | Data General                   |
| HP/UX    | Hawlette-Packard               |
| MS/UX    | NEC                            |
| PC/IX    | Interactive System Corporation |
| SCO UNIX | SCO                            |
| SINIX    | Siemens                        |
| ULTRIX   | DEC                            |
| UNICOS   | Cray Research                  |
| UNIX     | AT&T, SCO, Sun Microsystem     |
| VENIX    | VenturaCom, Inc                |
| XENIX    | SCO / Microsoft                |
| UNIXWARE | CALIFORNIA                     |

# Sistem Operasi Mobile

1. Sistem operasi Symbian (Ericsson, Nokia, Motorola, dan Psion )
  - o Dirancang khusus untuk ponsel
  - o versi 9.3 Symbian
2. Windows Mobile (Microsoft) Windows Phone
  - o Windows Mobile 6.0 Profesional, Photon
  - o kecocokan dalam hal platform serta ekstensi yang mudah dengan software komputer
  - o Mobile Blog, GPS, menonton televisi, serta Mobile Database.
  - o bersifat tertutup
3. Mobile Linux (Motorola)
  - o Didukung oleh IBM, Oracle, dan Intel
  - o aplikasi tambahan yang belum banyak beredar
4. Sistem operasi BlackBerry (Canada, Research in Motion (RIM) )
  - o Push-email
  - o GPS, internet mobile, akses Wi-Fi
5. System operasi Palm
  - o prosesor Intel PXA270, 312MHz
6. Java
  - o polyponik (jenis lama)
  - o Menggunakan bahasa Java
7. Android (Google)
  - o berbasis Linux sehingga termasuk open source
8. IOS
  - o Keluarga Iphone, Ipad dll (Apple)

# OS mobile



symbian  
os



iOS 4

Windows  
phone



RIM

BlackBerry

bada

android

Microsoft  
.NET

# OS Mobile (Contd)



# Jenis-jenis Pengoperasian Komputer

1. Single Program Operation (mono programming)
  - o Memproses satu program dalam satu waktu
  - o Micro computer
  - o Menggunakan S.O. sederhana
2. Batch Processing
  - o Sejumlah program dikumpulkan (batch) dan di-run secara kelompok (group)
  - o Running satu pada satu waktu tetapi untuk proses I/O memungkinkan overlap
  - o Program berada dalam antrian
3. Multi programming/ multi tasking
  - o Sejumlah program dapat dijalankan pada suatu waktu
  - o Resources (sumber daya) dialokasikan sesuai dengan kebutuhan.
4. Multi access
  - o Sejumlah user dapat berinteraksi, melalui terminal

Suatu Struktur Sistem Operasi Yang Ideal mempunyai komponen-komponen sbb:

- Nukleus
- Memory management
- I/O control
- Backing Store management
- Resource Allocation & Scheduling
- Protection

4 komponen utama S.O. dikategorikan sbb:

manajemen proses, manajemen memori, manajemen file dan manajemen I/O

# Nukleus

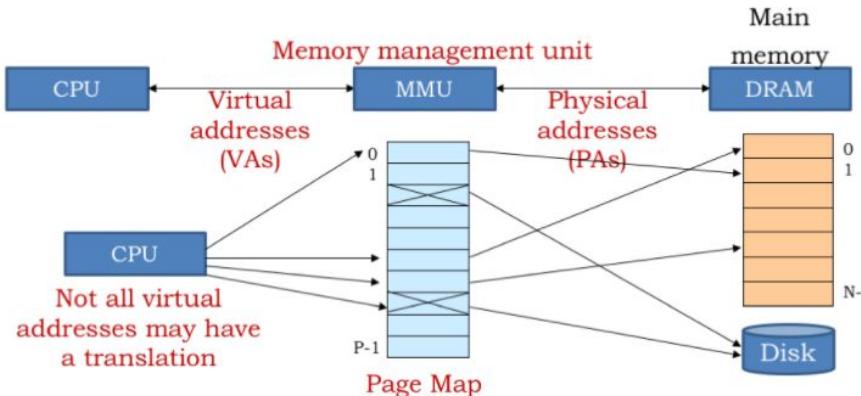
- Modul paling rendah dari suatu S.O
- ditunjang langsung ke H/W
- melayani modul-modul / layer lainnya
- contoh tugas
  - penanganan interrupt
  - lokasi waktu kerja proses
  - membantu komunikasi antar program

# Memory management

- Keterbatasan main memory
- Berfungsi untuk mengalokasikan memory bagi program yang sangat membutuhkan

contoh : virtual memori

memori komputer terlihat jauh lebih besar daripada yang sebenarnya



# Pengendalian I/O

- Problem : variasi dalam karakteristik dan kecepatan
- Berfungsi untuk menyamakan perbedaan-perbedaan tsb sehingga I/O menjadi device independent
- Menyembunyikan kompleksitas akses I/O
- Teknik SPOOL-ing
- Teknik Buffering
- Teknik Caching

# Backing Store Management

- File sistem
- Memelihara struktur semua informasi mengenai data dan program
- Menjamin efisiensi pemakaian data dan program yang disimpan dalam file
- BSM bersama dengan MM (Memory management) modul bekerjasama dalam transfer data antara main memory dan backing store
- Memelihara access privileges setiap file (share,private atau secret)

# Resource Allocation & Scheduling

- Menghindari deadlock
- Menerapkan resource allocation policy
- Menerapkan penjadwalan pemakaian processor berdasarkan kebijakan penjadwalan (scheduling policy) misalnya berdasarkan tingkat prioritas setiap program

contoh :

scheduling policy : timeslicing

# Protection

- Terhadap pemakaian yang illegal
- Terhadap keamanan file
- Terhadap main memory pada saat eksekusi program
- Pengamanan bertingkat (level of privilege)
- Nukleus (tertinggi) ---> Aplikasi (terendah)

# Sistem Operasi - 3

Rudi Rosadi, S.Si., M.Kom /  
Rahmatullah Arrizal P, S.Kom, M.T

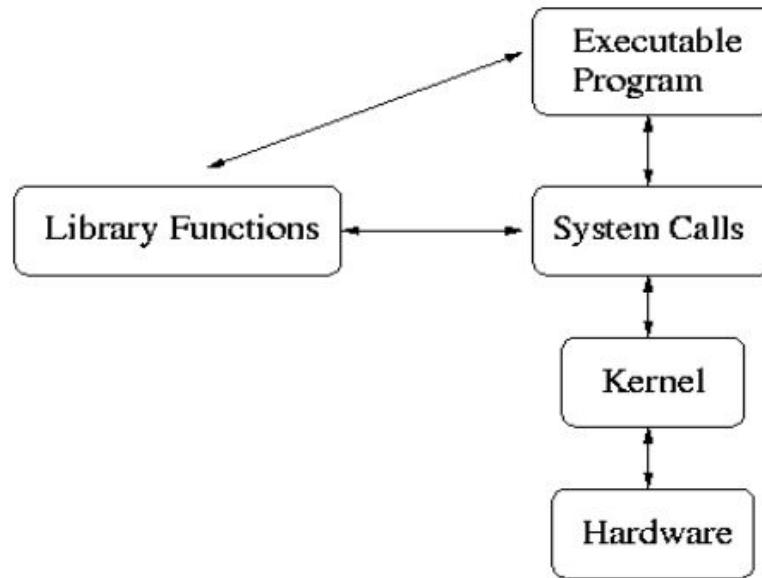
# System Call dan Struktur S.O

# System Call

- Hubungan antara S.O. dgn user program dilakukan melalui ‘extended instruction’ yg disebut dengan : system call atau API (Application Programming Interface).
- System call adalah tata cara pemanggilan di program aplikasi untuk memperoleh layanan yang disediakan oleh sistem operasi
- Fungsi system call : create,delete,use objek s/w yang diatur oleh S.O. (misal proses dan file)

# Posisi System Call

Posisi system calls itu sendiri berada di antara kernel dan executable program atau program yang sedang berjalan



# System Call

- System call merupakan sarana komunikasi antara S.O. dengan user program
- Yang bersesuaian (hubungan) untuk setiap system call adalah procedure/fungsi library yang bisa dipanggil oleh user program

Contoh:

System call READ digunakan melalui bahasa C pada UNIX dan MS DOS

count = read (file, buffer, nbytes)

\* read bukan system call tetapi prosedur di C

\* prosedur ini memanggil system call READ (bersesuaian)

# Langkah System Call

count = read (file, buffer, nbytes)

Prosedur read

-> system call READ

-> data dari file dibaca ke buffer

-> jumlah byte yang dibaca dikirim ke count, biasanya = nbytes

-> jika gagal, count akan bernilai 1

# 1. System Call untuk Control Proses

- Proses pada dasarnya merupakan program yang sedang dijalankan
- Terdiri dari:
  - executable program, data & stack, program counter, stack pointer, register dll.

Process control merupakan system calls yang mengendalikan prosesproses yang berjalan.

- load
- execute
- create process
- terminate process
- get/set process attributes
- wait for time, wait event, signal event
- allocate, free memory

# Proses pada time sharing

Misalkan pada sistem time sharing

- Secara periodik S.O. menghentikan satu proses yang sedang berjalan dan memulai proses yang lain (karena sudah kelebihan waktu CPU)
- Proses yang dihentikan tsb belum selesai (ditunda).
- Semua informasi tentang proses harus disimpan
- Tempat penyimpanan informasi proses : tabel proses

Tabel proses bisa berbentuk array atau linked list

- Kunci System call untuk manajemen proses terletak pada penciptaan dan penghentian proses (creation & termination)

# Proses pada multiprogramming

Dalam sistem multiprograming, user diberi identitas (uid)

- Setiap proses yang diciptakan selalu dikaitkan dgn uid sbg pemilik proses (owner)
- User dapat dikelompokkan ke dalam grup dan diberi identitas grup (gid)
- Uid dan gid digunakan untuk proteksi

## 2. System Call untuk File Manipulation

Kumpulan system calls yang bertugas untuk melakukan manipulasi file seperti pembacaan, penulisan, penghapusan dan pengubahan

- File : Abstraksi bagaimana informasi disimpan dan diambil kembali.
- File digunakan untuk memudahkan user (fungsi S.O)
- System call dibutuhkan untuk create,remove, read,write,open,close file
- Banyak S.O. mensupport konsep direktori untuk pengelompokan file.
- System call perlu untuk create, remove direktori ataupun file dalam direktori
- Hirarki proses dan file diorganisasi sbg tree.

# Proteksi Files

- Jika beberapa user dapat menggunakan satu komputer, perlu dipikirkan pengamanan (proteksi) file-file setiap pemakai.
- Misalnya proteksi file dan direktori pada S.O. UNIX
- Menggunakan 9 bit kode proteksi
- Kode proteksi terdiri dari 3 field @ 3 bit
  - owner, grup, everyone
- Setiap field terdiri dari rwx bit yaitu :
  - 1 bit untuk baca (read)
  - 1 bit untuk tulis (write)
  - 1 bit untuk eksekusi (execute)
- Contoh : rwx r-x –x

# File Descriptor / Handle

- Sebuah file harus dibuka sebelum dapat dioperasikan (read/write)
- Jika diizinkan (access is permitted), sistem akan menghasilkan suatu nilai (small integer) yang disebut dgn file descriptor yang akan digunakan untuk operasi berikutnya
- Jika akses gagal maka kode kesalahan akan dihasilkan.

# Abstraksi I/O dgn File khusus

- Menggunakan file khusus / special file
- Memodelkan peralatan I/O seperti file

Ada 2 jenis :

- Block Special file

digunakan untuk peralatan yang dapat diakses secara block dan acak (block stream)

misalnya : disk

- Character Special File

digunakan untuk peralatan yang mempunyai fasilitas alur karakter (character stream)

misalnya : terminal, printer (selain disk)

### 3. System Call untuk Device Manipulation

System calls yang mengatur penggunaan peralatan-peralatan yang terhubung pada mesin tersebut.

- request device, release device
- read, write, reposition
- get/set device attributes
- logically attach or detach devices

## 4. System Call untuk Information Maintenance

System calls yang menghubungkan user dengan sistem operasi dalam hal berbagi informasi.

- get/set time or date
- get/set system data
- get/set process, file, or device attributes

## 5. System Call untuk Communication

Ada dua model komunikasi yaitu pertukaran informasi dilakukan melalui fasilitas komunikasi antar proses yang disediakan oleh sistem operasi (Message-Passing) dan pertukaran dengan menggunakan memori (sharedmemory).

- create, delete communication connection
- send, receive messages
- transfer status information
- attach or detach remote devices

# Tiga cara memberikan parameter dari program ke sistem operasi:

- Melalui registers (sumber daya di CPU).
- Menyimpan parameter pada data struktur (table) di memori, dan alamat table tsb ditunjuk oleh pointer yang disimpan di register.
- Push (store) melalui “stack” pada memori dan OS mengambilnya melalui pop pada stack tsb

# Shell

- S.O menunggu instruksi dari pengguna.
- Program yang membaca instruksi dan mengartikan /menerjemahkan perintah (control statement) -> command Interpreter

dalam UNIX : Shell

dalam MS DOS : Command

- Command Interpreter, editor, compiler bukan bagian dari S.O. tetapi sangat penting dalam pemakaian S.O.

# Shell

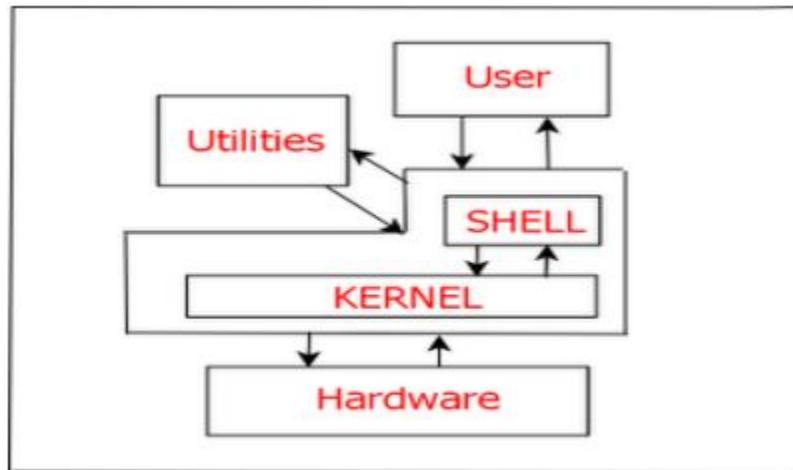
- Umumnya Shell menyediakan prompt sebagai user interface,
  - tempat dimana user mengetikkan perintah-perintah yang diinginkan baik berupa perintah internal shell (internal command), ataupun perintah eksekusi suatu file program (eksternal command),
- Shell memungkinkan user menyusun sekumpulan perintah pada sebuah atau beberapa file untuk dieksekusi sebagai program.

# Macam-maca Shell

- Bourne shell, C shell, Emacs shell, Stand alone shell, rc shell, bean shell adalah beberapa contoh shell dari UNIX.
- cmd.exe
- command.com
- Windows power shell
- Singularity shell
- Google shell
- Console
- iSeries QSHELL

# Posisi shell dalam sistem operasi

Shell digunakan sebagai jembatan antara user dengan Kernel untuk mendapatkan hasil atau output yang diinginkan.



# Redirection

- Standar output dapat diarahkan ke file  
misal : date > fileout
- Standar input dapat diarahkan dari file  
misal : sort <filein> fileout
- Output dari sebuah program dapat digunakan sebagai input bagi program yang lain dengan menghubungkan kedua program menggunakan “pipe”

Proses 1 ----- pipe ----- proses 2

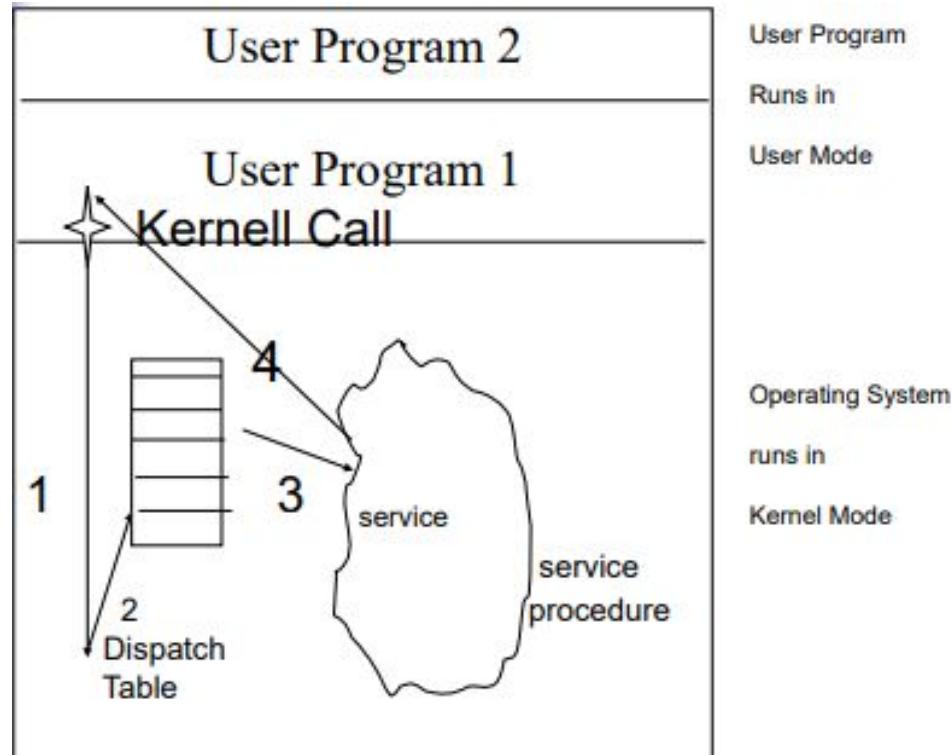
Misal : cat file1 file2 file3 | sort >fileout

# Struktur Sistem Operasi

- Sistem Monolitik
- Sistem Berlapis (Layered System)
- Sistem dengan Mesin Maya (Virtual Machines)
- Sistem dengan Client-Server
- Sistem berorientasi objek

# Struktur Sistem Monolithic

1. User program trap ke kernel
2. S.O menentukan no service
3. S.O mengalokasikan dan memanggil service procedure
4. Kontrol dikembalikan ke user program



# Sistem Berlapis (Layered System)

- Konsep : “Sistem operasi dibentuk secara hirarki berdasarkan lapisan-lapisan, dimana lapisan-lapisan memberi layanan lapisan lebih atas”.
- Contoh :
  - ◆ Sistem Operasi THE (Technique Hogesschool Endiche), Djikstra, 1968.
  - ◆ Sistem Operasi MULTICS (menggunakan concentric rings).

Tabel Lapisan-lapisan pada Sistem Operasi TH

| Layer | Function                                |
|-------|---|
| 5     | The Operator                            |
| 4     | User Programs                           |
| 3     | I/O Management                          |
| 2     | Operator-Process Communication          |
| 1     | Memory & Drum Management                |
| 0     | Processor Allocation & Multiprogramming |

# Sistem dengan Mesin Maya (Virtual Machines)

- Konsep : Awalnya struktur ini membuat seolah-olah user mempunyai seluruh komputer dengan simulasi atas pemroses yang digunakan.
- Sistem operasi melakukan simulasi mesin nyata.
- Mesin hasil simulasi digunakan user, mesin maya merupakan tiruan 100% atas mesin nyata.
- Semua user diberi ilusi mempunyai satu mesin yang sama-sama canggih.
- Contoh:
  - CP/CMS
  - VM/370
  - Virtual Machine Monitor
  - CMS (Conversational Monitor Stem)

# Sistem dengan Client-Server

- Konsep : Server adalah proses yang menyediakan layanan, dan Client adalah proses yang memerlukan / meminta layanan.
- Proses client yang memerlukan layanan mengirim pesan ke server dan menanti pesan jawaban.
- Proses server setelah melakukan tugas yang diminta, mengirim hasil dalam bentuk pesan jawaban ke proses client.
- Server hanya menanggapi permintaan client dan tidak memulai percakapan dengan client”.

# Sistem berorientasi objek

- Konsep : “Layanan diimplementasikan sebagai objek”.
- Model ini terstruktur dan memisahkan antara layanan yang disediakan dan implementasinya.
- Contoh : Sistem operasi X-kernel

# Manajemen Proses dan Interupsi

# Proses

- Proses adalah abstraksi tentang suatu program yang sedang dijalankan
- Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan managemen proses seperti:
  - Pembuatan dan penghapusan proses pengguna dan sistem proses.
  - Menunda atau melanjutkan proses.
  - Menyediakan mekanisme untuk proses sinkronisasi.
  - Menyediakan mekanisme untuk proses komunikasi.
  - Menyediakan mekanisme untuk penanganan deadlock.
- Komponen proses
  - program yang sedang dieksekusi
  - data
  - sumber daya (resources)
  - status dari eksekusi
- Semua komputer modern bisa melakukan beberapa pekerjaan dalam saat yang sama.

# MODEL PROSES

## Multiprogramming (Multitasking )

- Manajemen banyak proses pada satu pemroses
- Melakukan proses satu persatu secara bergantian dalam waktu yang sangat cepat / bersamaan (hardware level). Setiap proses mempunyai satu virtual CPU.

Program-program yang dijalankan bersifat:

- Saling tak bergantung (independen)
- Proses terpisah satu dari lainnya dan tidak saling berpengaruh.
- Satu program pada satu saat (one program at any instant)
- Pada satu waktu sesungguhnya hanya satu proses yang dilayani pemroses menggunakan interleave (saling melanjutkan/bersambung) bukan overlap diantara program-program.

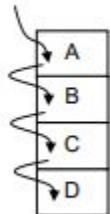
## Pseudoparallelism

Melakukan lebih dari satu pekerjaan dalam waktu yang bersamaan / pseudoparallelism (user level).

Semua S/W dikelompokkan menjadi sejumlah proses sekuensial. Menggunakan model pseudoparalelisme Suatu prosesor tunggal dapat dibagi diantara beberapa proses menggunakan suatu algoritma penjadwalan.

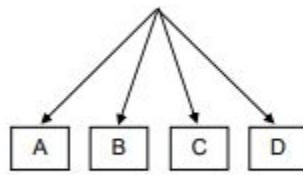
Secara konsep setiap proses mempunyai 1 virtual CPU, tetapi pada kenyataannya adalah multiprogramming. Maka akan lebih mudah menganggap kumpulan proses yang berjalan secara parallel.

Satu  
program counter



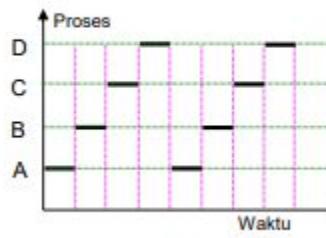
(a)

Empat  
program counter



(b)

Model  
Proses



(c)

Keterangan :

- a.multiprogramming untuk 4 program di memori
- b.model konseptual untuk 4 proses independent, sequential
- c.hanya 1 program yang aktif dalam 1 waktu = pseudoparalel

Contoh :

computer scientist (cs) sedang memasak kue (resep=program (algoritma), cs=prosesor, bahan=input data, proses = baca resep, campur bahan dan panggang kue) Anaknya menangis digigit lebah, cs berhenti dan memberi pertolongan (prioritas lebih tinggi), selesai lanjutkan dengan masak lagi

# ISTILAH LAIN

## Multiprocessing

- Manajemen banyak proses di komputer multiprocessor (banyak pemroses didalamnya)
- Dulunya sistem ini hanya terdapat disistem besar, mainframe dan mini komputer.
- Saat ini komputer workstation telah dapat dilengkapi multiprocessor.
- Sistem operasi Microsoft Windows NT, Vista, Windows 7/8, UNIX, Linux menyediakan dukungan multiprocessing.

## Distributed processing

- Manajemen banyak proses yang dieksekusi di banyak sistem komputer yang tersebar (terdistribusi)

# HIRARKI PROSES

- Ada cara untuk menghapus dan membentuk proses
- Pada UNIX , proses dibentuk dengan (system call) FORK Proses utama dan subproses yang dipanggil dapat berjalan secara paralel
- Pada MS-DOS, system call me-load suatu file biner tertentu ke memori dan mengeksekusinya sebagai subproses.
- Proses utama dan subproses tidak dapat berjalan secara paralel

# KEADAAN PROSES

Proses butuh berinteraksi dengan proses lain dan perlu diketahui kondisi dari proses lainnya itu.

Ada 3 keadaan proses :

1. Running

Kondisi dimana suatu proses sedang menggunakan CPU pada saat tersebut

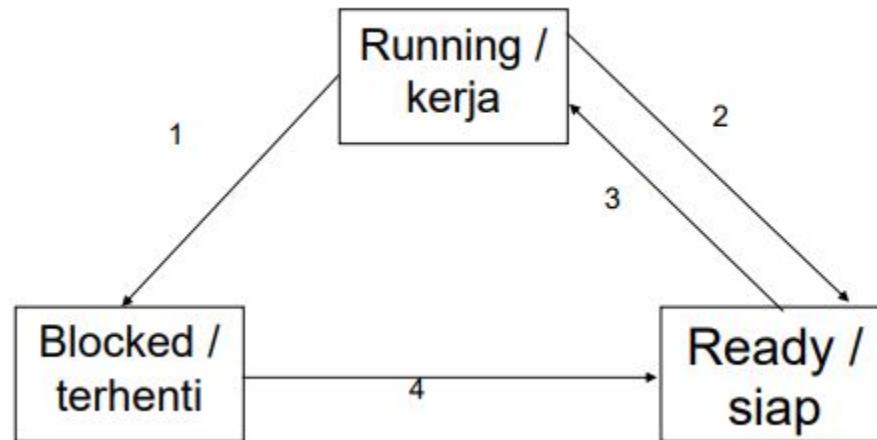
2. Ready

Kondisi dimana proses sebenarnya runnable, namun berhenti sementara karena CPU sedang digunakan proses lain

3. Blocked

Kondisi dimana proses belum dapat dijalankan karena menunggu kelengkapan proses eksternal

# 4 Transisi Status Proses



# 4 Transisi Proses

## 1. Running menjadi blocked

transisi ini terjadi karena ditemukan proses tidak dapat dilanjutkan karena ada sesuatu yang terjadi

## 2. Running menjadi ready

terjadi karena penjadwal proses memutuskan bahwa suatu proses terlalu lama menggunakan CPU

## 3. Ready menjadi running

terjadi jika semua proses telah mendapatkan bagiannya menggunakan CPU dan saatnya proses dalam kondisi ready untuk menggunakan CPU

## 4. Blocked menjadi ready

terjadi jika kelengkapan eksternal yang ditunggu proses sudah terpenuhi

# IMPLEMENTASI PROSES

- OS menggunakan tabel proses, satu entry per proses
- Tabel proses berisi :
  - informasi tentang status proses
  - program counter
  - stack pointer
  - alokasi memori
  - status-status file yang sedang dibuka
  - informasi akunting dan penjadwalan
  - segala sesuatu yang perlu disimpan pada saat berpindah dari kondisi running ke ready
- Isi tabel proses secara umum memuat : manajemen proses, manajemen memori, manajemen file.

# Contoh Tabel Proses :

| Process management        | Memory management       | File management        |
|---------------------------|-------------------------|------------------------|
| Register                  | Pointer to text segment | UMASK mask             |
| Program counter           | Pointer to data segment | Root directory         |
| Program status word       | Pointer to bss segment  | Working directory      |
| Stack pointer             | Exit status             | File descriptors       |
| Process state             | Signal status           | Effective uid          |
| Time when process started | Process id              | Effective gid          |
| CPU time used             | Parent process          | System call parameters |
| Children's CPU time       | Process group           | Various flag bits      |
| Time of next alarm        | Real uid                |                        |
| Message queue pointers    | Effective uid           |                        |
| Pending signal bits       | Real gid                |                        |
| Process id                | Effective gid           |                        |
| Various flag bits         | Bit maps for signals    |                        |
|                           | Various flag bits       |                        |

# Interupsi

- **Interupsi** : Kerja prosesor pada suatu proses terhenti oleh pensaklaran konteks.
- **Pensaklaran konteks** : perubahan kegiatan prosesor dari proses ke proses yang terjadi diantara proses sistem / proses aplikasi
- **Konteks** : kegiatan prosesor terhadap sesuatu hal, berasal dari sistem operasi, sistem bahasa dan sistem utilitas.
- **Blok kendali proses** : suatu bagian memori untuk mencatat keadaan proses, yang terbagi atas wilayah dimana setiap wilayah untuk mencatat informasi yang berbeda.

## 2 Cara Interupsi pada Processor

- **Interupsi langsung**

Berasal dari luar prosesor (peripheral / alat mengirim sinyal kepada prosesor untuk meminta pelayanan)

- **Interupsi Tanya / Polling**

Berasal dari prosesor (prosesor secara bergiliran mengecek apakah ada peripheral yang memerlukan pelayanan atau tidak)

Interupsi dapat di-enable dan disable tergantung pada levelnya.

# Pembangkit interupsi dapat berasal dari :(1)

- **Program**, di dalam program telah dirancang pada bagian tertentu akan terjadi pensaklaran konteks, yang menimbulkan interupsi, contohnya pada saat penggunaan alat / prosesor secara bergantian.
- **Prosesor**, prosesor sendiri dapat membangkitkan interupsi, yang biasa mengolah logika dan aritmatika. Jika melampaui ukuran tampung register di dalam prosesor, maka terjadi kekeliruan yang akan menginterupsi kerjanya sendiri dan menyerahkan kendali prosesor pada sistem operasi. Misalnya pembagian dengan bilangan nol.
- **Satuan kendali**, tugas untuk melaksanakan interupsi terletak pada satuan kendali, sehingga satuan kendali dapat membangkitkan interupsi. Misalnya kekeliruan instruksi
- **Kunci waktu / clock**, menggunakan interupsi berkala. Misalnya pada program looping yang tak terhingga, diinterupsi pada setiap selang waktu 60 detik.

## Pembangkit interupsi dapat berasal dari :(2)

- **Peripheral I/O**, I/O jika akan bekerja memberitahukan pada prosesor dengan interupsi prosesor dan juga ketika pekerjaan selesai atau pada saat terjadi kekeliruan paritas.
- **Memori**, karena terjadi kekeliruan, misalnya ketika prosesor ingin mencapai alamat memori yang terletak di luar bentangan alamat memori yang ada.
- **Sumber daya lain**, misal dibangkitkan oleh operator sistem komputer yang mengerti cara interupsi.

# Tindak lanjut interupsi : (1)

- **Penata interupsi / interrupt handler**

jika terjadi interupsi, maka kendali prosesor diserahkan ke bagian penata interupsi pada sistem operasi, maka penata interupsi inilah yang melaksanakan interupsi.

- Instruksi yang sedang diolah oleh prosesor dibiarkan sampai selesai program.
- Penata interupsi merekam semua informasi proses ke dalam blok kendali proses.
- Penata interupsi mengidentifikasi jenis dan asal interupsi.
- Penata interupsi mengambil tindakan sesuai dengan yang dimaksud interupsi.
- Penata interupsi mempersiapkan segala sesuatu untuk pelanjutan proses yang diinterupsi.

# Tindak lanjut interupsi : (2)

- **Penata keliru / error handler**

yaitu interupsi karena kekeliruan pada pengolahan proses dan bagian pada sistem operasi yang menata kegiatan akibat kekeliruan adalah penata keliru.

- **Pemulihan**, komputer telah dilengkapi dengan sandi penemuan dan pemulihan kekeliruan, contohnya telah dilengkapi dengan sandi Hamming sehingga ketika menemukan kekeliruan sandi akan mengoreksi kekeliruan itu, proses pulih ke bentuk semula sebelum terjadi kekeliruan.
- **Pengulangan**, mengatur agar proses yang membangkitkan interupsi keliru dikerjakan ulang, jika kekeliruan dapat diatasi maka proses akan berlangsung seperti biasa, jika tidak teratasi maka interupsi akan menempuh tindak lanjut keluar dari proses.
- **Keluar dari proses**, penata keliru menyiapkan tampilan berita keliru dari monitor, setelah itu prosesor keluar dari proses, ini adalah tindakan terakhir jika tidak dapat menolong proses yang keliru tersebut.

# Sistem Operasi - 5

Rudi Rosadi, S.Si., M.Kom /  
Rahmatullah Arrizal P, S.Kom, M.T

# Penjadwalan Proses

- Penjadwalan merupakan kumpulan kebijaksanaan dan **mekanisme** di sistem operasi yang berkaitan dengan **urutan kerja** yang dilakukan sistem komputer.
- Proses Apa? Kapan dan Berapa lama berlangsung?
- Jika banyak proses yang akan dijalankan maka SO harus memutuskan proses mana yang dijalankan terlebih dahulu
- Bagian SO yang menjadi pembuat keputusan disebut dgn : Penjadwal (Scheduler)
- Algoritmanya : Scheduling Algorithm
- Sasaran utama penjadwalan proses adalah optimasi kinerja menurut kriteria tertentu

# Konsep Dasar Penjadwalan

- Sistem multiprogramming/multitasking, terjadi beberapa proses berjalan dalam suatu waktu.
- Konsep dasar multiprogramming : suatu proses akan menggunakan CPU sampai proses tersebut dalam status wait (misalnya meminta I/O) atau selesai
- Pada saat wait, maka CPU akan menganggur, u/ mengatasinya CPU dialihkan ke proses lain, demikian seterusnya.
- Proses dijalankan :
  - Terjadi siklus eksekusi CPU dan wait I/O yang disebut CPU-I/O burst.
- Eksekusi proses dimulai dengan CPU burst dan dilanjutkan dengan I/O burst, diikuti CPU burst lain, kemudian I/O burst lain.

# Kriteria Penjadwalan

## 1. Fairness (adil) :

- proses-proses diperlakukan sama, yaitu mendapat jatah waktu pemrosesan yang sama dan tidak ada proses yang tak kebagian layanan pemrosesan sehingga mengalami kekurangan waktu.
- Meyakinkan bahwa tiap-tiap proses akan mendapatkan pembagian waktu penggunaan CPU secara terbuka

## 2. Response time (waktu tanggap) :

- Waktu yang dibutuhkan CPU untuk menanggapi proses yang masuk
- Sistem Interaktif: Adalah waktu yang dihabiskan dari saat karakter terakhir dari perintah dimasukkan atau transaksi sampai hasil pertama muncul di layar (terminal)
- Sistem Waktu Nyata (Realtime) : Simulasi yang menyamai proses yang sebenarnya di dunia nyata, terdapat batasan waktu
- Sasaran meminimumkan waktu tanggap

## 3. Waiting Time :

- Waktu yang diperlukan oleh suatu proses untuk menunggu di ready queue

# Kriteria Penjadwalan (Cont'd)

## 4. Turnaround time :

- Banyaknya waktu yang diperlukan untuk mengeksekusi proses, dari mulai menunggu untuk meminta tempat di memori utama, menunggu di ready queue, eksekusi oleh CPU, dan mengerjakan I/O.
- Turn around time = waktu eksekusi + waktu menunggu
- Sasaran penjadwalan adalah meminimumkan turn around time.

## 5. Throughput :

- Adalah jumlah kerja atau jumlah job yang dapat diselesaikan dalam satu unit waktu.
- Sasaran memaksimumkan jumlah pekerjaan

## 6. Efisiensi :

- Menjaga agar CPU senantiasa sibuk sepanjang waktu
- Utilitas CPU dinyatakan dalam bentuk % yaitu 0-100%. Namun dalam kenyataannya hanya berkisar antara 40-90%

# Penjadwalan CPU

- Pada saat CPU menganggur, SO menyeleksi proses-proses yang ada di memori utama (ready queue) untuk dieksekusi dan mengalokasikan CPU untuk salah satu proses tersebut, seleksi ini disebut short-term scheduler (CPU scheduler)
- Keputusan untuk menjadwalkan CPU mengikuti 4 keadaan :
  1. Apabila proses berpindah dari keadaan running ke waiting;
  2. Apabila proses berpindah dari keadaan running ke ready;
  3. Apabila proses berpindah dari keadaan waiting ke ready;
  4. Apabila proses berhenti.

# Strategi Penjadwalan

- Apabila model penjadwalan yang dipilih menggunakan keadaan 1 dan 4, maka penjadwalan semacam ini disebut non-preemptive.
  - Apabila yang digunakan adalah keadaan 2 dan 3, maka disebut dengan preemptive
- 
- Preemptive Scheduling :
    - Strategi yang memungkinkan sebuah proses yang dijalankan ditunda sementara
    - Contoh : Round Robin, Prioritas
  - Non Preemptive Scheduling
    - Tidak bisa ditunda
    - Contoh : Shortest Job First, FCFS

# First-Come First-Served Scheduling (FCFS)

- Proses yang pertama kali meminta jatah waktu untuk menggunakan CPU akan dilayani terlebih dahulu.
- Proses yang meminta CPU pertama kali akan dialokasikan ke CPU pertama kali

# Contoh

| Process | Burst Time |
|---------|------------|
| P1      | 24         |
| P2      | 3          |
| P3      | 3          |

Gant Chart dengan penjadwalan FCFS adalah sebagai berikut :



- Waktu tunggu untuk **P1** adalah 0, **P2** adalah 24 dan **P3** adalah 27 sehingga rata-rata waktu tunggu adalah  $(0 + 24 + 27)/3 = 17$  milidetik.
- Sedangkan apabila proses datang dengan urutan **P2**, **P3**, dan **P1**, hasil penjadwalan CPU dapat dilihat pada gant chart berikut :



# Job Terpendek Pertama (Shortest Job First)

- Digunakan untuk sistem batch
- Run time diketahui lebih dahulu
- Mengetahuinya dengan teknik estimasi
- Pada penjadwalan SJF, proses yang memiliki CPU burst paling kecil dilayani terlebih dahulu
- Terdapat dua skema :
  1. Non preemptive, bila CPU diberikan pada proses, maka tidak bisa ditunda sampai CPU burst selesai.
  2. Preemptive, jika proses baru datang dengan panjang CPU burst lebih pendek dari sisa waktu proses yang saat itu sedang dieksekusi, proses ini ditunda dan diganti dengan proses baru. Skema ini disebut dengan Shortest-Remaining-Time-First (SRTF).

- Meskipun algoritma ini optimal, namun pada kenyataannya sulit untuk diimplementasikan karena sulit untuk mengetahui panjang CPU burst berikutnya.
- Namun nilai ini dapat diprediksi. CPU burst berikutnya biasanya diprediksi sebagai suatu rata-rata eksponensial yang ditentukan dari CPU burst sebelumnya atau “Exponential Average”

$$\tau_{n+1} = \alpha t_0 + (1 - \alpha) \tau_n$$

- $\tau_{n+1}$  = Panjang CPU burst yang diperkirakan
- $\tau_n$  = Panjang CPU burst sebelumnya
- $t_n$  = Panjang CPU burst yang ke-n (yang sedang berlangsung)
- $\alpha$  = Ukuran Pembanding antara  $\tau_{n+1}$  dengan  $\tau_n$  (0 sampai 1)

Sebagai contoh, jika  $\alpha = 0,5$ , dan:

|                      |    |   |   |   |    |    |    |     |     |
|----------------------|----|---|---|---|----|----|----|-----|-----|
| CPU burst ( $t_n$ ): | 6  | 4 | 6 | 4 | 13 | 13 | 13 | ... |     |
| $\tau_n$ :           | 10 | 8 | 6 | 6 | 5  | 9  | 11 | 12  | ... |

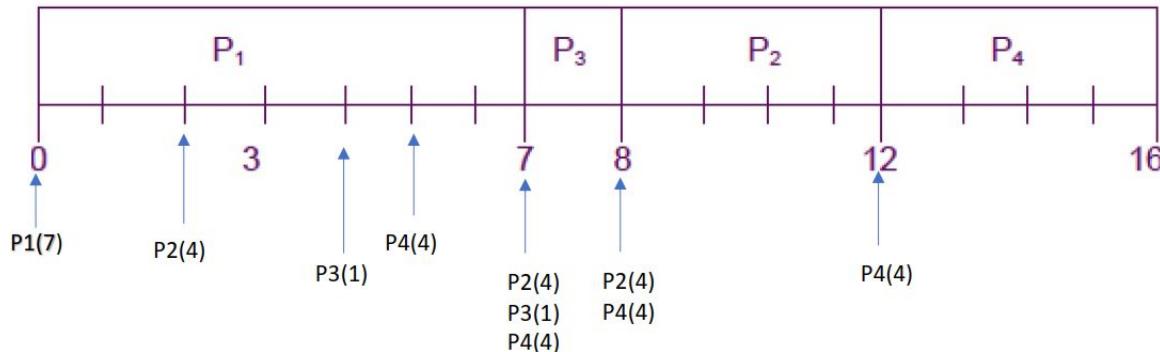
Pada awalnya  $t_1 = 6$  dan  $\tau_1 = 10$ , sehingga :

$$\tau_2 = 0,5 * 6 + (1 - 0,5) * 10 = 8$$

$$\tau_3 = 0,5 * 4 + (1 - 0,5) * 8 = 6$$

# Contoh Non-Preemptive

| Process | Arrival Time | Burst Time |
|---------|--------------|------------|
| P1      | 0.0          | 7          |
| P2      | 2.0          | 4          |
| P3      | 4.0          | 1          |
| P4      | 5.0          | 4          |

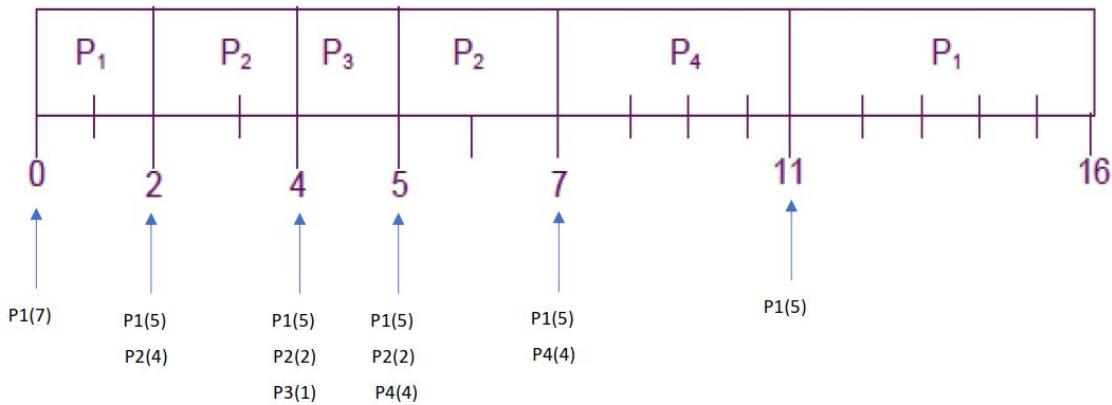


$$\begin{aligned}WT : P1 &= 0 \\P2 &= 8 - 2 = 6 \\P3 &= 7 - 4 = 3 \\P4 &= 12 - 5 = 7\end{aligned}$$

$$\begin{aligned}AWT &= (P1 + P2 + P3 + P4) / 4 \\&= (0 + 6 + 3 + 7) / 4 = 4 \text{ ms}\end{aligned}$$

# Contoh Preemptive

| Process | Arrival Time | Burst Time |
|---------|--------------|------------|
| P1      | 0.0          | 7          |
| P2      | 2.0          | 4          |
| P3      | 4.0          | 1          |
| P4      | 5.0          | 4          |



$$\begin{aligned}WT : P1 &= 0 + (11 - 2) = 9 \\P2 &= (2 - 2) + (5 - 4) = 1 \\P3 &= 0 \\P4 &= (7 - 5) = 2\end{aligned}$$

$$\begin{aligned}AWT &= (P1 + P2 + P3 + P4) / 4 \\&= (9 + 1 + 0 + 2) / 4 = 3 \text{ ms}\end{aligned}$$

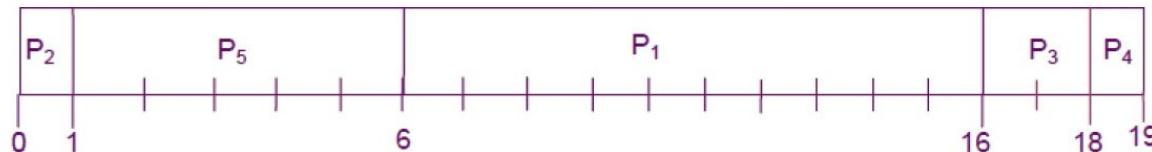
# Penjadwalan Prioritas

- Setiap proses diberi nomor prioritas (integer)
- CPU dialokasikan untuk proses yang memiliki prioritas paling tinggi (nilai integer terkecil biasanya merupakan prioritas terbesar)
- Jika beberapa proses memiliki prioritas yang sama, maka akan digunakan algoritma FCFS.
- Untuk mencegah proses dengan prioritas tinggi terus menerus maka prioritas proses akan diturunkan setiap interupsi sampai ada prioritas lain yang lebih tinggi.

## Contoh

- Misalnya terdapat lima proses **P1, P2, P3, P4** dan **P5** yang datang secara berurutan dengan **CPU burst** dalam milidetik

| Process | Burst time | Priority |
|---------|------------|----------|
| P1      | 10         | 3        |
| P2      | 1          | 1        |
| P3      | 2          | 3        |
| P4      | 1          | 4        |
| P5      | 5          | 2        |



Waktu tunggu untuk **P1** adalah 6, **P2** adalah 0, **P3** adalah 16, **P4** adalah 18 dan **P5** adalah 1 sehingga rata-rata waktu tunggu adalah  $(6 + 0 + 16 + 18 + 1)/5 = 8.2$  milidetik

# Soal

- Terdapat 5 job (P1, P2, P3, P4, P5) yang datang hampir pada saat yang bersamaan. Estimasi waktu eksekusi (burst time) masing-masing 10, 6, 12, 9 dan 5 detik, dengan arrival time (0.0, 3.0, 5.0, 7.0, 8.0)

| Process | Arrival Time | Burst Time |
|---------|--------------|------------|
| P1      | 0.0          | 10         |
| P2      | 3.0          | 6          |
| P3      | 5.0          | 12         |
| P4      | 7.0          | 9          |
| P5      | 8.0          | 5          |

Buatlah Gant Chart dan tentukan rata-rata waktu tunggu algoritma penjadwalan

- FCFS
- SJF (Non Preemptive)
- SJFP (Preemptive)

# Sistem Operasi - 6

Rudi Rosadi, S.Si., M.Kom /  
Rahmatullah Arrizal P, S.Kom, M.T

# Konsep Dasar Round Robin

- Menggunakan time sharing
- Hampir sama dengan FCFS, tapi bersifat preemptive
- Tiap proses dibatasi waktu prosesnya (quantum time)
- Keunggulan :
  - Keseragaman waktu
- Kelemahan :
  - quantum time sangat besar → switching yang terjadi akan semakin sedikit (seperti FCFS)
  - quantum time terlalu kecil → switching yang terjadi akan semakin banyak, sehingga banyak waktu yang terbuang

## Ketentuan Algoritma Round Robin:

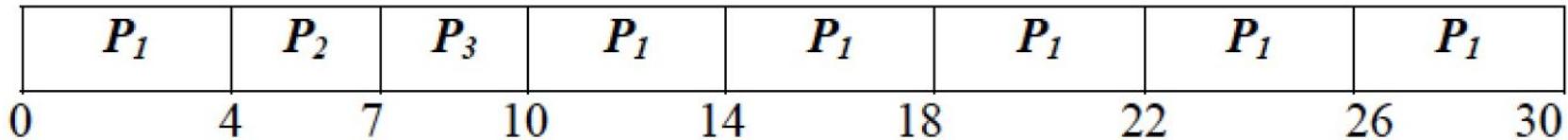
- Jika proses memiliki CPU Burst < Quantum Time, maka proses akan melepaskan CPU, jika telah selesai digunakan → CPU dapat segera digunakan oleh proses selanjutnya
- Jika proses memiliki CPU Burst > Quantum Time, maka proses tersebut akan dihentikan jika sudah mencapai quantum time dan selanjutnya mengantri kembali pada posisi tail queue (ekor dari ready queue), CPU kemudian menjalankan proses berikutnya
- Jika quantum time belum habis dan proses menunggu suatu kejadian (selesainya operasi I/O), maka proses menjadi blocked dan CPU dialihkan ke proses lain

## Contoh

Misalkan ada 3 proses: P1, P2, dan P3 yang meminta pelayanan CPU dengan quantum-time sebesar 4 milidetik

| <u>Process</u> | <u>Burst Time</u> |
|----------------|-------------------|
| $P_1$          | 24                |
| $P_2$          | 3                 |
| $P_3$          | 3                 |

- Penjadwalan proses dengan algoritma round robin dapat dilihat pada gant chart berikut



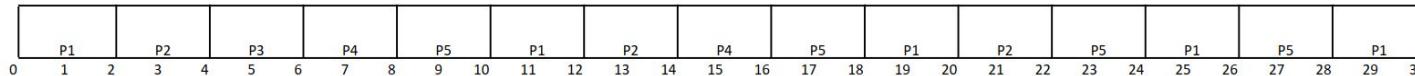
- Waktu tunggu untuk P1 adalah 6, P2 adalah 4, dan P3 adalah 7 sehingga rata-rata waktu tunggu adalah  $(6 + 4 + 7)/3 = 5.66$  milidetik

# Waktu turnaround (TA)

- Turnaround time (TA) : Banyaknya waktu yang diperlukan untuk mengeksekusi proses, dari mulai menunggu untuk meminta tempat di memori utama, menunggu di ready queue, eksekusi oleh CPU, dan mengerjakan I/O.
- $TA = \text{Waktu Tunggu} + \text{Lama Eksekusi}$
- Waktu turnaround juga tergantung ukuran waktu quantum.
- Rata-rata waktu turnaround tidak meningkat bila waktu quantum dinaikkan
- Secara umum, rata-rata waktu turnaround dapat ditingkatkan jika banyak proses menyelesaikan CPU burst berikutnya sebagai satu waktu quantum.
  - Sebagai contoh, terdapat tiga proses masing-masing 10 unit waktu dan waktu quantum 1 unit waktu, rata-rata waktu turnaround adalah 29. Jika waktu quantum 10, sebaliknya, rata-rata waktu turnaround turun menjadi 20.

# Soal Latihan

1. Terdapat 5 job yang datang hampir pada saat yang bersamaan. Estimasi waktu eksekusi (burst time) masing-masing 10, 6, 2, 4 dan 8 detik. Tentukan rata-rata waktu turnaround (ATAT= Average Turn Around Time) untuk penjadwalan CPU dengan menggunakan algoritma Round Robin (quantum time = 2)



Waiting Time(WT) :

$$P1=0+(10-2)+(18-12)+(24-20)+(28-26)=20$$

$$P2=2+(12-4)+(20-14)=16$$

$$P3=4$$

$$P4=6+(14-8)=12$$

$$P5=8+(16-10)+(22-18)+(26-24)=20$$

Turn Around Time(TAT)=

Waktu tunggu + Lama Eksekusi :

$$P1=20+10=30$$

$$P2=16+6=22$$

$$P3=4+2=6$$

$$P4=12+4=16$$

$$P5=20+8=28$$

$$AWT=(23+16+4+12+20)/5=15$$

$$ATAT=(30+22+6+26+28)/5=20,4$$

2. Diketahui proses berikut :

| <i>Proses</i> | <i>Arrival Time</i> | <i>Burst Time</i> |
|---------------|---------------------|-------------------|
| P1            | 0.0                 | 8                 |
| P2            | 0.4                 | 4                 |
| P3            | 1.0                 | 1                 |

- Tentukan rata-rata waktu tunggu dan rata-rata waktu turnaround dengan algoritma penjadwalan Round Robin dengan quantum time = 1

3. Diketahui proses berikut :

| Process Id | Arrival time | Burst time |
|------------|--------------|------------|
| P1         | 0            | 5          |
| P2         | 1            | 3          |
| P3         | 2            | 1          |
| P4         | 3            | 2          |
| P5         | 4            | 3          |

Tentukan rata-rata waktu tunggu dan rata-rata waktu turnaround dengan algoritma penjadwalan Round Robin dengan quantum time = 2

#### 4. Diketahui proses berikut :

| Process Id | Arrival time | Burst time |
|------------|--------------|------------|
| P1         | 0            | 4          |
| P2         | 1            | 5          |
| P3         | 2            | 2          |
| P4         | 3            | 1          |
| P5         | 4            | 6          |
| P6         | 6            | 3          |

Tentukan rata-rata waktu tunggu dan rata-rata waktu turnaround dengan algoritma penjadwalan Round Robin dengan quantum time = 2

# Sistem Operasi - 7

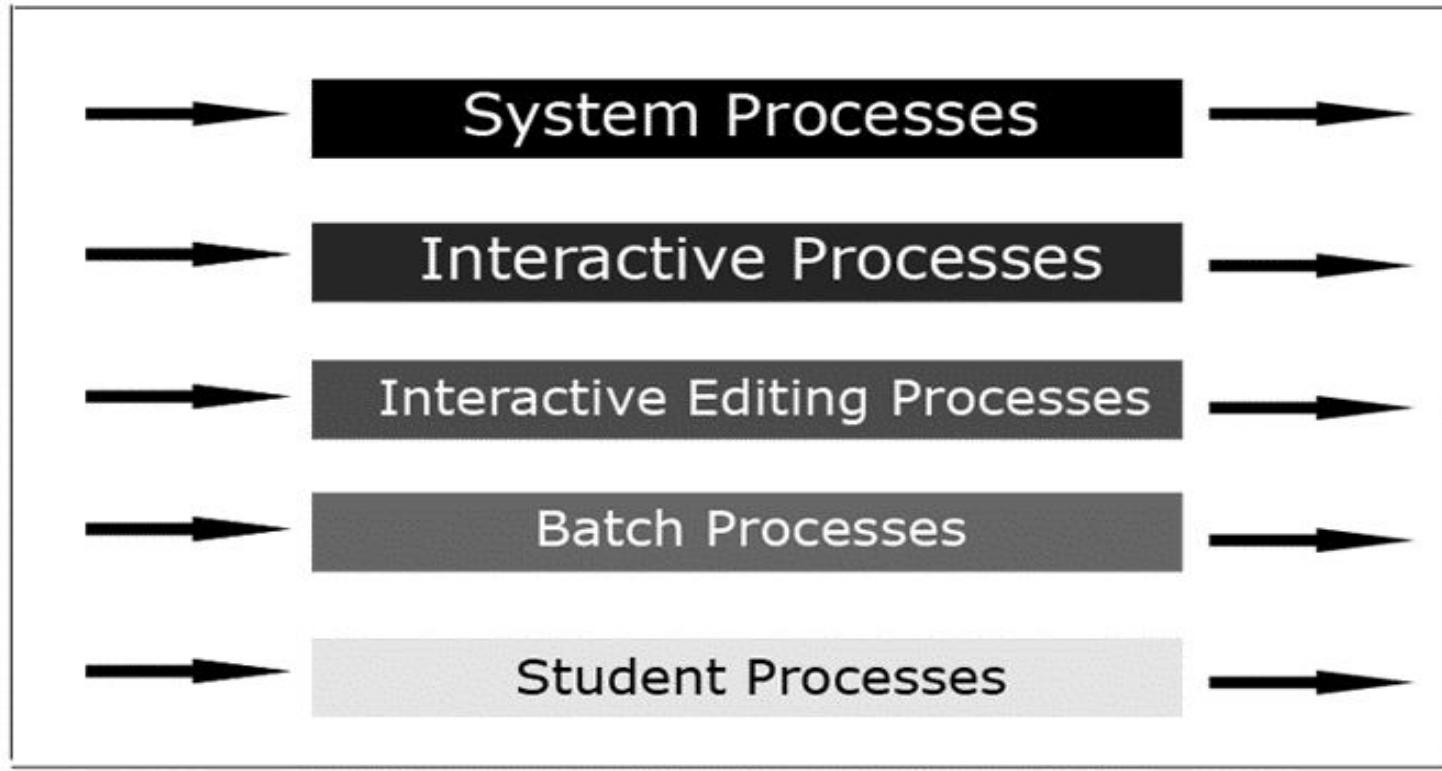
Rudi Rosadi, S.Si., M.Kom /  
Rahmatullah Arrizal P, S.Kom, M.T

# Algoritma Penjadwalan

Multilevel Queue dll

# Multilevel Queue Scheduling

- Membagi beberapa proses dalam kelompok-kelompok
- Tiap kelompok mempunyai antrian, setiap antrian memiliki algoritma penjadwalan masing masing
- Menggunakan metode prioritas tertentu
- Foreground proses memiliki prioritas yang lebih tinggi dibanding background proses



# Contoh, menggunakan algoritma FCFS, dengan prioritas

| Process Name | Arrival Time | Execute Time | Type       |
|--------------|--------------|--------------|------------|
| P0           | 0            | 5            | Foreground |
| P1           | 1            | 8            | Background |
| P2           | 3            | 7            | Background |
| P3           | 4            | 3            | Foreground |
| P4           | 5            | 3            | Foreground |
| P5           | 8            | 11           | Background |
| P6           | 15           | 3            | Foreground |
| P7           | 25           | 4            | Foreground |

### Execute Time   Non Pre-emptive Process Scheduling

|    |  |  |
|----|--|--|
| 0  | F<br><br>B<br>       | P0 arrives to queue 'F' and gets processed   |
| 1  | F<br><br>B<br>     | P1 arrives to queue 'B'  |
| 3  | F<br><br>B<br>     | P2 arrives to queue 'B'  |
| 4  | F<br><br>B<br>     | P3 arrives to queue 'F'  |
| 5  | F<br><br>B<br>     | P0 gets completed. So P3 is processed. P4 arrives to queue 'F'                             |
| 8  | F<br><br>B<br>     | P3 gets completed. So P4 is processed. P5 arrives to queue 'B'                             |
| 11 | F<br><br>B<br>     | P4 gets completed. All process in queue 'F' is completed. So P1 in queue 'B' is processed  |
| 15 | F<br><br>B<br>     | P6 arrives to queue 'F'. Since it is a non pre-emptive scheduling, P1 continues to execute |
| 19 | F<br><br>B<br>     | P1 gets completed. Since P6 is in queue 'F', P6 is processed                               |
| 22 | F<br><br>B<br>     | P6 gets completed. Since queue 'F' is empty, P2 in queue 'B' is processed                  |
| 25 | F<br><br>B<br>     | P7 arrives to queue 'F'. Since it is a non pre-emptive scheduling, P2 continues to execute |
| 29 | F<br><br>B<br>   | P2 gets completed. Since P7 is in queue 'F', P7 is processed                               |
| 33 | F<br><br>B<br> | P7 gets completed. Since queue 'F' is empty, P5 in queue 'B' is processed                  |

### Execute Time   Pre-emptive Process Scheduling

|    |  |   |
|----|--|---|
| 0  | F<br><br>B<br>       | P0 arrives to queue 'F' and gets processed  |
| 1  | F<br><br>B<br>     | P1 arrives to queue 'B'   |
| 3  | F<br><br>B<br>     | P2 arrives to queue 'B'   |
| 4  | F<br><br>B<br>     | P3 arrives to queue 'F'   |
| 5  | F<br><br>B<br>     | P0 gets completed. So P3 is processed. P4 arrives to queue 'F'  |
| 8  | F<br><br>B<br>     | P3 gets completed. So P4 is processed. P5 arrives to queue 'B'  |
| 11 | F<br><br>B<br>     | P4 gets completed. All process in queue 'F' is completed. So P1 in queue 'B' is processed                 |
| 15 | F<br><br>B<br>     | P6 arrives to queue 'F'. Since it is a pre-emptive scheduling, P1 is pre-empted and P6 is processed       |
| 18 | F<br><br>B<br>     | P6 gets completed. All process in queue 'F' is completed. So P1 in queue 'B' is processed for remaining 4 |
| 22 | F<br><br>B<br>     | P1 gets completed. Since queue 'F' is empty, P2 in queue 'B' is processed                                 |
| 25 | F<br><br>B<br>     | P7 arrives to queue 'F'. Since it is a pre-emptive scheduling, P2 is pre-empted and P7 is processed       |
| 29 | F<br><br>B<br>   | P7 gets completed. All process in queue 'F' is completed. So P2 in queue 'B' is processed for remaining 4 |
| 33 | F<br><br>B<br> | P2 gets completed. Since queue 'F' is empty, P5 in queue 'B' is processed                                 |

# *Multilevel Feedback Queue Scheduling*

- Mirip dengan multilevel queue, perbedaannya ialah algoritma ini mengijinkan proses untuk pindah antrian
- Jika proses akan menggunakan CPU dalam waktu yang cukup lama, maka proses akan berpindah ke antrian yang prioritasnya lebih rendah
- Sebaliknya, jika proses terlalu lama menunggu pada prioritas lebih rendah, maka proses akan berpindah pada ke antrian yang prioritasnya lebih tinggi

**High Priority**

**Queue 1**



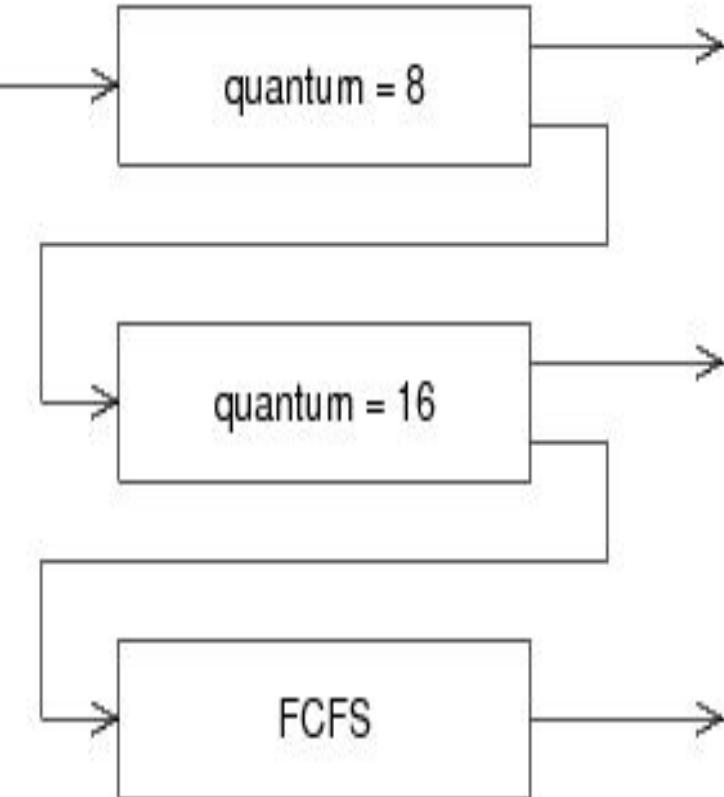
**Queue 2**



**Queue 3**



**Low Priority**



1. Semua proses yang baru datang akan diletakkan pada queue 0 ( *quantum*= 8 ms).
2. Jika suatu proses tidak dapat diselesaikan dalam 8 ms, maka proses tersebut akan dihentikan dan dipindahkan ke *queue 1* ( *quantum*= 16 ms).
3. *Queue 1* hanya akan dikerjakan jika tidak ada lagi proses di *queue 0*, dan jika suatu proses di *queue 1* tidak selesai dalam 16 ms, maka proses tersebut akan dipindahkan ke *queue 2*.
4. *Queue 2* akan dikerjakan bila *queue 0* dan *1* kosong, dan akan berjalan dengan algoritma FCFS.

# Parameter-parameter yang dibutuhkan

- Jumlah Queue/antrian
- Algoritma penjadwalan untuk tiap queue
- Metode yang diperlukan untuk menentukan kapan menaikan proses antrian ke antrian dengan prioritas lebih tinggi
- Metode yang diperlukan untuk menentukan kapan menurunkan proses antrian ke antrian dengan prioritas lebih rendah
- Metode yang diperlukan untuk menentukan letak antrian, jika ada suatu proses yang dilayani

# Guaranteed Scheduling

- Jika ada  $N$  user, maka tiap-tiap user akan dijanjikan mendapatkan  $1/N$  dari daya pemroses CPU.
- Untuk mewujudkannya, sistem harus mengetahui berapa CPU time yang diperlukan oleh setiap proses dalam satu user, dan juga CPU time yang diperlukan oleh tiap-tiap user

# Contoh

| User | CPU Time | CPU Actual | Rasio |
|------|----------|------------|-------|
| A    | 5        | 3          | 3/4   |
| B    | 4        | 6          | 6/4   |
| C    | 8        | 2          | 2/4   |
| D    | 1        | 1          | 1/4   |
| E    | 2        | 1          | 1/4   |

Ket : Jumlah CPU total time=20 ms, tiap user = 4 ms

# Implementasi Penjadwalan Prosesor

Procedure A()

```
{  
    while(true)  
    {  
        <hitung A1>;          p11  
        write(x);            p12  
        <hitung A2>;          p13  
        read(y);             p14  
    }  
}
```

Procedure B()

```
{  
    while(true)  
    {  
        read(x);           p21  
        <hitung B1>;      p22  
        write(y);          p23  
        <hitung B2>;      p24  
    }  
}
```

Diketahui x dan y adalah variable global (shared variable)

Dikerjakan serial (1 prosesor) atau parallel (2 Prosesor)

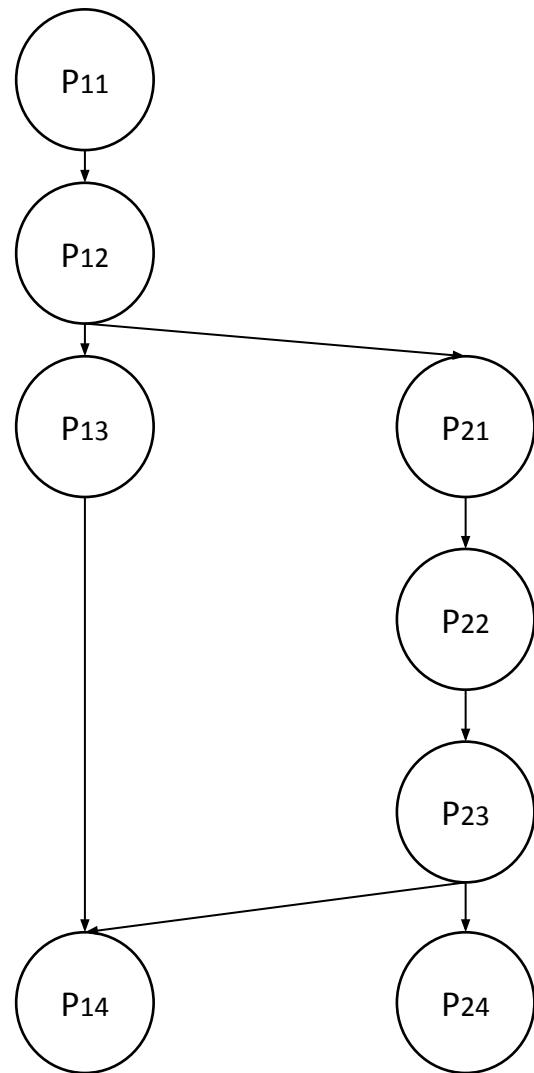
Masalah : Prosedur mana yang terlebih dahulu dikerjakan ?

Solusi dengan Precedence Graph/ *procedure of Processes*

# Precedence graph

$P = \{ p_i \mid 1 \leq i \leq n \}$       -> himpunan proses  
 $<. = \{ (p_i, p_j) \mid 1 \leq i, j \leq n \}$       -> pasangan berurutan pada p  
 $\pi = ( P, <. )$       -> komputasi

- $(p_i, p_j)$ :  $p_j$  tidak dapat dikerjakan sebelum  $p_i$  selesai
- Tidak boleh ada looping



Komputasi pada Precedence Graph

```
 $\pi = [$ 
  (p11, p12, p13, p14, p21,
  p22, p23, p24),
  { (p11, p12), (p12, p13), (p12, p21), (p21,
  p22), (p22, p23), (p23, p14), (p13, p14),
  (p23, p24) }
 $]$ 
```

# Operasi Multiprosessor

- FORK (LabelX), membangun proses baru (mengaktifkan prosessor lain) dan mengerjakannya mulai dari statement yang diberi label X
- JOIN(Cacah), menggabungkan kembali beberapa proses menjadi proses tunggal. Variabel cacah menunjukkan berapa kali JOIN dikerjakan. Setiap kali join dikerjakan secara otomatis nilai cacah berkurang satu
- QUIT( ), keluar dari ekseskusi

# Contoh

Procedure A&B

```
while (true)
{
    cacah = 1;
    < hitung p11 >;
    < hitung p12 >;
    FORK ( L1 );
    < hitung p13 >;
L2: JOIN (cacah);
    < hitung p14>;
    QUIT( );
L1: < hitung p21 >;
    < hitung p22 >;
    < hitung p23 >;
    FORK ( L3);
    GOTO (L2);
L3: < hitung p24 >;
    QUIT ( );
}
```

# Latihan Soal No.1

- Diketahui :

| Proses | Arrival Time | Burst Time |
|--------|--------------|------------|
| P1     | 0            | 6          |
| P2     | 1            | 8          |
| P3     | 2            | 10         |
| P4     | 3            | 5          |

- Tentukan rata-rata waktu tunggu untuk :
  - FCFS
  - SJF- Preemptive
  - Round-Robin ( $Q=2\text{ms}$ )

# Latihan Soal No 2

```
Procedure XX ( )
{ while (true)
{
    x = 10;          //p1
    y = 3;          //p2
    a = y * 2;      //p3
    b = 2 * x - 25; // p4
    c = 3 * a + x; // p5
    d = 2 * c - a; // p6
    e = b - 2 * d; // p7
    z = e + 1;      // p8
}
}
```

- Buatlah precedence graph dari Prosedur XX
- Buatlah urutan proses dengan menggunakan FORK-JOIN-QUIT dari prosedur XX