

THREAD-AT

KAPITULLI 5

Dr. Sc. Isak Shabani

Thread-at

- Hyrje
- Modelet Multithreading
- Libraritë e Threadave
- Çështjet me Threading
- Shembuj në Sistemet Operative
- Thredat në Windows XP
- Thredat në Linux

Objektivat

- Të njohim nocionin e thread-ave (fijeve), njësi themelore e ngarkimit të CPU-së e cila formon bazën e sistemeve kompjuterike me shumë thread .
- Të diskutohen API-t për Pthreads, Win32 dhe libraritë e thredave të Java.
- Të shqyrtohen çështjet e lidhura me programimin me shumë threda.

Dallimi ne mes te Thredave dhe Proceseve

Cfare jane Proceset ?

- Instance te ekzekutueshme te nje aplikacioni
- Programi është entitet pasiv, procesi është aktiv
- Një program mund të ketë disa procese
- Procesi prind krijon procese fëmijë, të cilët, në këtë mënyrë krijojnë procese të tjera, duke formuar një pemë të proceseve
- Komunikimi ne mes te proceseve i veshtire

Cfare jane Thread-at ?

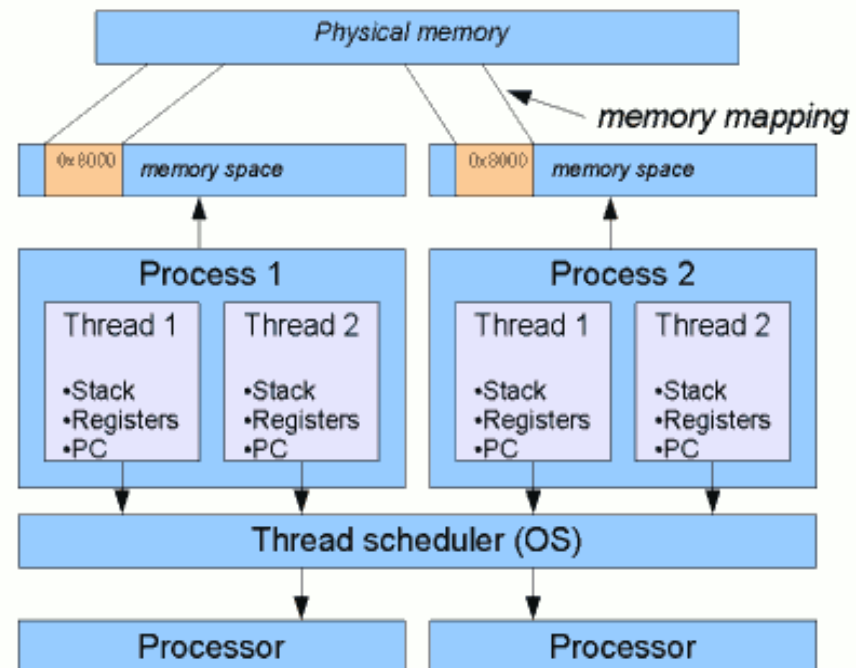
- Thread mund te beje cdo pune cka mund te beje nje proces
- Thread-at jane te nevojshme per procesimin paralel
- Multithreading mundeson kryerjen e shume puneve pernjehere
- Perdoren per pune te vogla
- Te varura nga njera tjetra

Dallimi ne mes proceseve dhe thread-ave

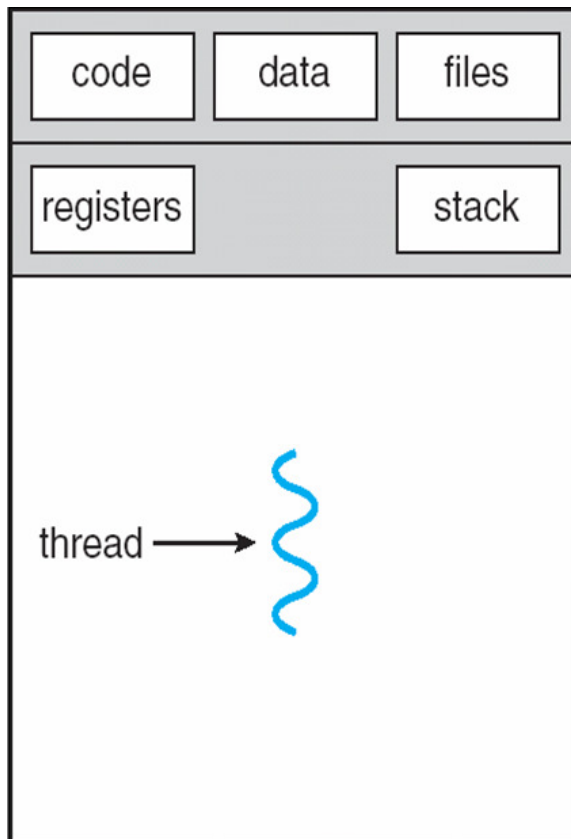
1. Thread krijohen me lehte
2. Proceset jane te pavarura njera prej tjetres, ndryshe nga thread-at
3. Procesi mund te permbaje shume thread-a
4. Thread-at perdorin shume pak burime ne krahasim me proceset

Si punojne Proceset dhe Threadat

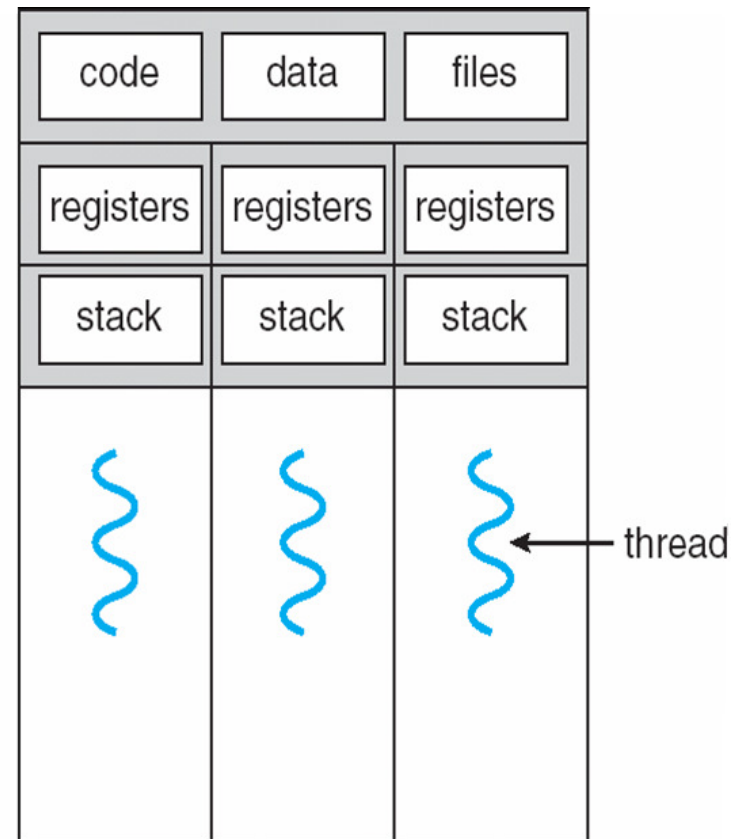
- Thread-i paraqet Proces Lightweight (LWP)
- Secili proces e ka hapsiren e vet ne memorie.
- Thread-i ndan hapsiren e memories me procesin prind.
- Thread ka stackun, regjistrat dhe program counterin.
- “housekeeping” bit.



Proceset me një thread dhe shumë threda



single-threaded process



multithreaded process

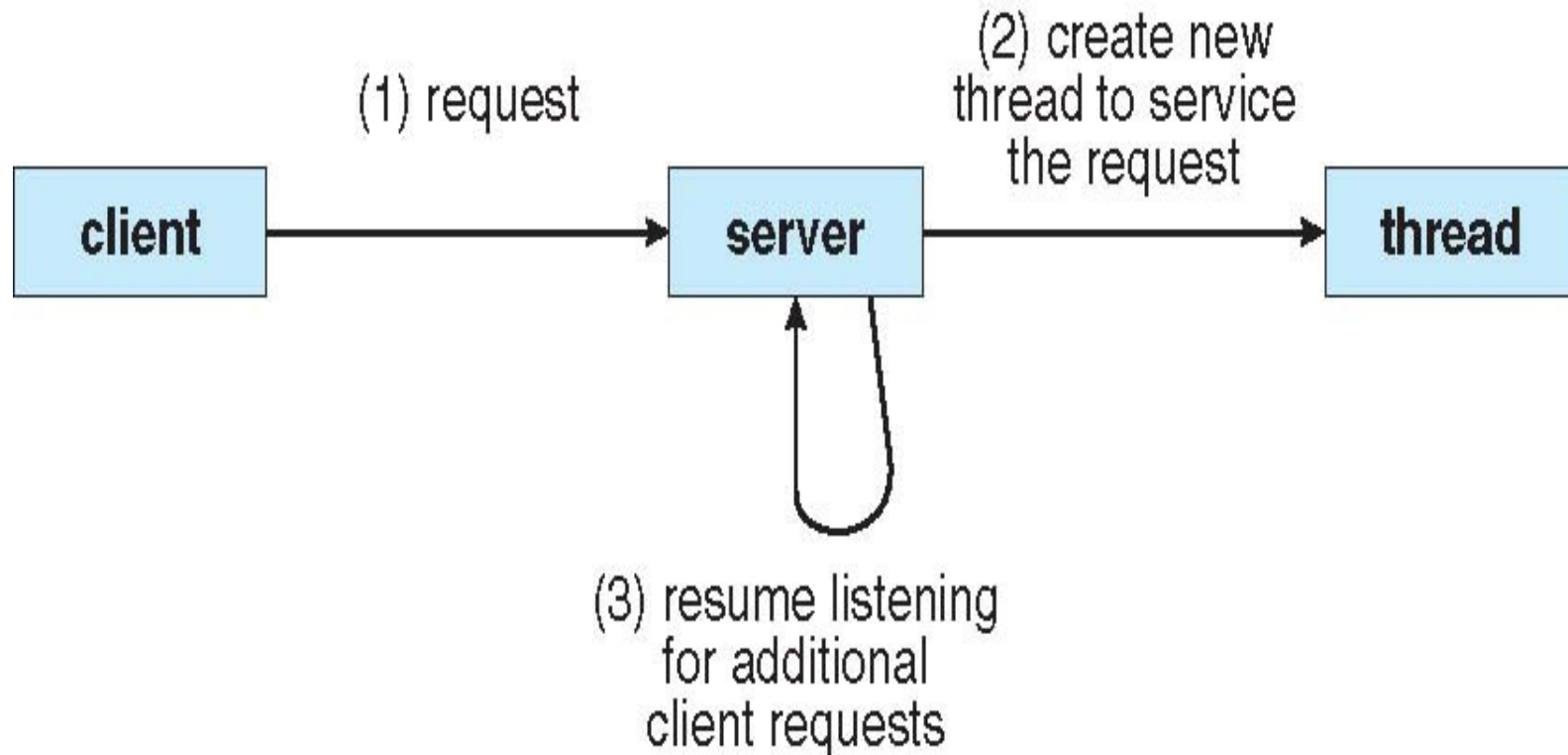
Përparësitë

- Përgjegjësia
- Ndarja e resurseve
- Ekonomikë
- Zgjerimi

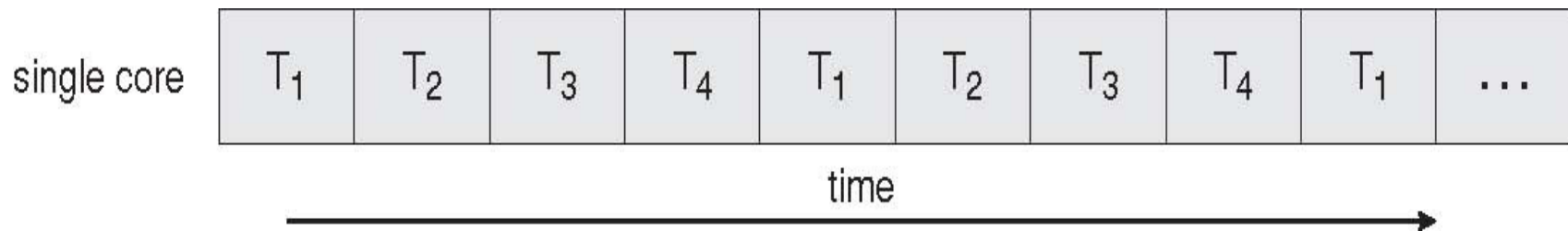
Programimi me shumë shtresa

- Sistemet me shumë shresa (Multicore) shtojnë presionin tek programerat, sfidat përfshijnë:
 - **Ndarjen e aktiviteteve**
 - **Balancim**
 - **Ndarjen e të dhënave**
 - **Vartësia e të dhënave**
 - **Testimi dhe gjetja e gabimeve (debugging)**

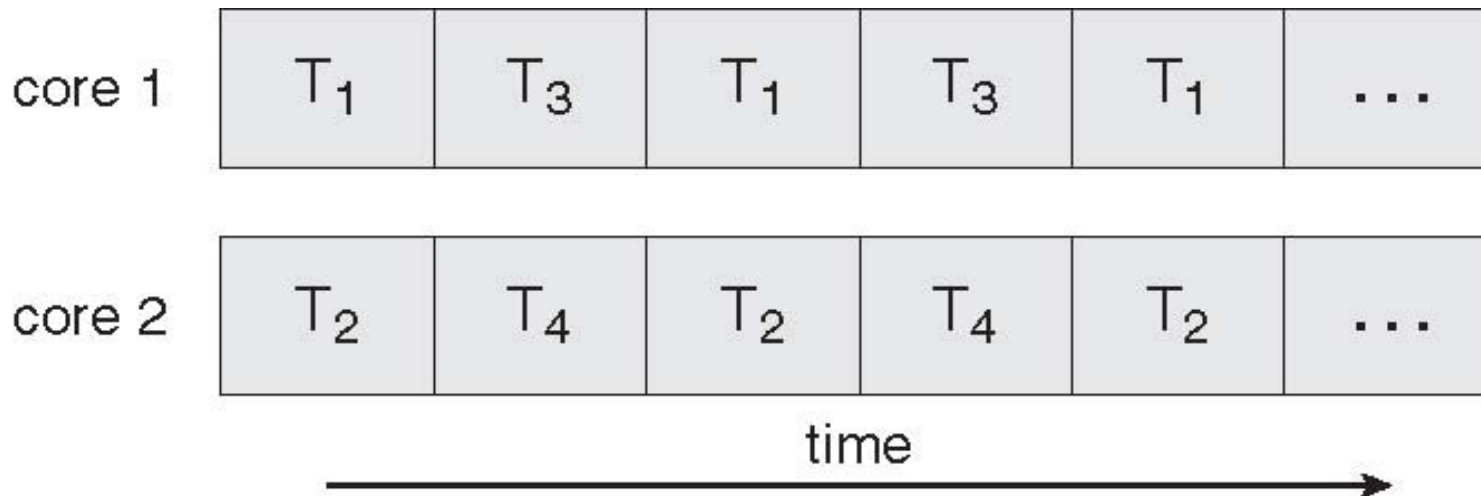
Arkitektura e Serverëve Multithread



Ekzekutimi konkurent në sistemin me një shtresë



Ekzekutimi Paralel në Sistemin me Shumë Shtresa



Thredat e Shfrytëzuesit

- Menaxhimi i thredave bëhet nga libraritë e thredave të nivelit të shfrytëzuesit.
- Tri libraritë primare të thredave janë:
 - POSIX **Pthreads**
 - Thredat Win32
 - Thredat ne Java

Thredat e Kernel-it

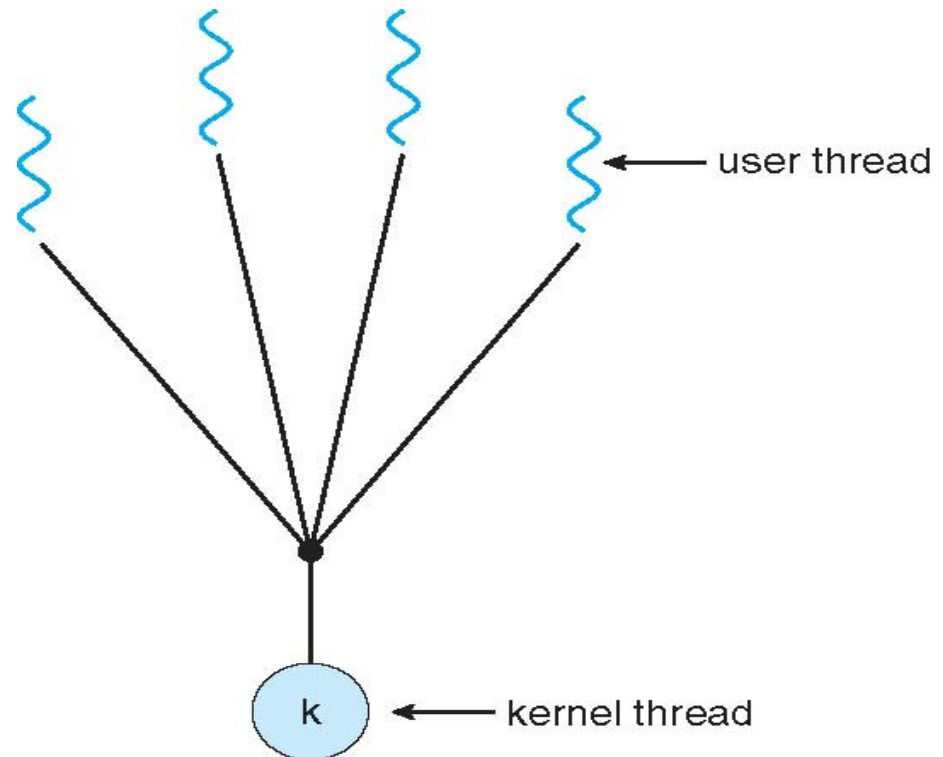
- Të përkrahura nga Kernel.
- Shembuj:
 - Windows XP/2000+
 - Solaris
 - Linux
 - Tru64 UNIX
 - Mac OS X

Modelet Multithreading

- Shumë-me-një
- Një-me-një
- Shumë-me-shumë

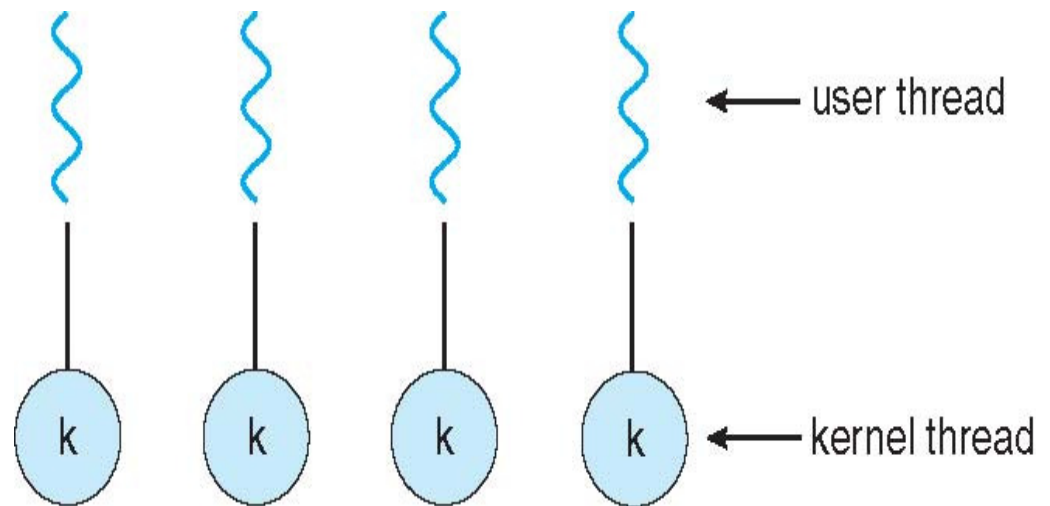
Modeli shumë-me-një

- Shumë thread me nivel-shfrytëzuesi të lidhura për një thread të vetme kerneli
- Shembuj:
 - **Solaris Green Threads**
 - **GNU Portable Threads**



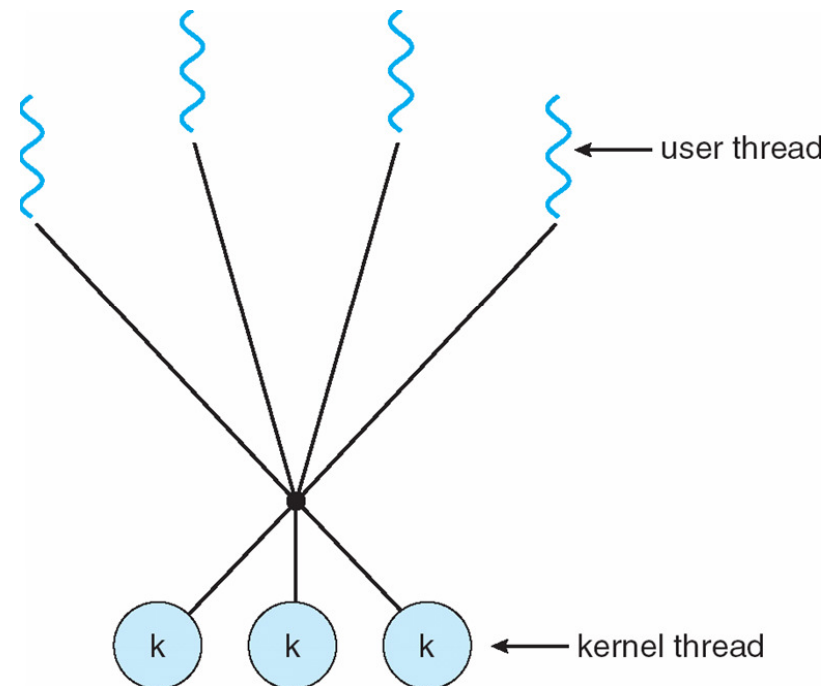
Modeli një-me-një

- Secili thread me nivel-të-shfrytëzuesit lidhet me një thread kernel
- Shembuj:
 - Windows NT/XP/2000
 - Linux
 - Solaris 9 dhe më të hershëm



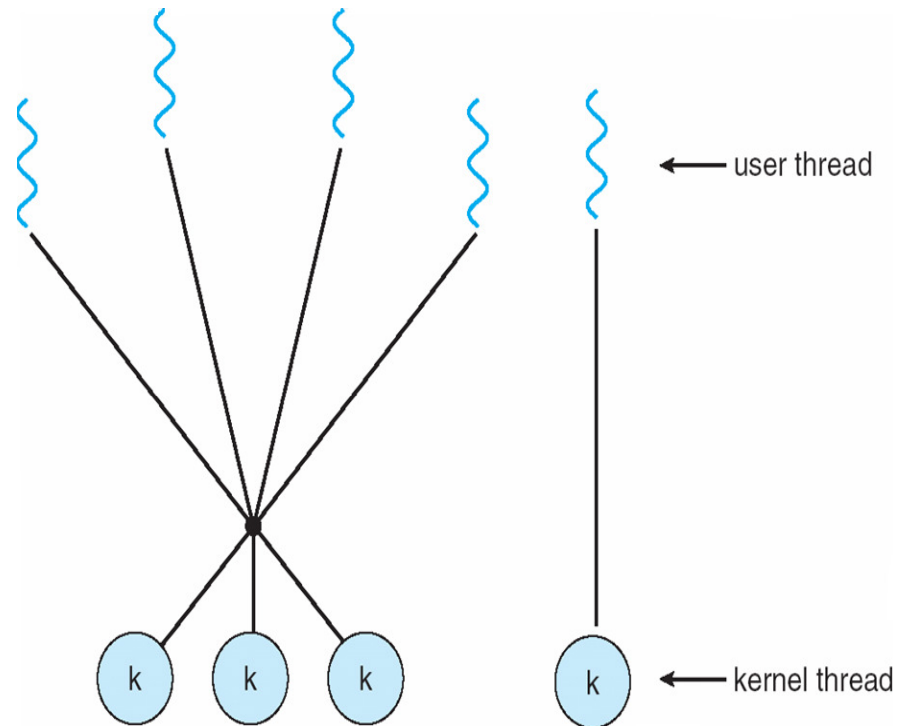
Modeli Shumë-me-Shumë

- Lejon disa thread me nivel shfrytëzuesi të lidhen me disa thread kerneli
- Lejon sistemin operativ të krijoj numër të mjaftueshëm të thredave të kernelit
- Solaris me fillim të versionit 9
- Windows NT/2000 me paketën *ThreadFiber*



Modeli Dy-Nivelësh

- Ngjashëm me Shumë:Shumë, përveç që lejon thread e shfrytëzuesit të **lidhet** me threadat e kernelit
- Shembuj:
 - IRIX
 - HP-UX
 - Tru64 UNIX
 - Solaris 8 e më herët.



Libraritë e thredave

- **Libraritë e thredave**u ofrojnë programerëve API për krijimin dhe menaxhimin e thredave
- Dy rrugë primare të implementimit
 - E tërë libraria në hapësirën e shfrytëzuesit
 - Libraria e nivelit Kernel e mbështetur nga SO

Pthreads

- Mund të ofrohen me nivel shfrytëzuesi ose nivel të kernelit
- Standardi i POSIX (IEEE 1003.1c) API për krijimin e thredave dhe sinkronizimin
- API përcakton sjelljen e librarisë së thredave, sa i përket implementimit varet nga zhvillimi i librarisë
- I përbashkët në sistemin operativ UNIX (Solaris, Linux, Mac OS X)

Thredat ne Java

- Thredat ne Java menaxhohen nga JVM
- Thredat ne Java mund të krijohen nga:
 - Implementimi i ndërfaqes lëshuese

```
public interface Runnable
{
    public abstract void run();
}
```

Threadat ne Java – Program Shembull

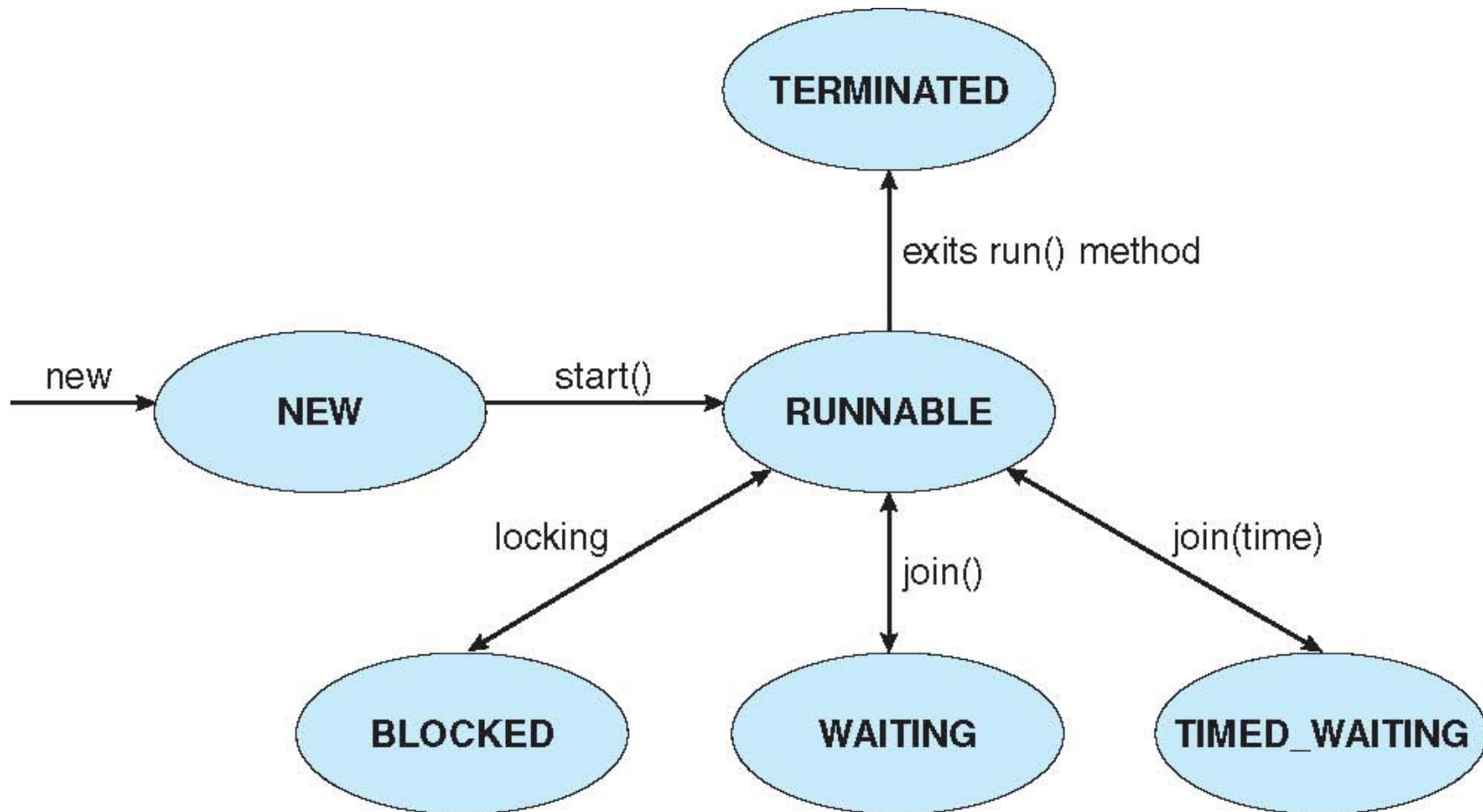
```
class MutableInteger
{
    private int value;
    public int getValue() {
        return value;
    }
    public void setValue(int value) {
        this.value = value;
    }
}

class Summation implements Runnable
{
    private int upper;
    private MutableInteger sumValue;
    public Summation(int upper, MutableInteger sumValue) {
        this.upper = upper;
        this.sumValue = sumValue;
    }
    public void run() {
        int sum = 0;
        for (int i = 0; i <= upper; i++)
            sum += i;
        sumValue.setValue(sum);
    }
}
```

Threadat ne Java – Program Shembull

```
public class Driver
{
    public static void main(String[] args) {
        if (args.length > 0) {
            if (Integer.parseInt(args[0]) < 0)
                System.err.println(args[0] + " must be >= 0.");
            else {
                // create the object to be shared
                MutableInteger sum = new MutableInteger();
                int upper = Integer.parseInt(args[0]);
                Thread thrd = new Thread(new Summation(upper, sum));
                thrd.start();
                try {
                    thrd.join();
                    System.out.println
                        ("The sum of "+upper+" is "+sum.getValue());
                } catch (InterruptedException ie) { }
            }
        }
        else
            System.err.println("Usage: Summation <integer value>");
    }
}
```

Gjendjet e Thredave ne Java



Thredat ne Java – Prodhuesi-Konsumatori

```
import java.util.Date;

public class Factory
{
    public static void main(String args[]) {
        // create the message queue
        Channel<Date> queue = new MessageQueue<Date>();

        // Create the producer and consumer threads and pass
        // each thread a reference to the MessageQueue object.
        Thread producer = new Thread(new Producer(queue));
        Thread consumer = new Thread(new Consumer(queue));

        // start the threads
        producer.start();
        consumer.start();
    }
}
```

Thredat ne Java – Prodhuesi-Konsumatori

```
import java.util.Date;

class Producer implements Runnable
{
    private Channel<Date> queue;

    public Producer(Channel<Date> queue) {
        this.queue = queue;
    }

    public void run() {
        Date message;

        while (true) {
            // nap for awhile
            SleepUtilities.nap();

            // produce an item and enter it into the buffer
            message = new Date();
            System.out.println("Producer produced " + message);
            queue.send(message);
        }
    }
}
```

Thredat ne Java – Prodhuesi-Konsumatori

```
import java.util.Date;

class Consumer implements Runnable
{
    private Channel<Date> queue;

    public Consumer(Channel<Date> queue) {
        this.queue = queue;
    }

    public void run() {
        Date message;

        while (true) {
            // nap for awhile
            SleepUtilities.nap();

            // consume an item from the buffer
            message = queue.receive();

            if (message != null)
                System.out.println("Consumer consumed " + message);
        }
    }
}
```

Çështjet me Thredat

- Semantikë e thirrjeve sistemore të **fork()** dhe **exec()**
- **Anulimi i Thredit target**
 - Asinkrone ose e vonuar
- Trajtimi i **Sinjalit**
- **Pishina e Thredave**
- **Të dhënat për Threda të caktuara**
- **Aktivizimi i Orarit**

Semantikë e `fork()` dhe `exec()`

- A e dyfishon **`fork()`** vetëm thredin thirrëse apo të gjitha thredat ?

Anulimi i Thredave

- Ndërprerja e thredit para se të ketë përfunduar
- Dy qasje të përgjithshme:
 - **Anulimi Asinkron** ndërpre në thredin e synuar menjëherë
 - **Anulimi i Vonuar (Deferred cancellation)** lejon thredin e synuar që të kontrolloj nëse duhet të anulohet

Trajtimi i Sinjalit

- Sinjalet përdoren në sistemin e UNIX për të njoftuar proceset se një ngjarje është shfaqur.
- **Trajtuesi i sinjalit** (signal handler) përdoret për procesimin e sinjaleve.
 - Sinjali gjenerohet nga një ngjarje e caktuar
 - Sinjali i dërgohet procesit
 - Sinjali është trajtuar
- Mundësitë:
 - Dërgoje sinjalin tek thredi ku do të aplikohet sinjali
 - Dërgoje sinjalit tek çdo thread në proces
 - Dërgoje sinjalin tek thredat e caktuara në proces
 - Caktoje thredin e caktuar të pranoj të gjitha sinjalet për procesin

Pishina e Fijeve (Thread Pool)

- Krijohet një numër thread në pishinë të cilët do të presin për punë
- Përparësitë:
 - Zakonisht më të shpejtë në shërbim të kërkesave me threda ekzistuese se sa të krijohet një thred i ri.
 - Lejon numrin e thredave në aplikacion(e) të lidhet me madhësinë e pishinës (ang. Pool).

Të Dhënat për thread të veçanta

- Lejon secilën thred të ketë kopje të vet të të dhënave
- E përdorshme kur nuk keni kontroll të krijimit të thredave (p.sh., kur përdoret pishina me threda)

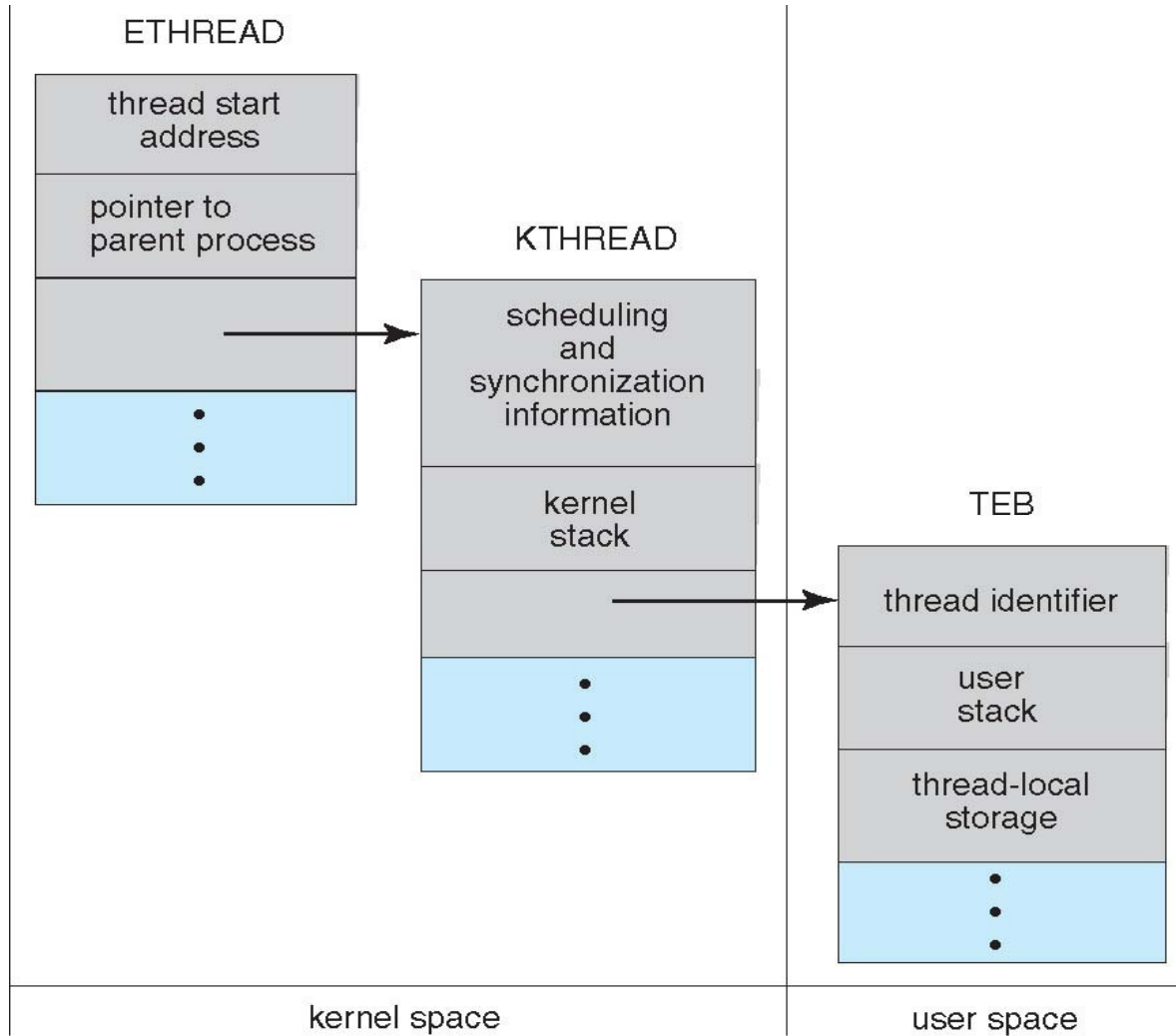
Aktivizimet e Orarit

- Të dyja Shumë:Shumë dhe modelet Dy-Nivelësh kërkojnë komunikimin për mirëmbajtje e numrit të nevojshëm të thredave të kernelit të alokuara për aplikacion
- Aktivizimet e orarit ofrojnë **upcalls** – mekanizëm për komunikim nga kerneli tek libraria e thredave
- Ky komunikim lejon aplikacionin të mirëmbajë numrin e saktë të thredave të kernelit

Shembuj të Sistemeve Operative

- Thredat ne Windows
- Thredat ne Linux

Thread in Windows XP



Thredat ne Linux

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.

Thredat ne Windows

- Implementon lidhjen një-me-një, në nivel të kernelit
- Secili thread përmbanë
 - ID e threadit
 - Bashkësinë e regjistrave (register set)
 - Linja të ndara të shfrytëzuesit dhe kernelit
 - Hapirë private e të dhënave
- Bashkësia e regjistrave, linjat (ang.stacks), dhe hapirë private e të dhënave janë të njohura si **kontekst** (ang. Context) i thredave
- Struktura e të dhënave primare të thredave përmbanë:
 - ETHREAD (blokun e ekzekutimit të thredave)
 - KTHREAD (blokun e thredave të kernel-it)
 - TEB (blokun e ambientit të thredave)

Thredat ne Linux

- Në Linux u referohen si *tasks (punë)* sesa *thread*
- Krijimi i thredave bëhet përmes thirrjes sistemore **clone()**
- **clone()** lejon një punë fëmijë të ndajë hapsirën e adresave me punën e prindit (procesit)