

THREAD

Mochammad Zen Samsono Hadi, ST. MSc. Ph.D

Materi

- Pengenalan Thread
- Pemrograman Multicore
- Model Multithread
- Thread Library
- Implicit Threading
- Permasalahan dalam Thread
- Thread dalam Berbagai Sistem Operasi

Tujuan

- Memperkenalkan **konsep thread** – unit dasar dari penggunaan CPU yang terbentuk dari sistem multithread
- Mengeksplor beberapa strategi implicit threading
- Membahas masalah pada program multithread

PENGENALAN THREAD

Thread

- Unit dasar dari pemanfaatan CPU
- Pada dasarnya **satu proses** terdiri atas **satu thread (single-thread)**
 - Dalam sekali waktu **hanya** ada **satu tugas** yang bisa dikerjakan dalam **satu proses**
- Padahal program-program terkini butuh menjalankan **beberapa tugas** dalam sekali waktu.
 - Butuh **multi-thread** dalam **satu proses**.

Contoh Kebutuhan Multithreading

- **Web browser**

- Thread untuk menampilkan halaman web, gambar
- Thread untuk mengunduh data dari jaringan

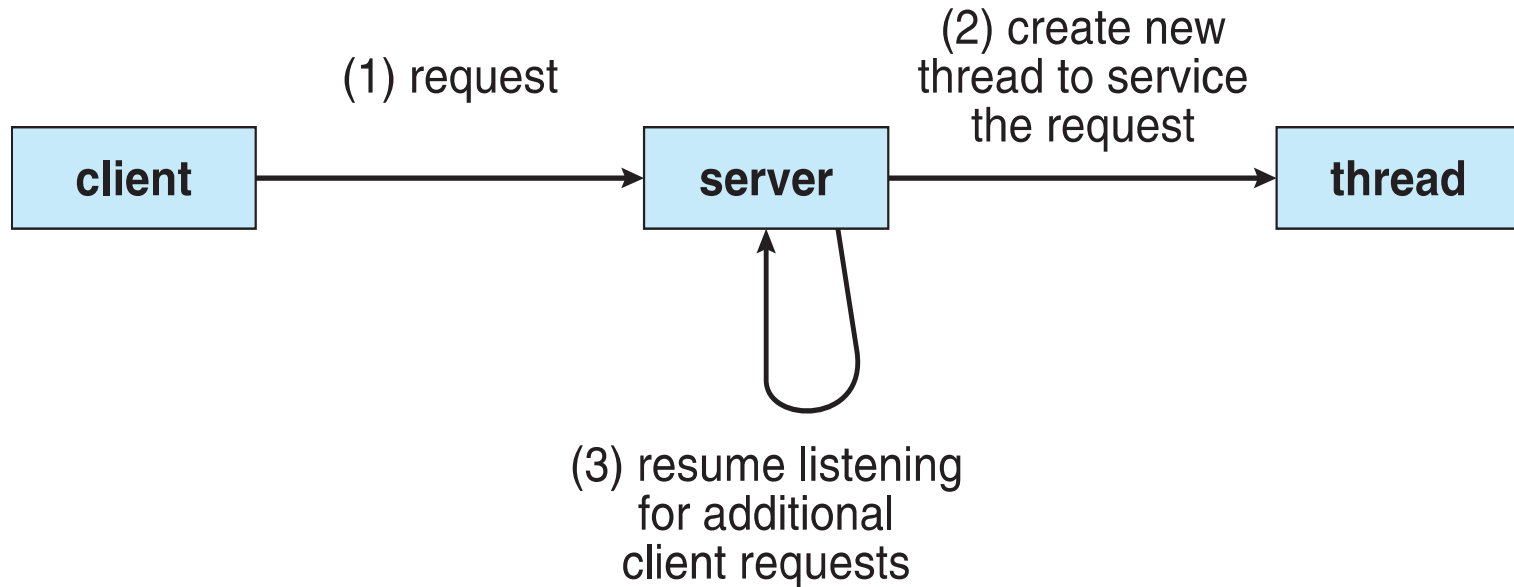
- **Pengolah kata**

- Thread untuk menerima dan menampilkan hasil pengetikan
- Thread untuk mengecek grammar

- **Webserver**

- Thread untuk menangani request dari banyak user

Contoh Multithread pada Webserver



Question

Mengapa harus multi-thread? Mengapa tidak pakai multiprocessing saja? **Apa keuntungan pakai multithread?**

Keuntungan Multithread

- **Responsiveness**

- Program tetap bisa berjalan walaupun ada sebagian tugas yang memakan waktu lama.
- Contoh : Web browser tetap dapat digunakan walaupun sedang ada aktifitas download.

- **Resource sharing**

- **Multiprocess** : Butuh implementasi shared memory atau message passing untuk berbagi data antar-proses.
- **Multithread** : Antar-thread dalam satu proses sudah berbagi data

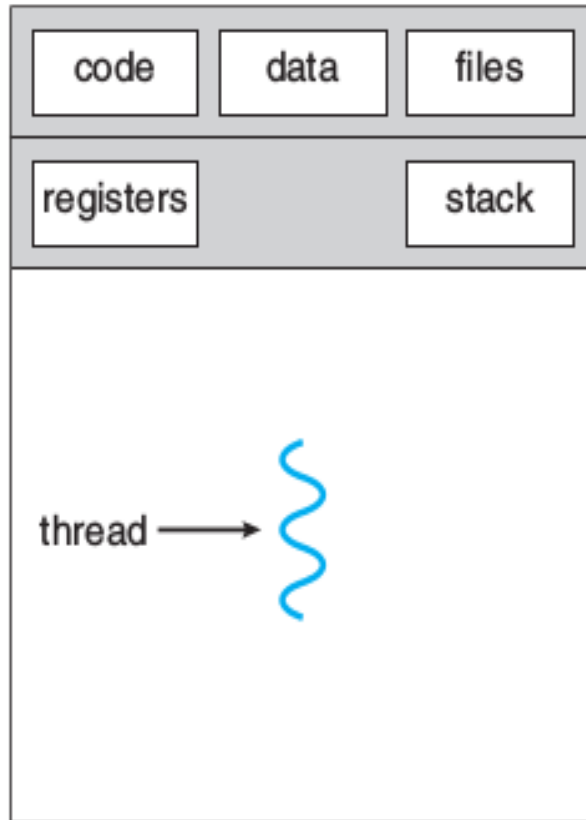
- **Economy**

- **Multiprocess** : Butuh alokasi banyak memori dan resource untuk membuat proses baru
- **Multithread** : Antar-thread sudah berbagi resource yang dialokasikan untuk sebuah proses

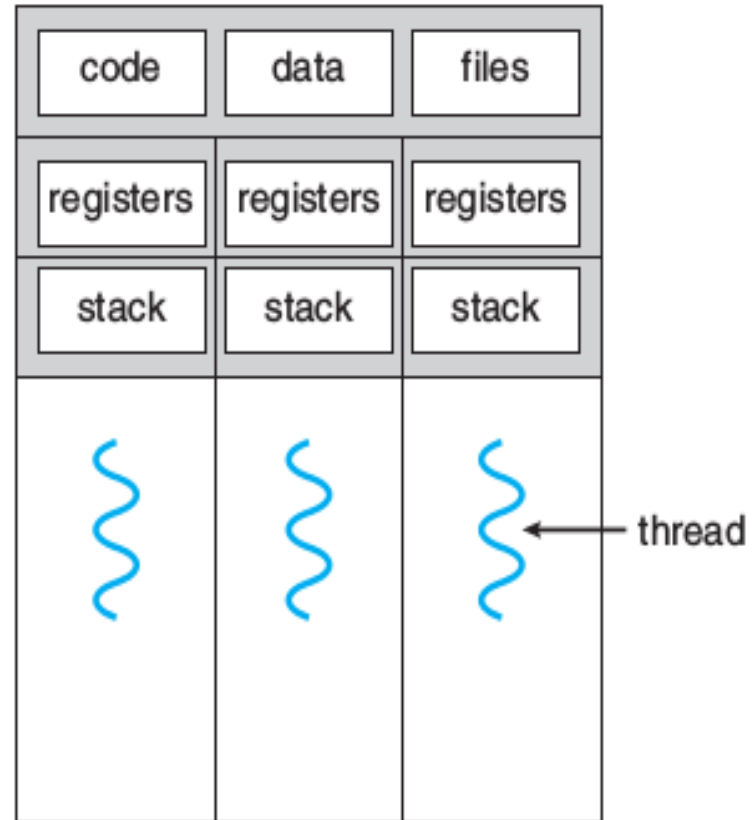
- **Scalability**

- Process : Satu processor hanya bisa mengeksekusi satu proses
- Thread : Setiap thread dalam satu proses dapat dieksekusi secara paralel oleh processor yang berbeda

Single Thread vs Multi Thread



single-threaded process



multithreaded process

MULTICORE PROGRAM

Multicore Program

- **Multicore/Multiprocessor** : ada lebih dari satu processor/core processor dalam satu system
- Pemrograman multithread memungkinkan pemanfaatan fitur multicore pada suatu system
- Contoh sebuah program dengan 4 threads
 - **Single core** :
 - Eksekusi thread dilakukan secara bergantian (**concurrent**)
 - **Multi Core** :
 - Eksekusi dilakukan secara **parallel** sesuai jumlah core
 - Memungkinkan **pembagian tugas** dalam satu program

Question

- Apa perbedaan concurrent dan parrarel?

Concurrency vs. Parallelism

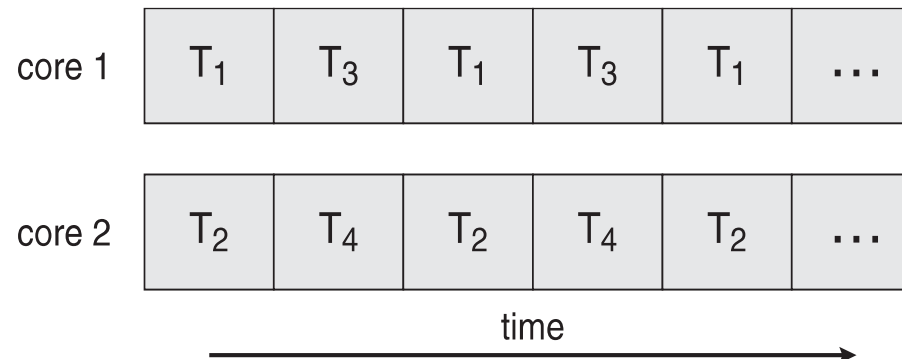
- **Concurrent execution pada single core**

- Hanya ada satu thread yang dieksekusi sekali waktu
- Thread dieksekusi dengan progress tertentu secara bergantian



- **Parallelism pada multi-core**

- Beberapa thread dieksekusi secara parallel sesuai jumlah core



Tantangan Pemrograman pada Sistem Multi-core

- **Identifikasi aktifitas**
 - Identifikasi bagian program mana saja yang dapat berjalan terpisah
- **Keseimbangan**
 - Tugas yang dipisah memiliki beban yang seimbang
- **Pemisahan data**
 - Bagaimana membagi data antar bagian program yang terpisah?
- **Ketergantungan data**
 - Bagaimana jika data pada satu tugas bergantung dengan data dari tugas lain?
- **Testing dan debugging**
 - Bagaimana melakukan testing dan debugging pada bagian program yang terpisah?

Tipe Parralelism

- Data parralelism
- Task parralelism

Data Parallelism

- Distribusi **bagian-bagian data** ke **masing-masing core processor berbeda** untuk diolah
- Operasi pada masing-masing core sama
- Contoh :
 - Operasi penjumlahan bilangan 1 sampai N
 - Ada 2 core processor
 - Penjumlahan data 1 sampai $(N/2)$ di core 1 dan $(N/2+1)$ sampai N di core 2

Task Parallelism

- Distribusi **tugas (thread)** berbeda ke **masing-masing core processor yang berbeda**
- Operasi pada masing-masing core beda
- Contoh :
 - Core 1 menghitung standard deviasi
 - Core 2 mencari nilai max

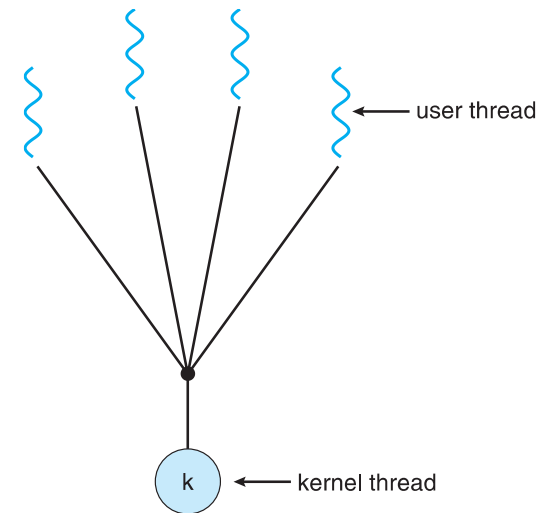
MULTITHREADING MODEL

User Thread vs Kernel Thread

- Berdasarkan pengendalinya, thread dibagi :
 - User thread
 - Kernel thread
- **User thread** dikendalikan oleh **program pada level user** tanpa campur tangan kernel
- **Kernel thread** dikendalikan langsung oleh **sistem operasi**
- Ada beberapa model hubungan antara kernel dan user thread

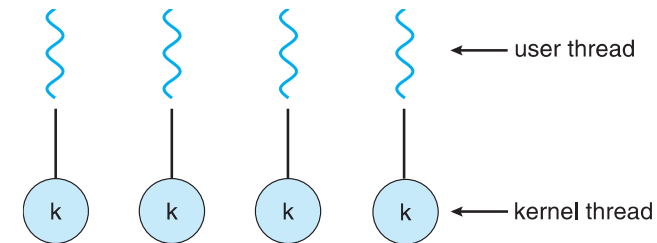
Model Many-to-One

- **Beberapa user-thread** dipetakan ke **satu kernel-thread**
- Kelebihan
 - Tidak ada batasan jumlah user-thread yang bisa dibuat
- Kelemahan
 - Satu thread melakukan blocking (misal I/O), seluruh proses ter-block
 - Multithread tidak dapat dijalankan secara parallel pada sistem multicore
- Contoh
 - **Solaris Green Threads**
 - **GNU Portable Threads**



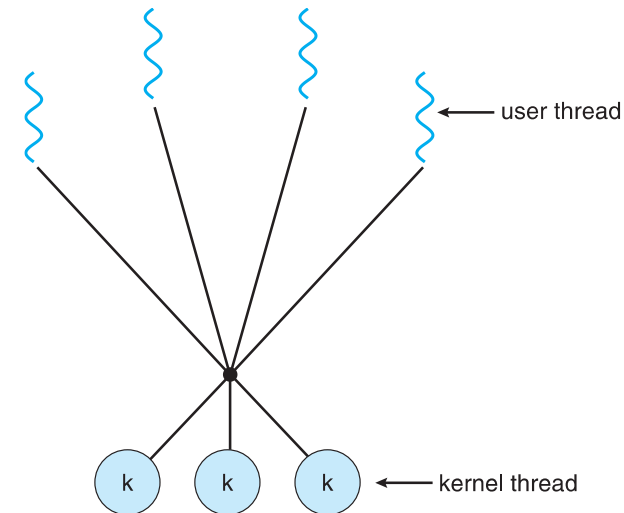
Model One-to-One

- **Satu user-thread** dipetakan ke **satu kernel-thread**
- Pembuatan satu user-thread = Pembuatan satu kernel-thread
- Kelebihan
 - Satu thread blocking tidak akan berpengaruh ke thread lain
 - Multithread dapat dijalankan secara parallel pada sistem multicore
- Kelemahan
 - Jumlah user-thread yang bisa dibuat **terbatas**
 - Pembuatan kernel-thread dapat membebani performa
- Contoh : Windows, Linux



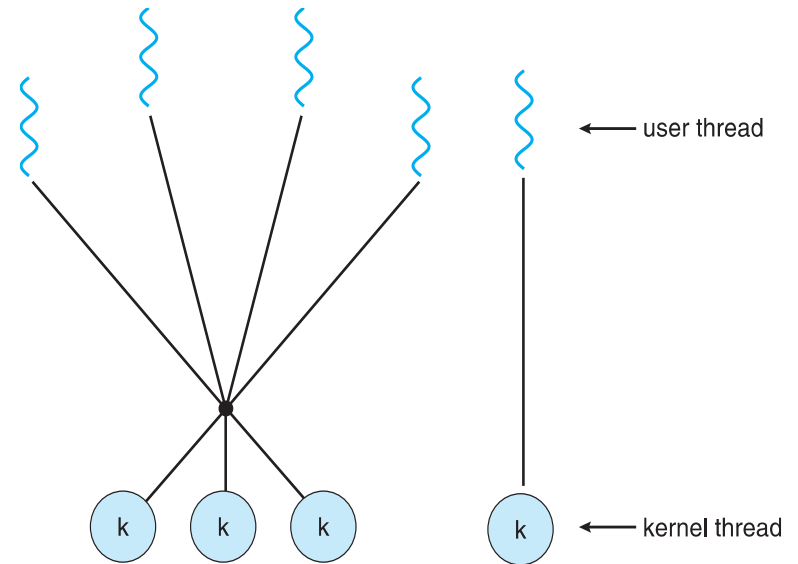
Model Many-to-Many

- **Beberapa user-thread** dipetakan ke **beberapa kernel-thread** (multiplexing)
- Jumlah kernel-thread yang dibuat bergantung aplikasi dan spesifikasi mesin
- Kelebihan :
 - Satu thread blocking tidak akan berpengaruh ke thread lain
 - Tidak ada batasan jumlah user-thread



Two Level Model

- Mirip M:M, tapi mengijinkan user-level thread dipetakan ke satu kernel thread
- Contoh :
 - IRIX
 - HP-UX
 - TRU64-Unix
 - Solaris 8 ke bawah

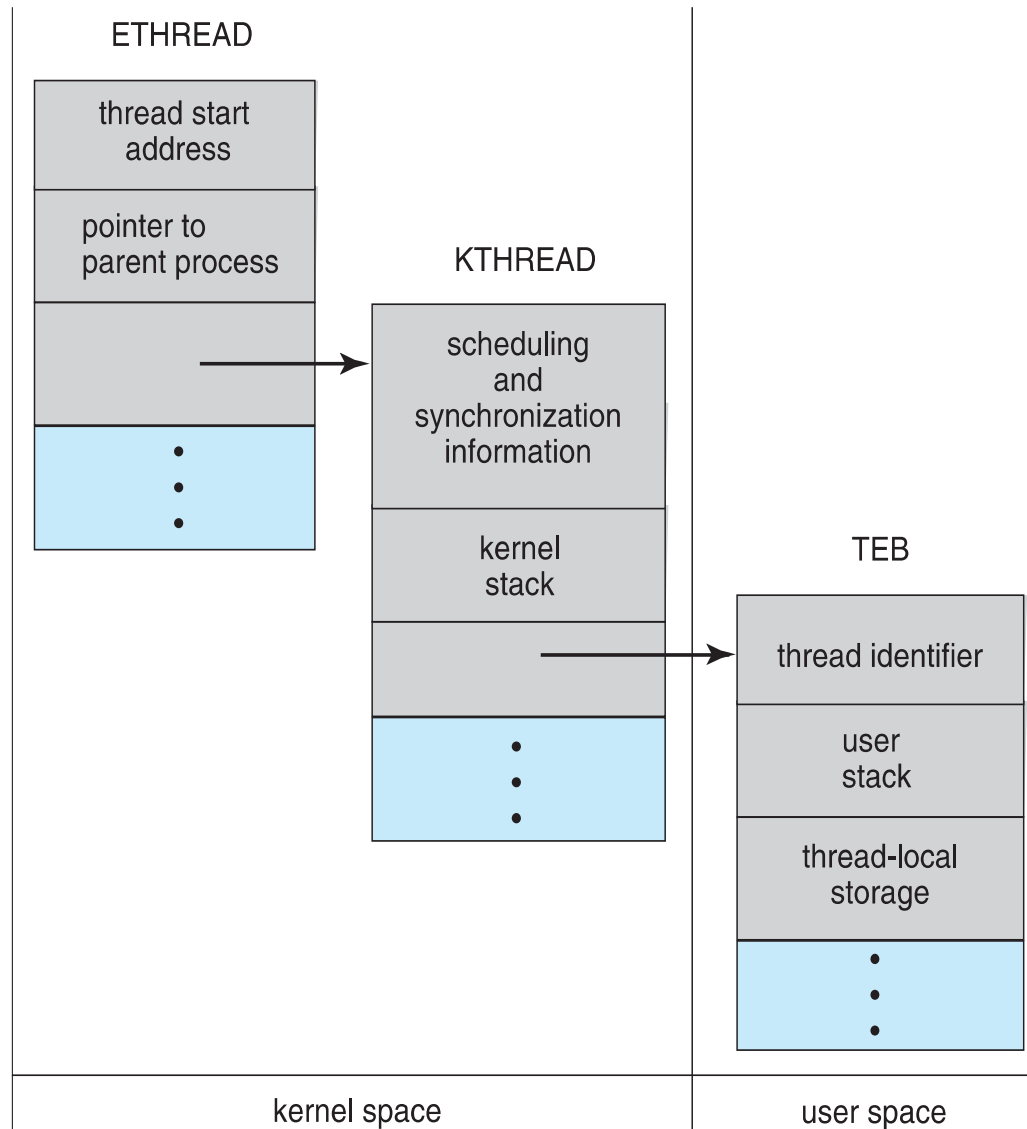


THREAD DALAM BERBAGAI SISTEM OPERASI

Windows Threads

- Thread di Windows dapat dibuat/diatur dengan Windows API
- Menerapkan one-to-one mapping, kernel-level
- Setiap thread mengandung
 - Thread id
 - Register set merepresentasikan status prosesor
 - Memisahkan user dan kernel stack ketika thread berjalan di user mode atau kernel mode
 - Private data storage area yang digunakan oleh run-time libraries and dynamic link libraries (DLLs)
- Register set, stacks, dan private storage area disebut sebagai **context** thread
- Struktur data sebuah thread meliputi:
 - ETHREAD (executive thread block) – pointer ke proses induk dari sebuah thread dan ke KTHREAD, dalam kernel space
 - KTHREAD (kernel thread block) – informasi penjadwalan dan sinkronisasi, kernel-mode stack, pointer ke TEB, dalam kernel space
 - TEB (thread environment block) – thread id, user-mode stack, thread-local storage, di user space

Windows Threads Data Structures



Linux Threads

- Linux menggunakan terminologi **tasks** daripada **threads**
- Pembuatan thread melalui system call **clone ()**
- **clone ()** memungkinkan thread anak untuk berbagi data dengan proses induk
 - Flags control behavior

flag	meaning
CLONE_FS	File-system information is shared.
CLONE_VM	The same memory space is shared.
CLONE_SIGHAND	Signal handlers are shared.
CLONE_FILES	The set of open files is shared.

- **struct task_struct** menunjuk ke process data structures (shared or unique)

TERIMA KASIH
