

Systemes d'exploitation

Threads

Tim Berners-Lee



- Anglais
- Oxford
- WWW

- threads
 - noyau
 - utilisateur
- NPTL
- Threads en Java
- Threads en Python



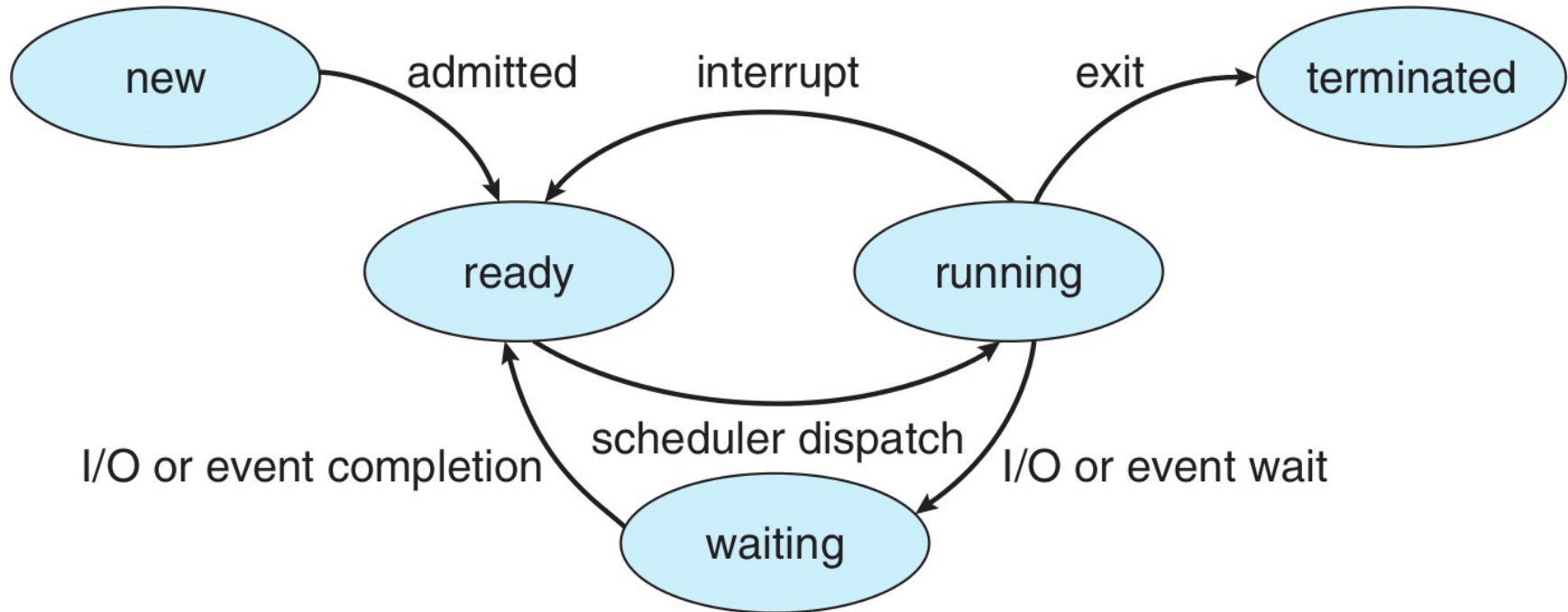
Bibliographie pour aujourd'hui

- Modern Operating Systems
 - Chapitre 2
 - 2.2
- Operating Systems Concepts
 - Chapitre 4

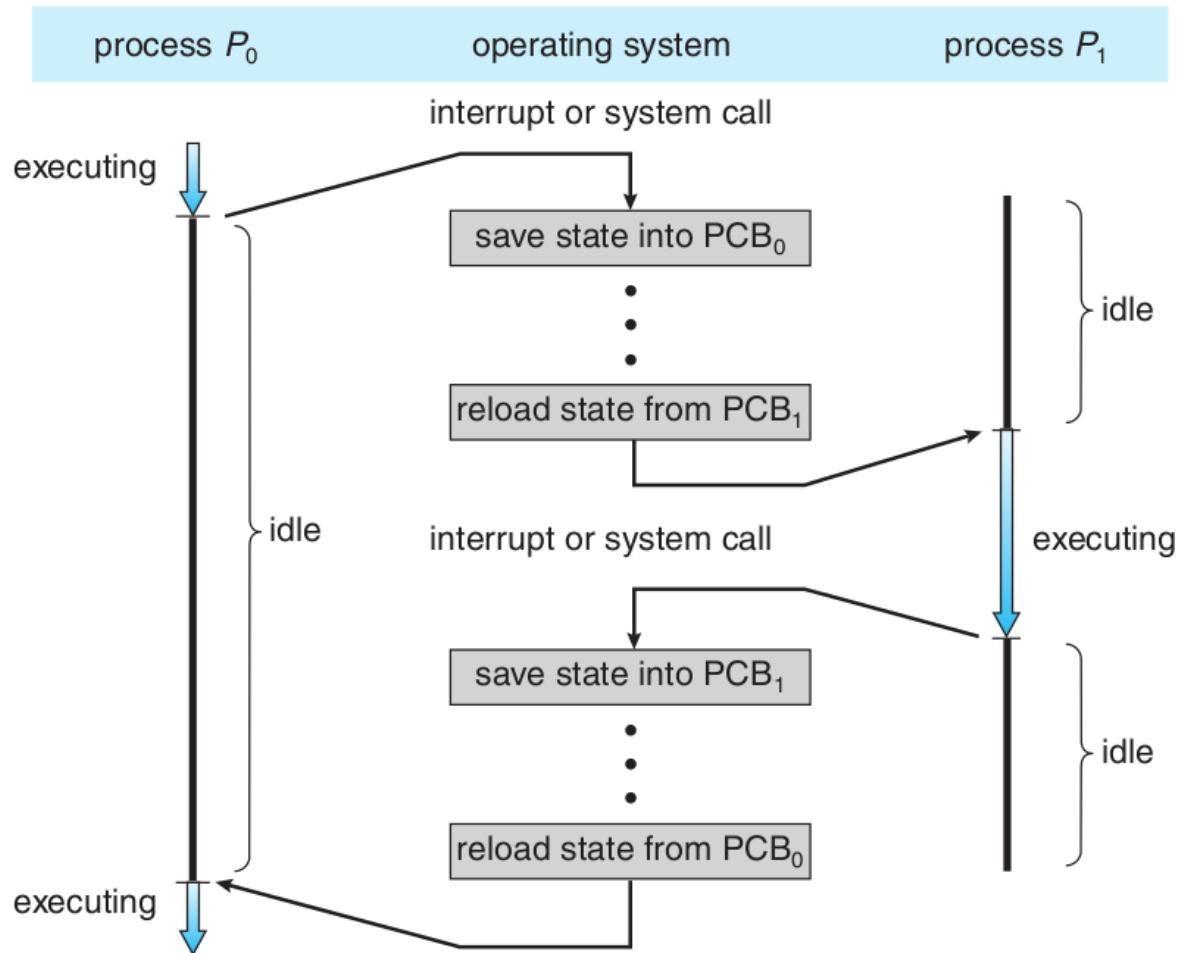
Types de processus (CPU)

- CPU bound (CPU intensive)
 - utilise beaucoup le processeur
- I/O bound (I/O intensive)
 - utilise rarement le processeur
 - faire des opérations d'E / S -> se bloque

Transitions entre les états



Changement de contexte



Exemple de processus

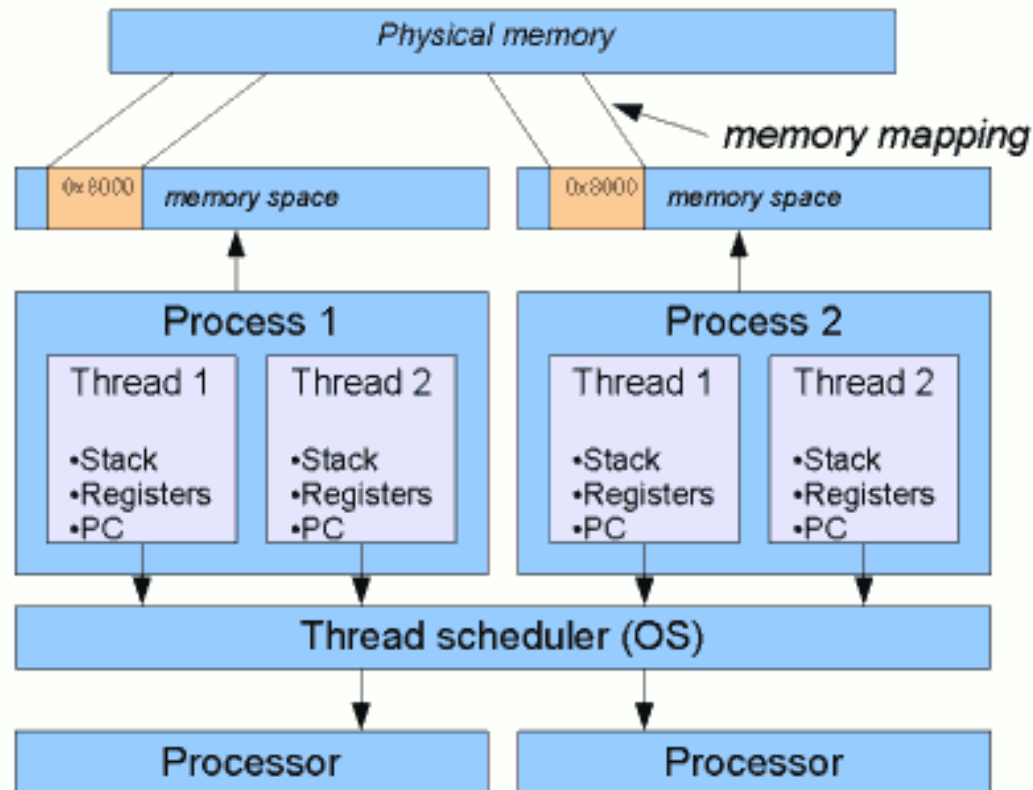
- Editeur du texte
- Serveur Web
- Jeux

THREAD

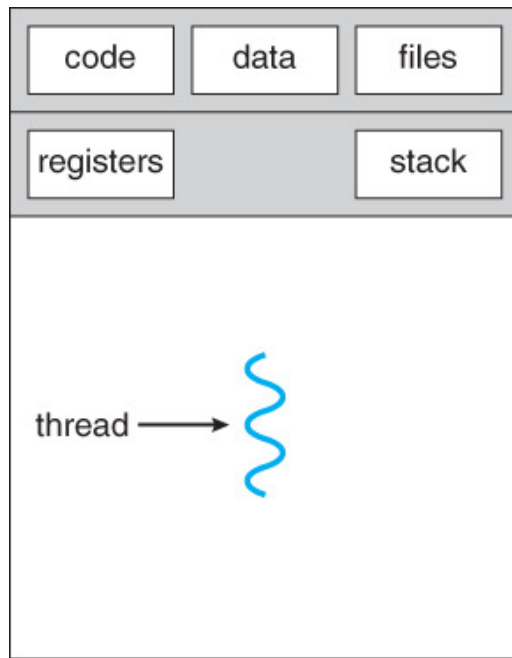
Threads

- L'unité de planification
- Un processus est formé de plusieurs de threads
- Lightweight Processes (LWP)

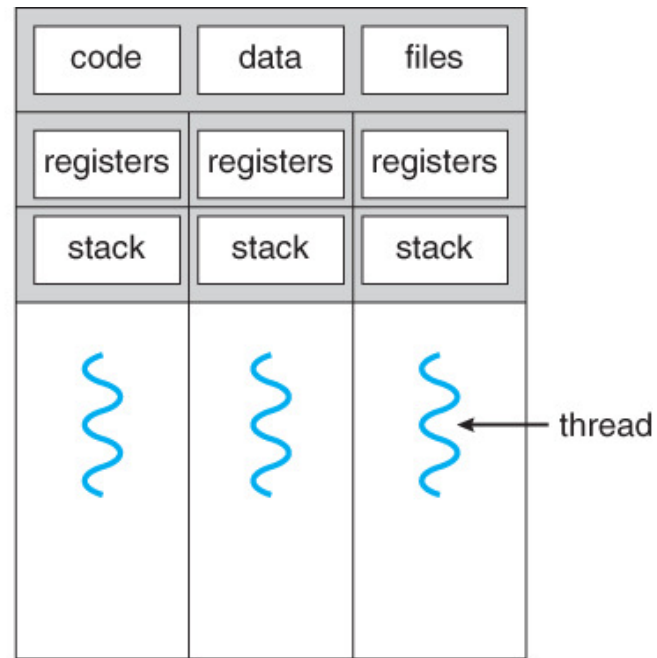
Processus et threads



Thread



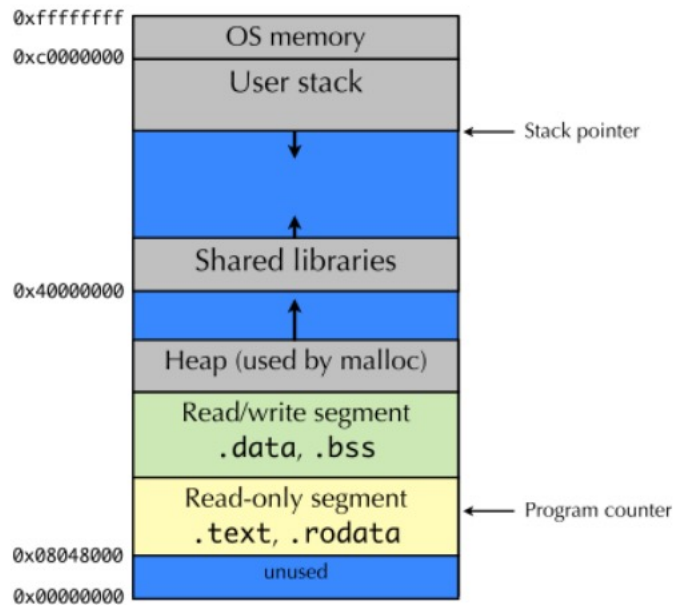
single-threaded process



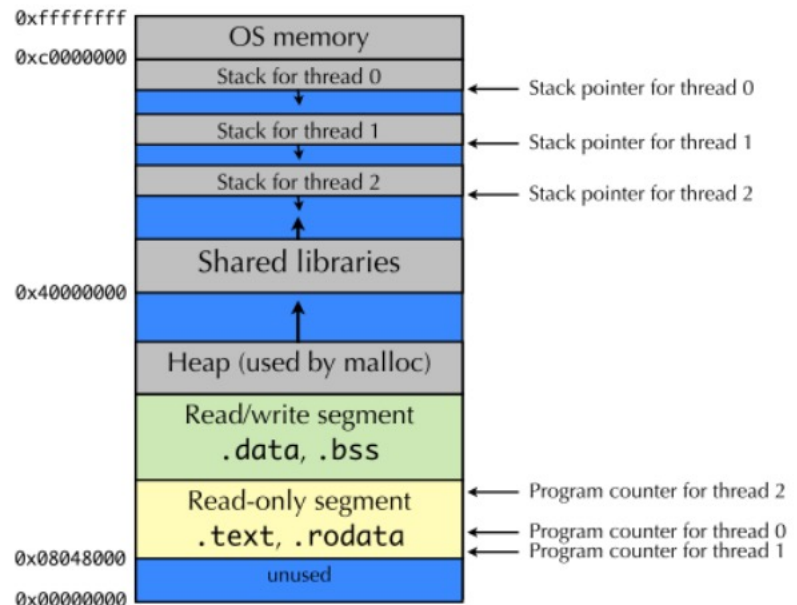
multithreaded process

Espace de adressage

Single threaded process:



Multi-threaded process:



Propriétés de thread

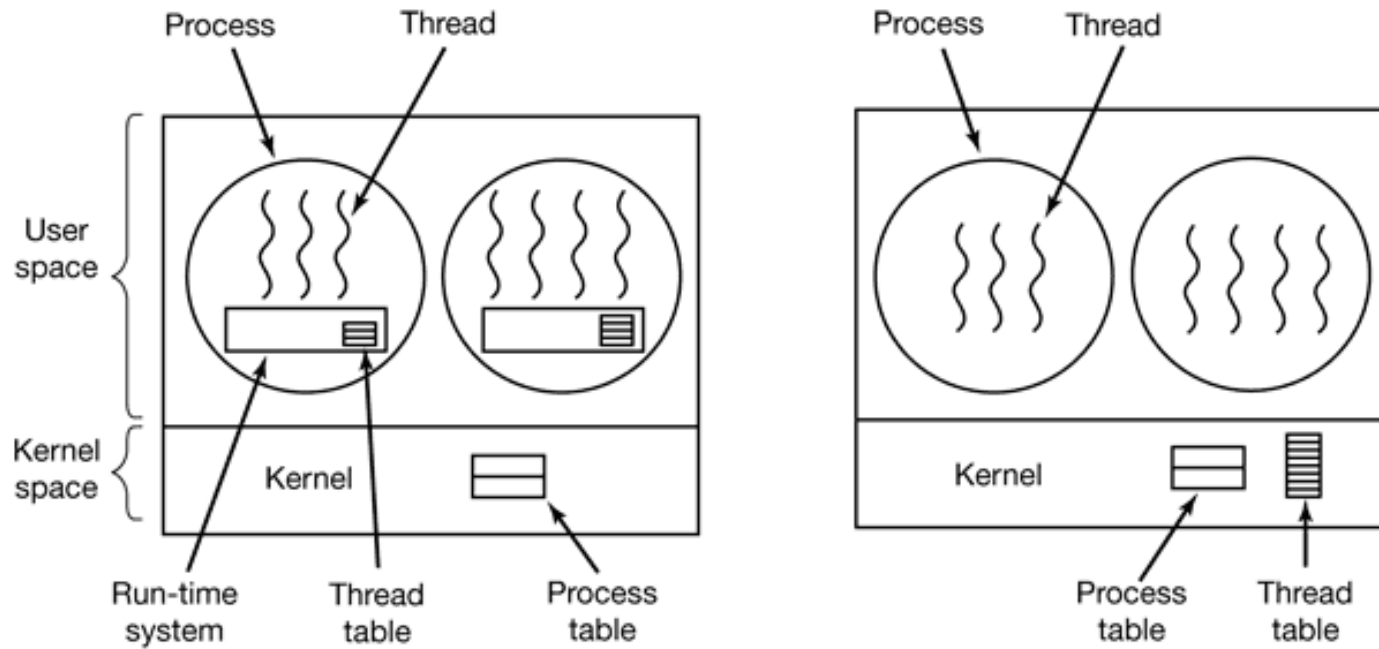
- TID – identificateur
- TLS – thread local storage
- Stack – la pile
- Registres de CPU
- État de processus
- Priority - la priorité

Types de thread

- Noyau
 - Sont connues par le noyau
 - Changement de contexte

- Utilisateur
 - Ne sont pas connus par le noyau
 - Pas de changement de contexte

Types de threads

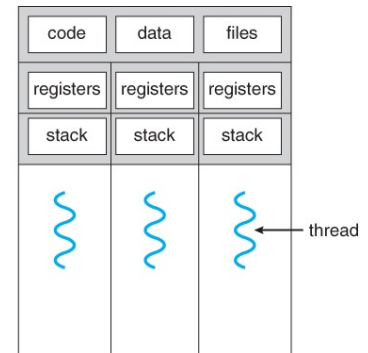
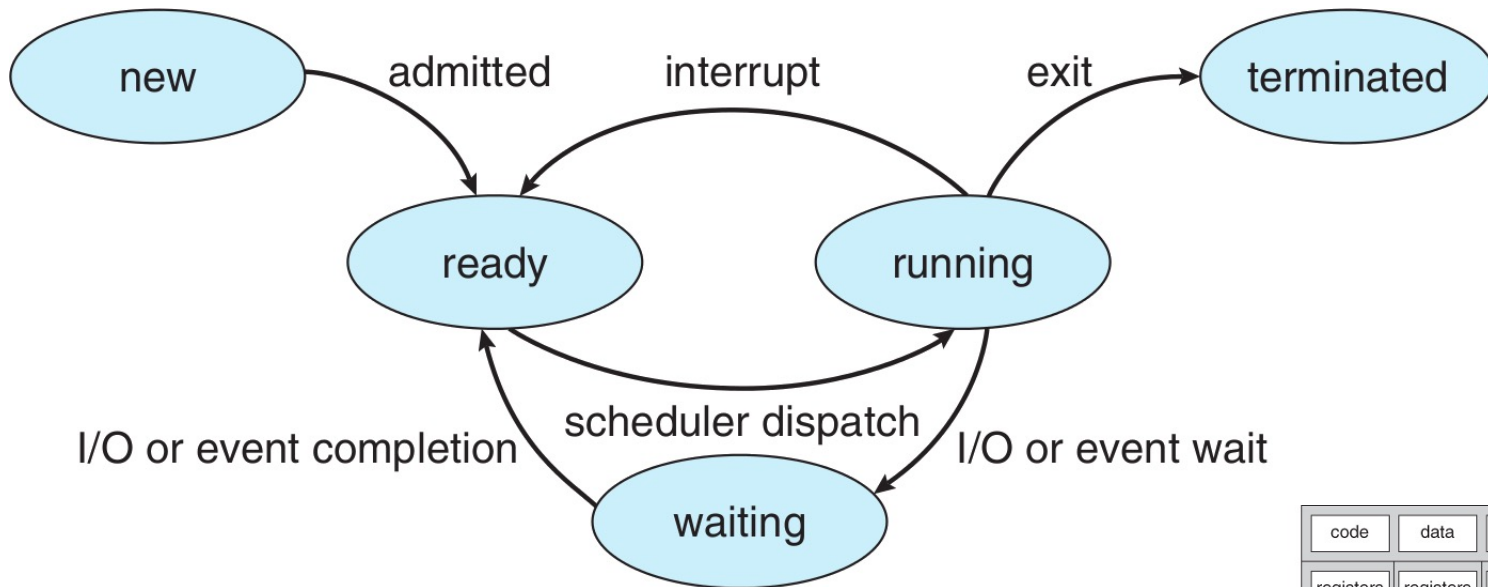


Question?

- Threads Utilisateur
 - Un thread fait un action bloquante
 - Qu'est qui ce passe?

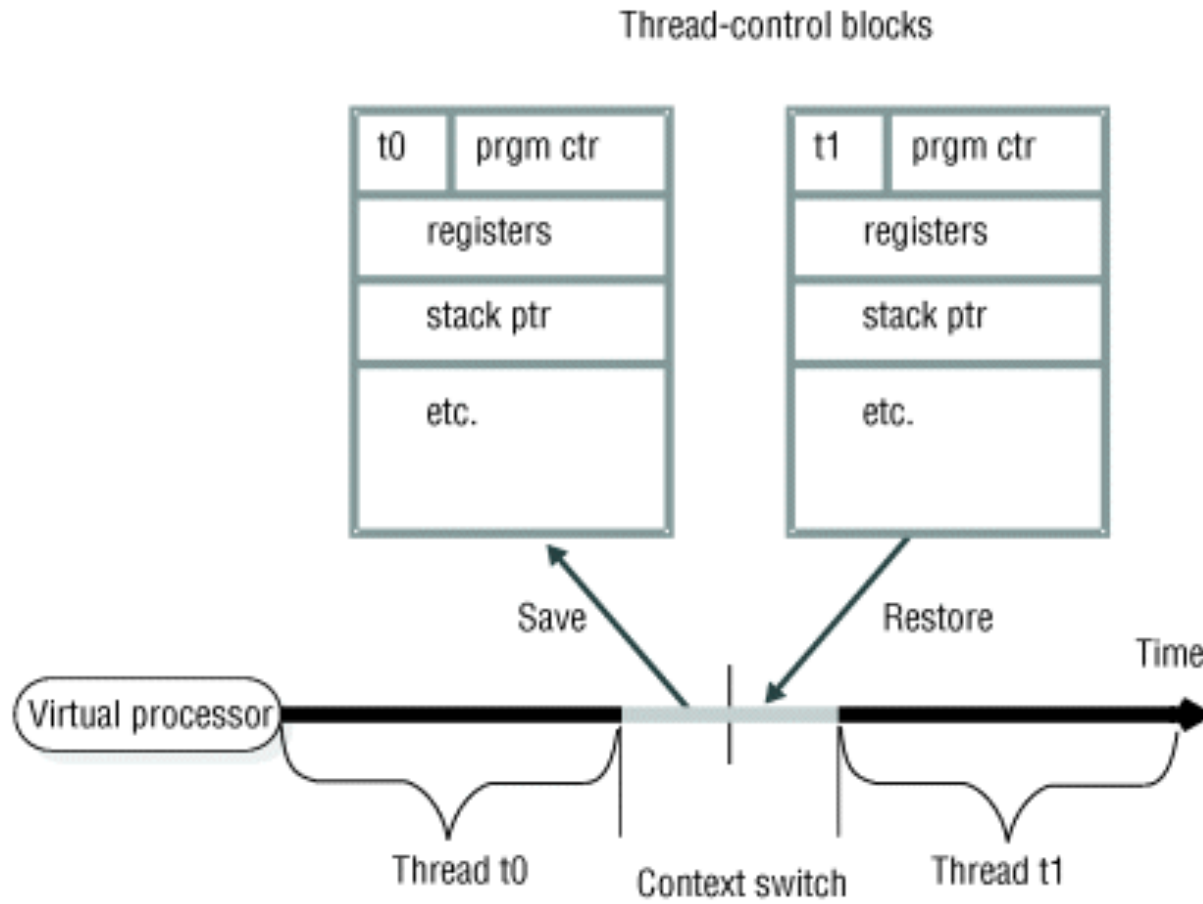
Toutes les threads se bloque

Les états de processus



multithreaded process

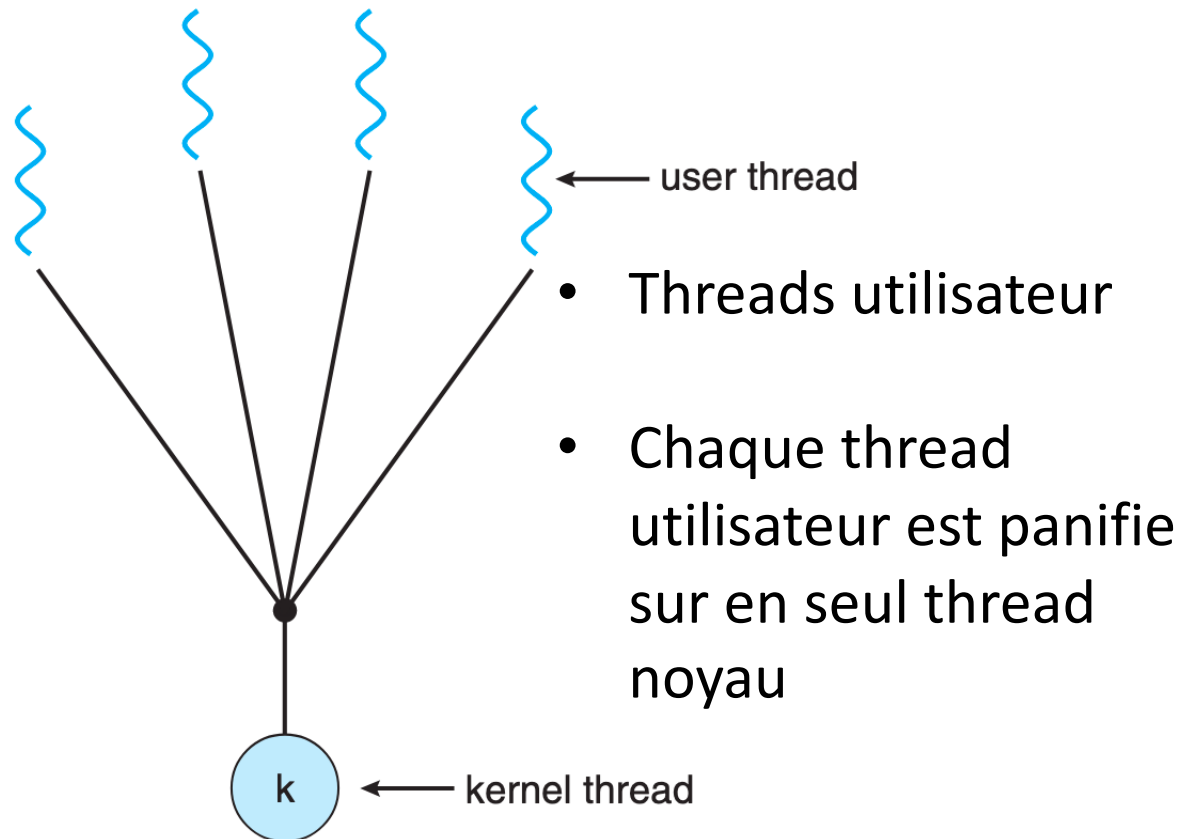
Changement de contexte



Implementation des threads

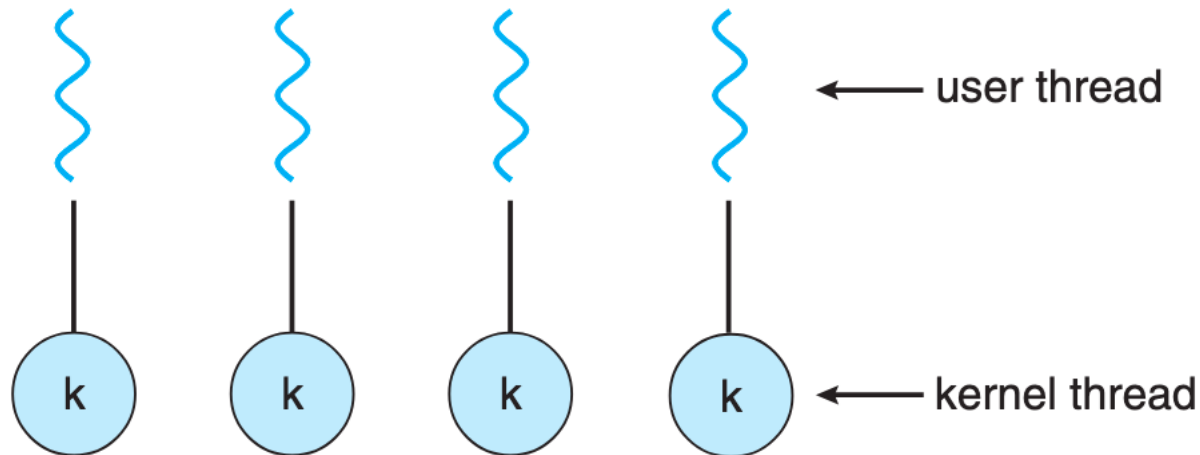
- Many to one
- One to one
- Many to many

Many to one

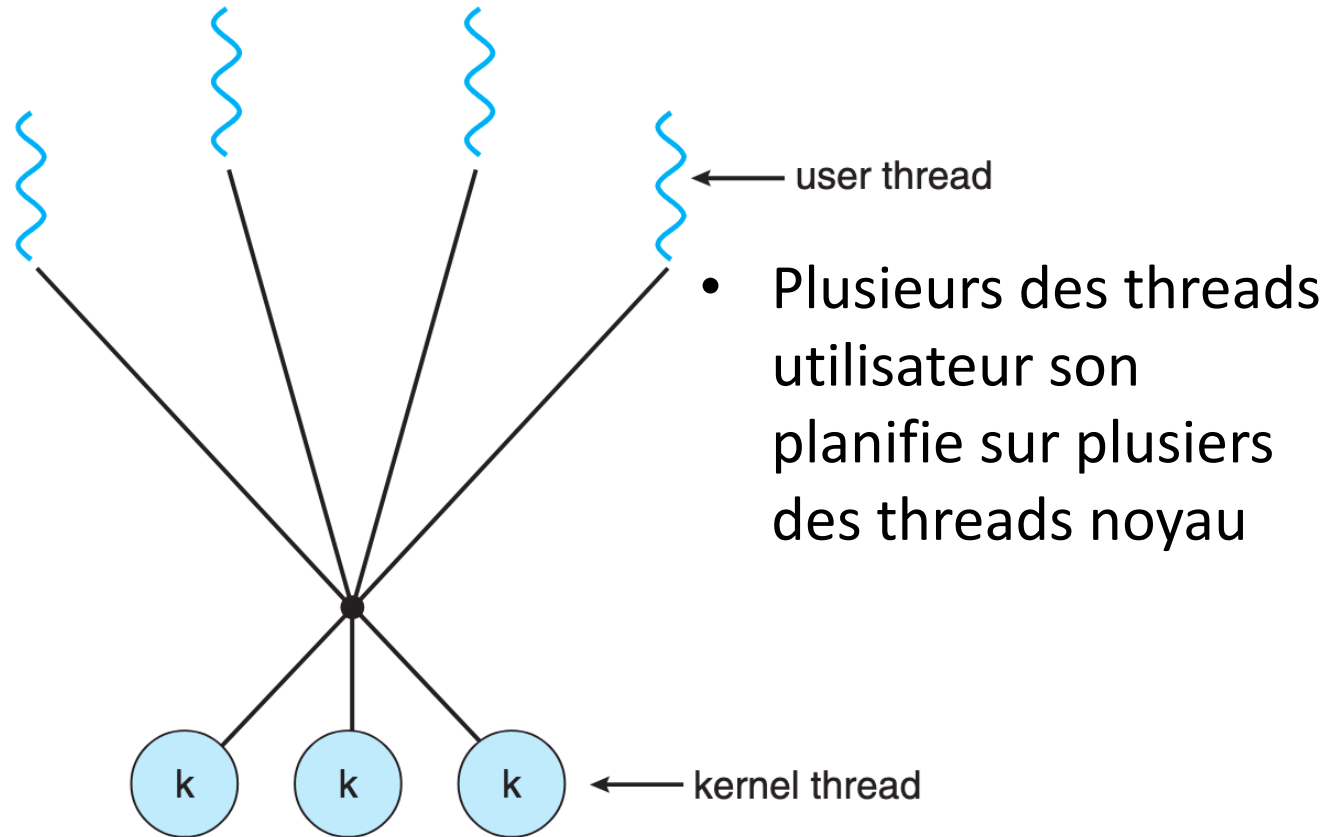


One to one

- Threads noyau
- Un thread est planifié sur un thread noyau



Many to many



Arrête d'un thread

- Asynchronous
 - Le thread este arrêté immédiatement
- Deferred
 - Le thread s'arrêté

Problemes

- fork et exec
 - fork fait un clone a tous les threads?
 - exec remplace un seul thread?

Implementation

- Windows
 - Threads
 - Fibers
- UNIX
 - pthreads (NPTL)
 - biblioteque
- Java
 - Many to many

Windows

- Threads
 - Noyau
 - Un processus a plusieurs des threads noyau

- Fibers
 - Utilisateur
 - Un thread noyau a plusieurs de thread utilisateur

POSIX

Nouveau Processus - UNIX

- **fork**: créer un nouveau processus (enfant) (presque identique au processus parent)
 - copie de mémoire
- **exec**: chargement d'informations d'un exécutable dans la mémoire du processus enfant
 - mémoire nouveau

Nouveau Thread - Linux

- **fork**: créer un nouveau processus (enfant)
(presque identique au processus parent)
 - Partage toutes les ressources
- Linux n'a pas l'idée de thread
- Un processus c'est un group des threads

pthread

- bibliothèque pthread (POSIX Thread)
- NPTL (New POSIX Thread Library)

Fonctions

```
// create a thread
int pthread_create(pthread_t *thread, const pthread_attr_t *attr,
                  void *(*start_routine)(void *), void *arg);

// stop a thread
int pthread_cancel(pthread_t thread);

// wait for a thread to finish
int pthread_join(pthread_t thread, void **value_ptr);

// send a signal to a thread
int pthread_kill(pthread_t thread, int sig);

// run a function once in a thread
int pthread_once(pthread_once_t *once_control, void (*init_routine)(void));

// detach a thread
int pthread_detach(pthread_t thread)

// exit from a thread
void pthread_exit(void *value_ptr)
```


Exemple

```
/* this function is run by thread */  
void *run(void *void_ptr)  
{  
    // ... thread code here  
    return NULL;  
}
```

Start

```
void *data;

int main ()
{
    int r;
    pthread_t run_thread;

    r = pthread_create(&run_thread, NULL, run, data);
    if (r != 0)
    {
        perror ("pthread_create");
    }
    return 0;
}
```

Problèmes

- fork și exec
 - tous les fils sont-ils clonés?
 - exec remplace-t-il un seul thread?

Threads et fork

- Un seul thread est cloné
- Ce n'est pas une bonne idée de l'utiliser dans un processus qui a des threads
 - descripteurs de fichiers
 - où s'arrêtent les autres fils
- détails
<https://www.linuxprogrammingblog.com/threads-and-fork-think-twice-before-using-them>

Exemple de descripteur de fichier

```
fd = open("file", O_RDWR | O_CREAT | O_TRUNC, 0600);  
if (fd < 0) {  
    perror ("open()");  
    return 0;  
}  
  
fcntl (fd, F_SETFD, FD_CLOEXEC);
```

Solution

```
int pthread_atfork(  
    void (*prepare)(void),  
    void (*parent)(void),  
    void (*child)(void)  
);
```

JAVA

Etendre Thread

```
class MyThread extends Thread
{
    public void run ()
    {
        // ... thread code here
    }
}
```


Implementer Runnable

```
class MyThread implements
Runnable
{
    public void run ()
    {
        // ... thread code here
    }
}
```

Start

```
// Thread
```

```
MyThread t = new MyThread();  
t.start ();
```

```
// Runnable
```

```
Thread t = new Thread (new MyThread());  
t.start ();
```

Arrête du thread

- quand la fonction run se fin

- thread
- thread utilisateur
- thread noyau
- fiber
- many to one
- one to many
- many to many
- join
- Thread ID
- TLS
- pthread
- NPTL
- asynchronous exit
- deferred exit

Questions

