

# Sistem Operasi Waktu Nyata

Sistem Waktu Nyata – 10<sup>th</sup> Meeting

# Pengertian Sistem Operasi

- Perangkat lunak (kumpulan program) yang bertugas mengendalikan perangkat keras secara langsung, sebagai jembatan antara program aplikasi dan perangkat keras sistem komputer.
- Tujuan :
  - Mempermudah penggunaan sistem komputer terutama untuk pemrogram (programmer)
  - Memberikan layanan-layanan (service) bagi program aplikasi untuk memanfaatkan sumber daya (resource) komputer.
  - Mengusahakan agar resource sistem komputer digunakan secara efisien.

# Tipe Sistem Operasi

- Stand-alone : sistem operasi yang digunakan pada komputer desktop atau notebook.
- Network : sistem operasi yang memberikan dukungan ekstensif untuk jaringan komputer, umumnya berfungsi sebagai sistem operasi server.
- Embedded : sistem operasi untuk piranti tertanam dan komputer kecil, tersimpan dalam chip ROM.



# Subsistem suatu SO

- Manajemen proses
  - Tugas dari manajemen proses antara lain meliputi : kendali proses tingkat rendah, penjadwalan dan pengiriman (*dispatching*) proses ke CPU, sinkronisasi dan komunikasi antar proses, dan manajemen kemacetan (*deadlock*)
- Manajemen memori
- Sistem file
- Sistem komunikasi/jaringan
- Sistem pengamanan



# Subsistem suatu SO

- **Program** adalah istilah umum untuk suatu unit komputasi dan digunakan dalam konteks pemrograman.
- **Proses** adalah unit yang dapat dieksekusi secara bebas (independently) yang ditangani oleh sistem operasi.
- **Thread** merupakan aktivitas kecil di mana beberapa thread dapat dijalankan (run) secara parallel, tanpa terjadi beban tambahan (overhead) yang berarti.
- **Task** adalah istilah umum untuk unit komputasi yang terjadwal secara bebas dan digunakan dalam konteks penjadwalan komputasi pada prosesor.
- **Task** dapat berupa proses atau *thread*.



# Subsistem suatu SO

- Operasi—operasi kendali proses tingkat rendah
  1. Penciptaan proses baru (*new*)
  2. Penghancuran proses (*terminated*)
  3. Penggandaan dua proses menjadi proses yang serupa
  4. Penggabungan (*joint*) dua proses.
- Penjadwalan proses berfungsi agar beberapa (lebih dari satu proses) dapat memperoleh jatah waktu prosesor (CPU) atau dieksekusi oleh prosesor.
- Pengirim (*dispatcher*) bertugas untuk pemindahan keadaan (state) proses-proses.

# Keadaan suatu proses

- **New** : Sistem operasi mengetahui keberadaan task, tetapi task belum dialokasikan suatu prioritas dan suatu konteks dan belum dimasukkan ke dalam daftar task terjadwal.
- **Ready** : Atribut dari task dan sumber daya yang diperlukan untuk mengeksekusi task tersebut harus sudah tersedia agar task dapat ditempatkan dalam keadaan siap (ready).
- **Running** : task yang sedang mengendalikan prosessor. Task yang memiliki prioritas paling tinggi yang akan berada pada keadaan dieksekusi.
- **Waiting** : Eksekusi task yang berada dalam keadaan ini ditangguhkan/ditunda dikarenakan task memerlukan beberapa sumber daya yang belum tersedia.
- **Terminated** : sistem operasi belum memerlukan keberadaan task ini, meskipun mungkin tinggal dalam memori computer.

# Teknik Sinkronisasi dan Pertukaran Data

- Teknik sinkronisasi
  - Signal
  - Semaphore dan monitor
- Sarana pertukaran data
  - Pipa (pipe)
  - Fifo (named pipe)
  - Memori Bersama (shared memory)
  - Pelewatan pesan (messaging passing)
  - Panggilan prosedur jarak jauh (RPC-Remote Procedure Call)



# Deadlock

- Eksekusi proses-proses dapat menyebabkan kemacetan (deadlock).
- Teknik penyelesaian untuk mengatasi deadlock
  - ✓ Pencegahan *deadlock*
  - ✓ Penghindaran *deadlock*
  - ✓ Deteksi *deadlock*
  - ✓ Pemulihan *deadlock*
- Pada awalnya 3 jenis implementasi sistem operasi :
  - ✓ Program tunggal (*monoprogramming*)
  - ✓ Program banyak (*multiprogramming batch*)
  - ✓ Program bagi-waktu (*time-sharing*)
- Sistem operasi mutakhir berkembang menjadi :
  - ✓ Tersebar (*distributed*)
  - ✓ Multiprosesing (*multitasking*)
  - ✓ Waktu-nyata (*real-time*)
  - ✓ Multimedia
  - ✓ Basis data



# *Real Time Operating System (RTOS)*

- RTOS adalah sistem (komputasi) dimana kebenaran (*correctness*) komputasi tergantung tidak hanya pada kebenaran hasil logika komputasi, tetapi juga tergantung pada waktu kapan hasil tersebut diserahkan.
- Kategori sistem waktu-nyata :
  - **Hard RTS** : sistem dinyatakan gagal jika waktu komputasi tidak memenuhi batas waktu (*deadline*).
  - **Soft RTS** : hasil komputasi yang terlambat dari batas waktu (*deadline*) hanya akan mengurangi kinerja dari sistem tetapi tidak menjadikan gagal.



# Kebutuhan (*Requirement*) RTOS

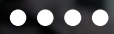
- Harus *multithreaded* dan dihalangi (*preemptive*)
- Harus mendukung prioritas dari suatu *task* dan *thread*
- Harus ada inheritance prioritas, adalah suatu mekanisme untuk menjamin bahwa task dengan prioritas lebih rendah tidak dapat menghalangi eksekusi task dengan prioritas lebih tinggi.
- Harus mendukung berbagai jenis mekanisme sinkronisasi task/thread
- Tanggapan dapat diramal
- Tanggapan berlangsung cepat.

# Ciri-ciri RTOS

- Perubahan suasana (*context switch*) yang cepat
- Berukuran kecil, dengan fungsional minimal yang dikehendaki
- Mampu menanggapi selaan luar (*external interrupt*) dengan cepat
- Multitasking dengan *Inter Process Communication* (IPC) seperti bendera semaphore, signal dan event
- Menggunakan file sekuen khusus dengan akumulasi data secara cepat
- Penjadwalan dihalangi (*preemptive*) berdasarkan prioritas
- Meminimalkan selang selama selaan (*interrupt*) dinonaktifkan.
- Terdapat primitif untuk *pause/resume* terhadap task selama waktu yang pasti.
- Alarm-alarm dan *time-out* khusus.



# THANK YOU



Nurfiana, S.Kom., M.Kom



(0721) 787214



nurfiana@darmajaya.ac.id



<https://www.darmajaya.ac.id/>