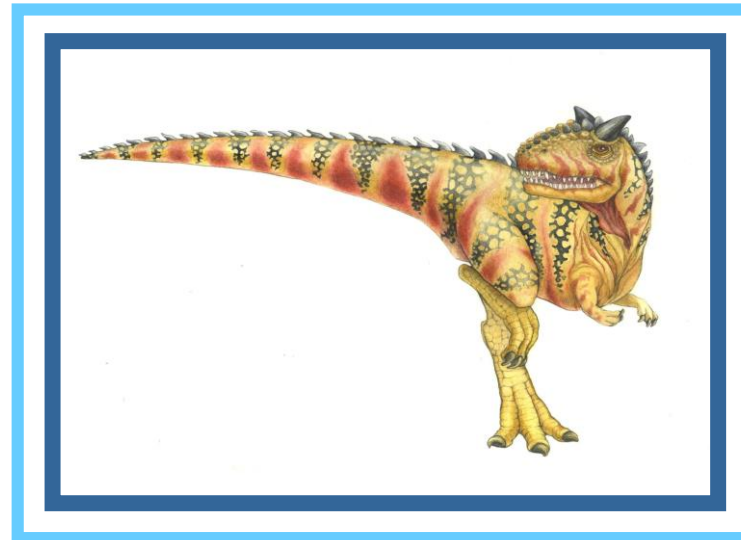


مؤشرات البرمجة: الفصل 4





مؤشرات البرمجة: الفصل 4

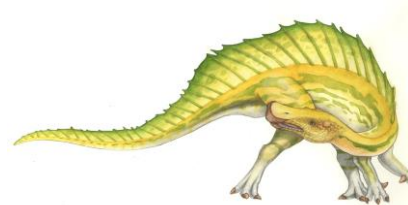
- n نظرة عامة
- n متعددة النواة البرمجة
- n نماذج خاصة تعدد
- n المكتبات الموضوع
- n ضمني التخطيط
- n قضايا خيوط
- n أمثلة نظام التشغيل





أهداف

- n لتقديم فكرة وجود موضوع الوحدة الأساسية للاستخدام وحدة المعالجة المركزية التي تشكل أساس أنظمة الكمبيوتر مؤشرات
- n لمناقشة واجهات برمجة التطبيقات لبثريديس، ويندوز، والمكتبات موضوع جاوة
- n لاستكشاف العديد من الاستراتيجيات التي توفر خيوط الضمني
- n لدراسة المسائل المتعلقة بالبرمجة مؤشرات
- n Linux و Windows لتغطية التشغيل دعم نظام للمواضيع في نظام التشغيل





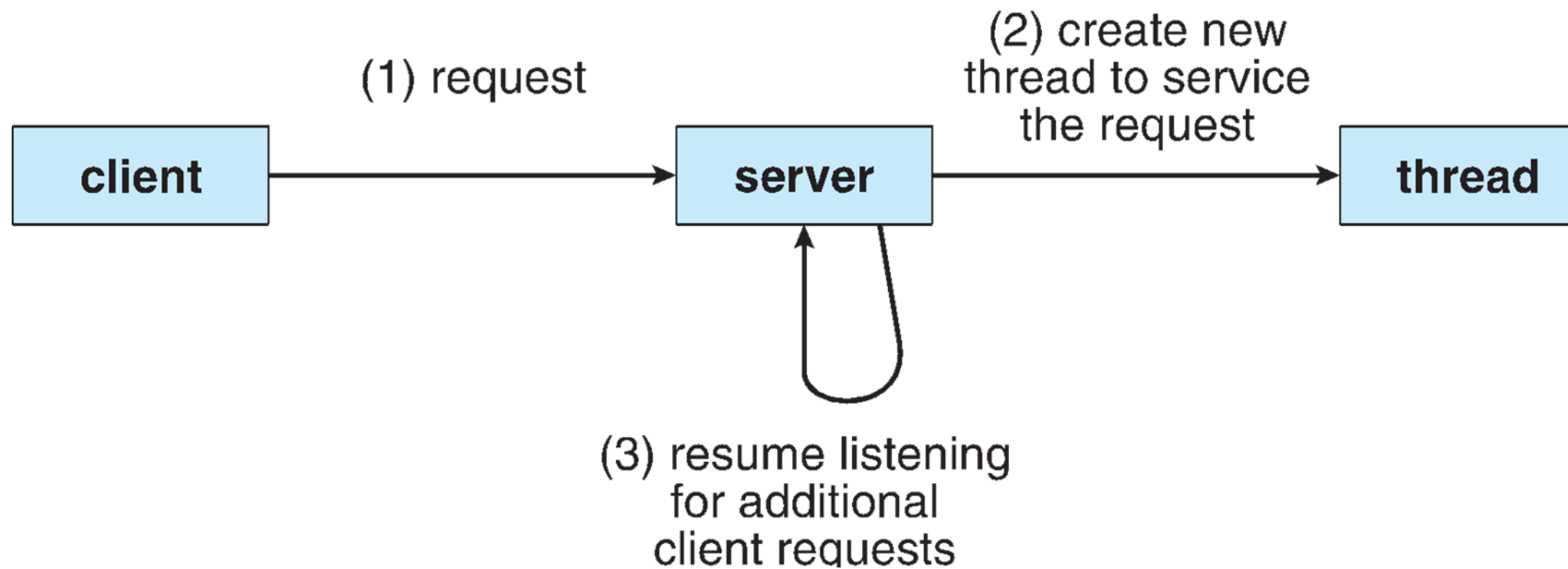
التحفيز

- n مؤشرات معظم التطبيقات الحديثة
- n المواضيع تشغيل داخل التطبيق
- n مهام متعددة مع التطبيق يمكن أن تنفذها المواضيع منفصلة
 - | تحديث العرض
 - | جلب البيانات
 - | التدقيق الإملائي
 - | الرد على طلب شبكة
- n إنشاء عملية ثقيل الوزن في حين إنشاء موضوع خفيف الوزن
- n يمكن تبسيط الرمز، وزيادة الكفاءة
- n ومؤشرات حبات عموما





خادم العمارة مؤشرات

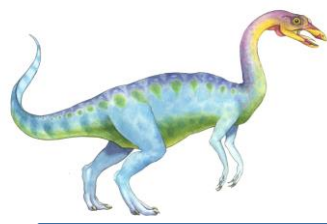




فوائد

- n قد تسمح باستمرار التنفيذ إذا تم حظر جزء من العملية، أهمية خاصة لواجهات المستخدم -الاستجابة
- n المواضيع موارد حصة العملية، أسهل من الذاكرة المشتركة أو تمرير رسالة -تقاسم الموارد
- n أرخص من إنشاء العملية، موضوع التحول خفض النفقات العامة من تبديل السياق -اقتصاد
- n عملية يمكن الاستفادة من أبنية متعددة المعالجات -قابلية التوسع





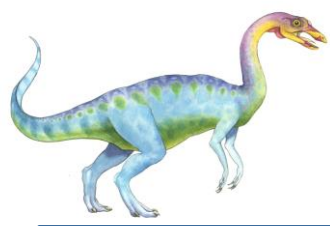
متعددة النواة البرمجة

- n **متعددة النواة أو متعدد المعالجات** نظم الضغط على المبرمجين، وتشمل التحديات
 - | تقسيم الأنشطة
 - | توازن
 - | تقسيم البيانات
 - | الاعتماد على البيانات
 - | الاختبار والتصحيح

- n **تواز** يعني نظام يمكن أن تؤدي أكثر من مهمة واحدة في وقت واحد
- n **التزامن** يدعم أكثر من مهمة واحدة تحرز تقدما
 - | الأساسية، جدولة توفير التزامن /واحد معالج
- n أنواع من التوازي
 - | توزع مجموعات فرعية من نفس البيانات عبر متعددة النوى، نفس العملية على كل -**التوازي البيانات**
 - | توزيع المواضيع عبر النوى، كل موضوع تنفيذ عملية فريدة من نوعها -**التوازي المهمة**

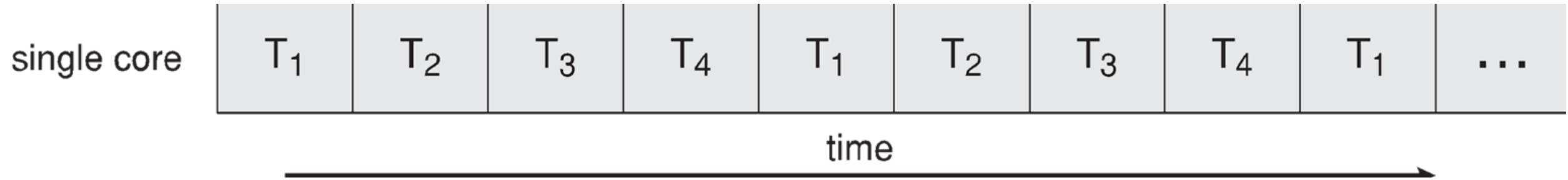
- n من المواضيع ينمو، حتى لا دعم المعماري للخيوط #كما
 - | وحدات المعالجة المركزية لها النوى وكذلك **المواضيع الأجهزة**
 - | المواضيع الأجهزة لكل نواة 8 النوى، و 8 مع SPARC T4 النظر في أوراكل



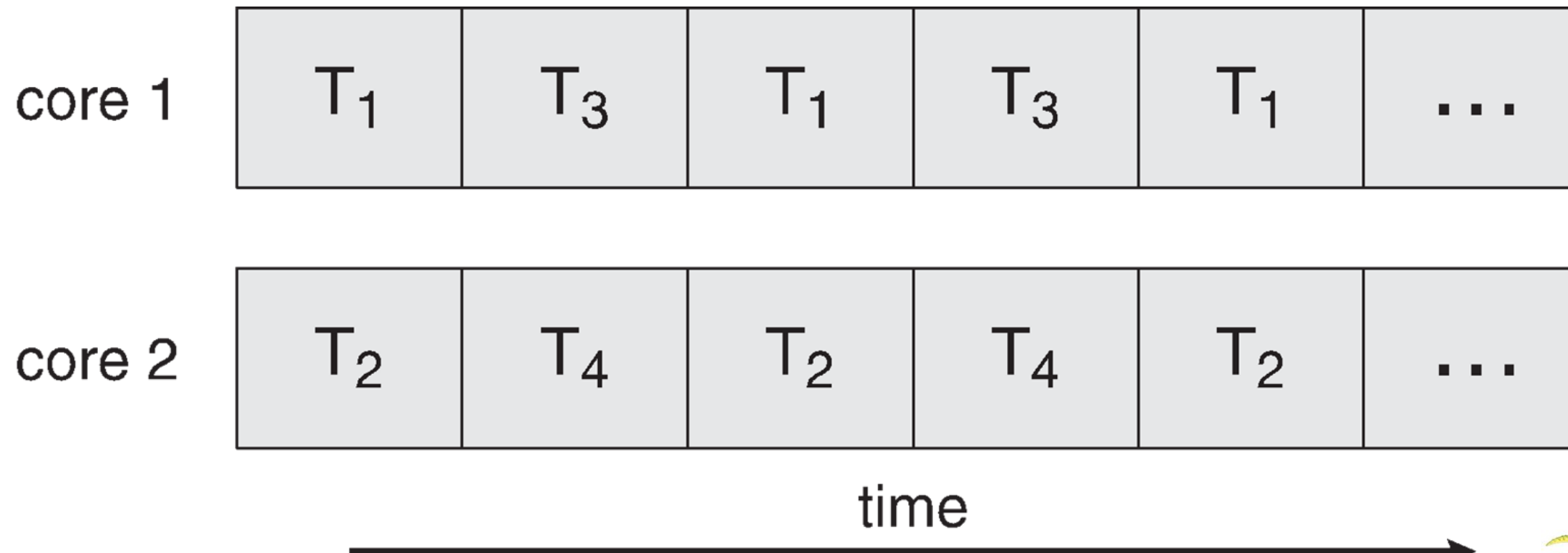


التزامن مقابل التوازي

n التنفيذ المتزامن على نظام واحد الأساسية

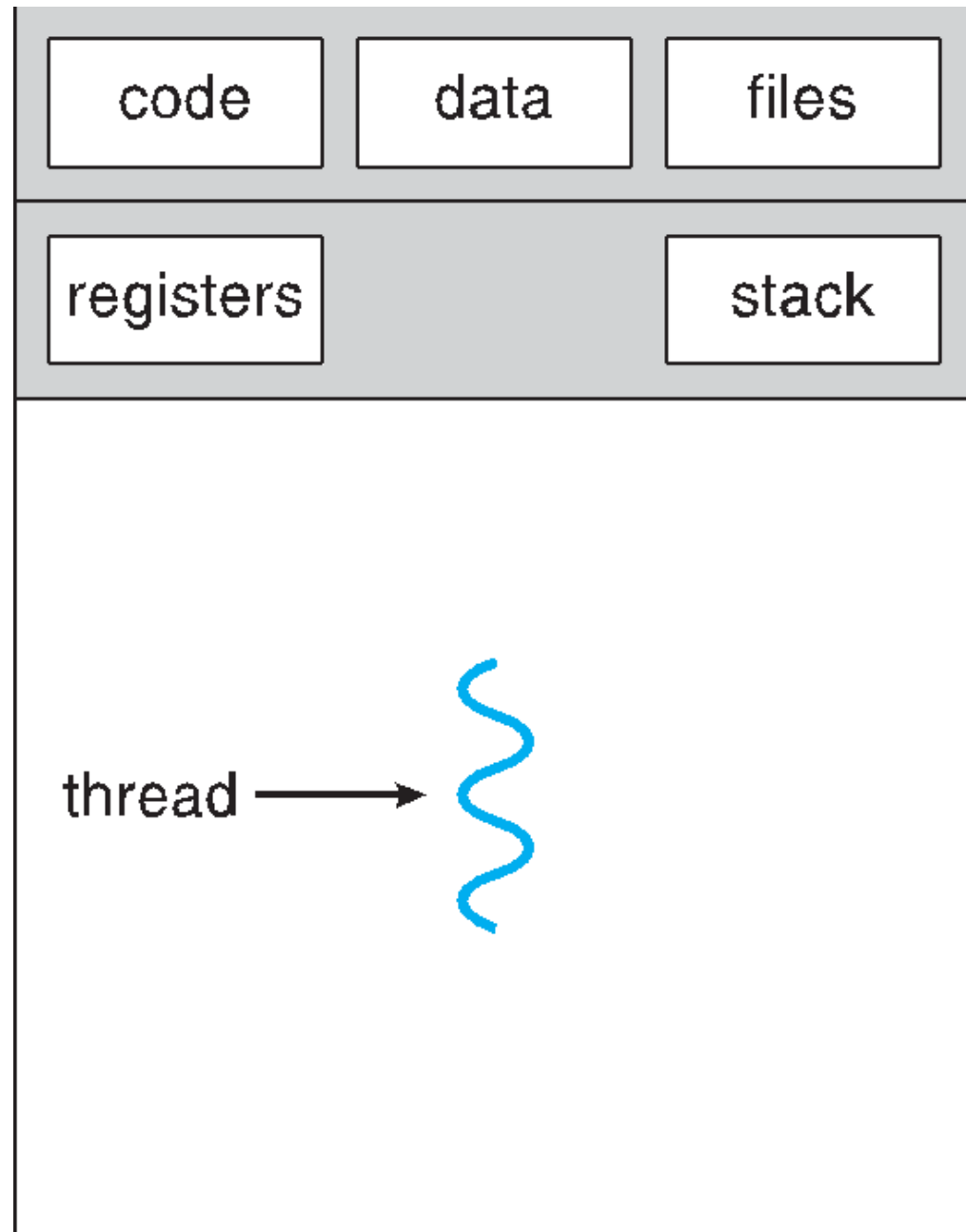


n multi-core: التوازي على نظام

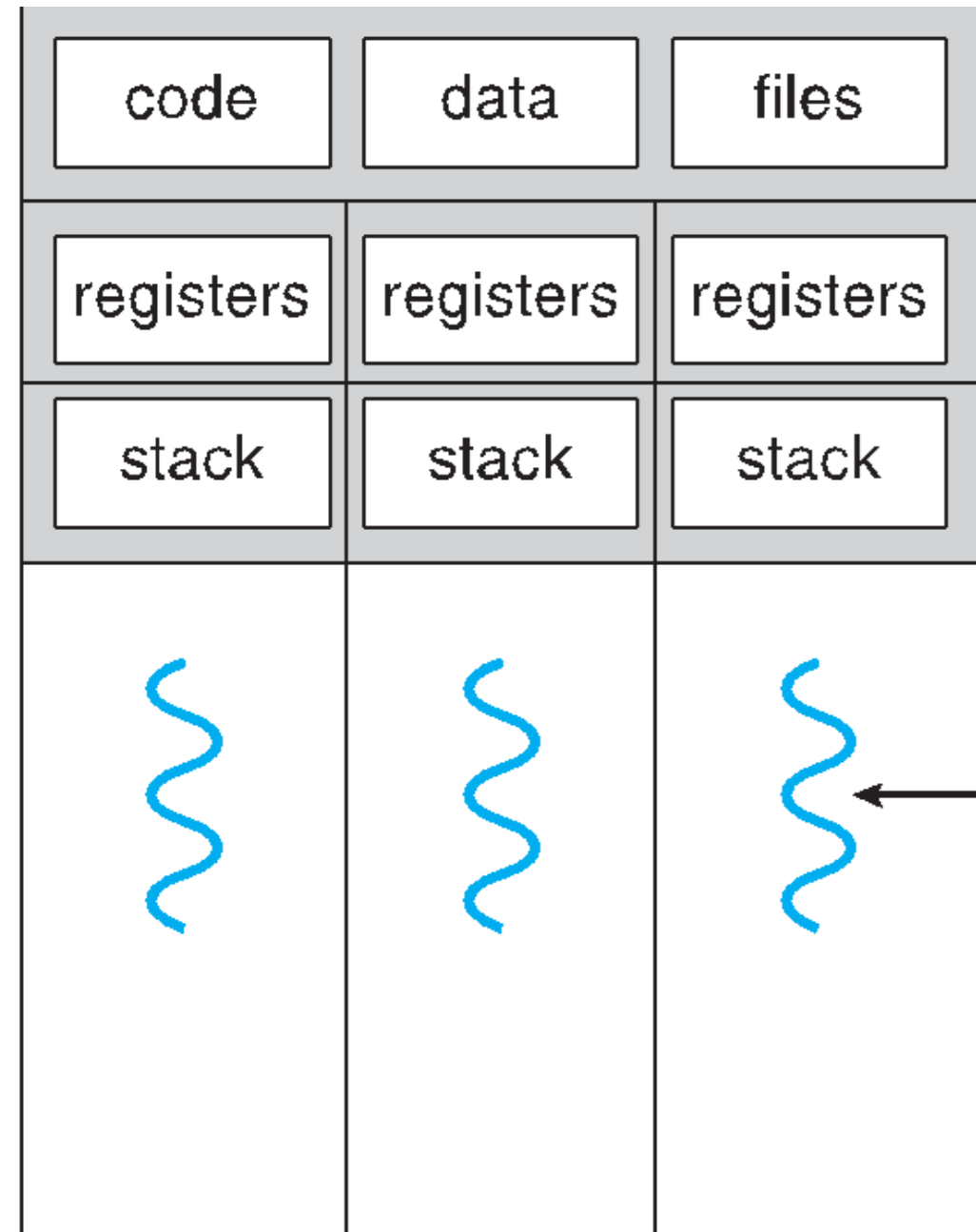




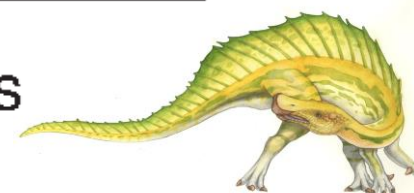
عمليات واحدة ومتعددة مؤشرات الترابط



single-threaded process



multithreaded process





قانون أمدال

- n يحدد المكاسب الأداء من إضافة النوى الإضافية للتطبيق على حد سواء مكونات وبالتوازي مع المسلسل
- n هو جزء التسلسلي S
- n نوى معالجة N

$$speedup \leq \frac{1}{S + \frac{(1-S)}{N}}$$

- n مرة 1.6 النوى النتائج في تسريع من 2-1% المسلسل، والانتقال 25% بالتوازي أي إذا طلب هو
- n $1 / S$ نهج اللانهاية، يقترب تسريع N مثل

جزء التسلسلي للتطبيق ليس له أثر غير متناسب على الأداء اكتسبت عن طريق إضافة النوى الإضافية

- n ولكن لا يأخذ القانون في الاعتبار أنظمة متعددة النوى المعاصرة؟





العضو المواضيع ونواة المواضيع

- n إدارة به مكتبة المواضيع على مستوى المستخدم - **المواضيع المستعمل**
- n ثلاث مكتبات موضوع الابتدائية
 - | **بثريديس** POSIX
 - | و Win32 المواضيع
 - | المواضيع جافا
- n بدعم من النواة - **المواضيع النواة**
- n تقريبا جميع أنظمة غرض التشغيل العامة، بما في ذلك - أمثلة
 - | نوافذ
 - | سولاريس
 - | لينكس
 - | يونيكس Tru64
 - | OS X ماك





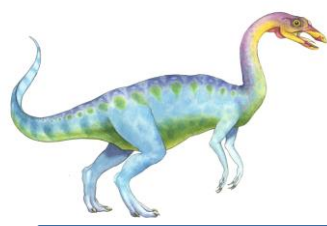
نماذج خاصية تعدد

n العديد مقابل واحد

n واحد لواحد

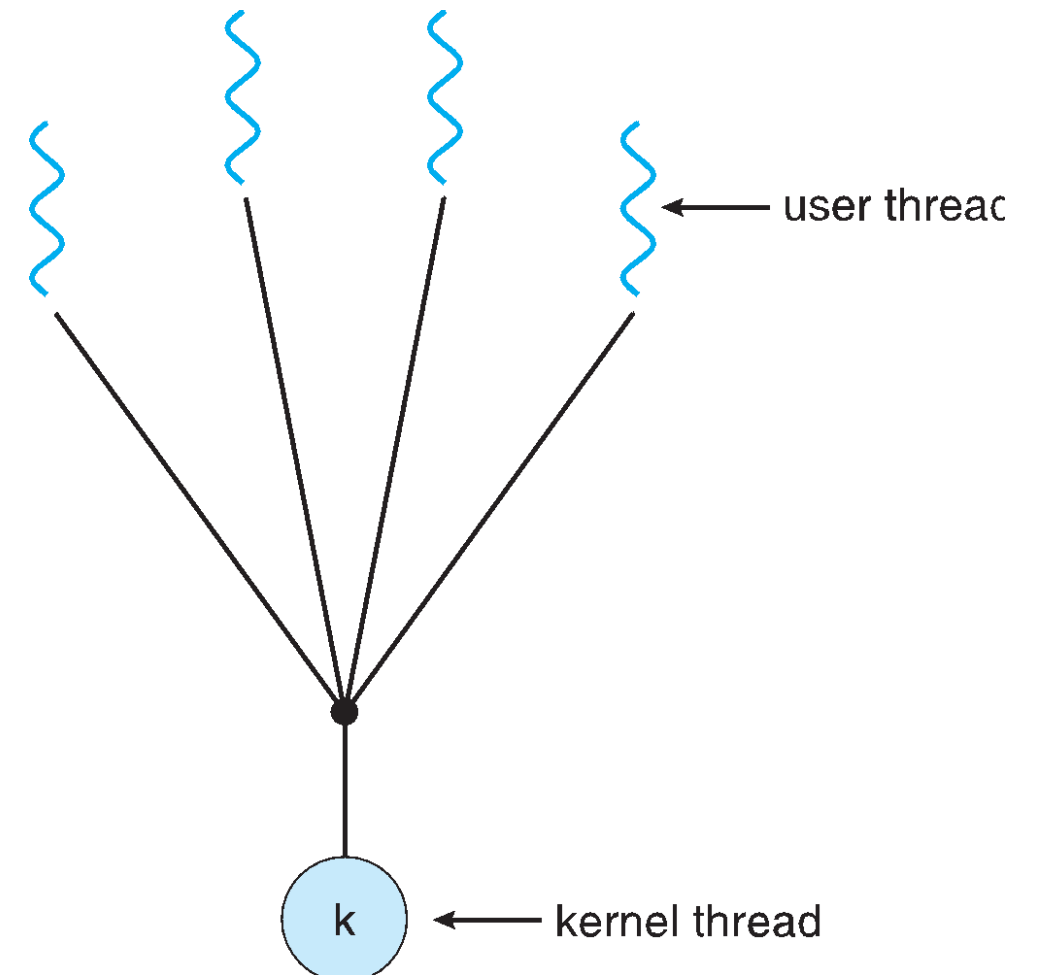
n لكثير-كثير





العديد مقابل واحد

- n العديد من المواضيع على مستوى المستخدم تعيينها إلى موضوع نواة واحدة
- n واحدة حجب موضوع يسبب كل لمنع
- n النظام بسبب واحد فقط قد `muticore` لا تعمل مواضيع متعددة في نفس الوقت على يكون في النواة في وقت
- n استخدام عدد قليل من النظم في هذا النموذج
- n الأمثلة على ذلك:
 - | سولاريس الخضراء المواضيع
 - | جنو المواضيع المحمولة

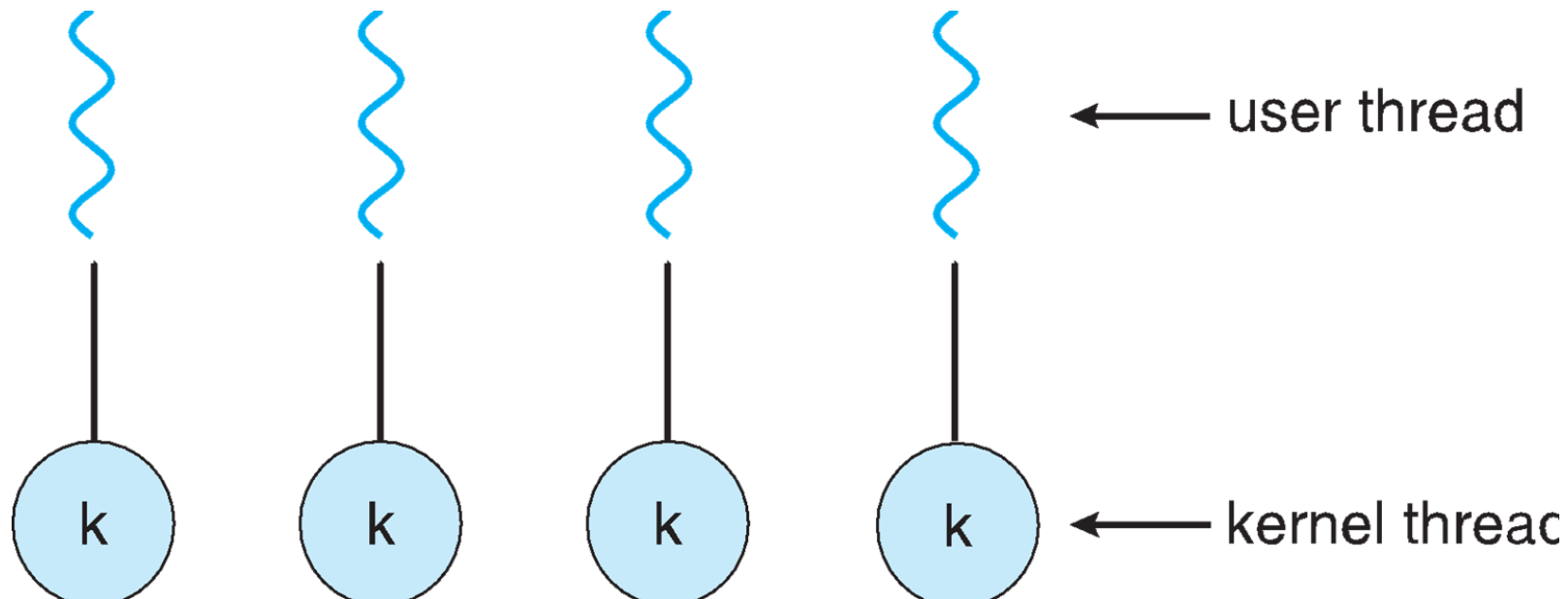




واحد لواحد

- n خرائط كل موضوع على مستوى المستخدم إلى النواة موضوع
- n إنشاء موضوع على مستوى المستخدم يخلق موضوع النواة
- n المزيد من التوافق من كثير، لواحد
- n عدد من المواضيع في عملية تقييد أحيانا بسبب النفقات العامة

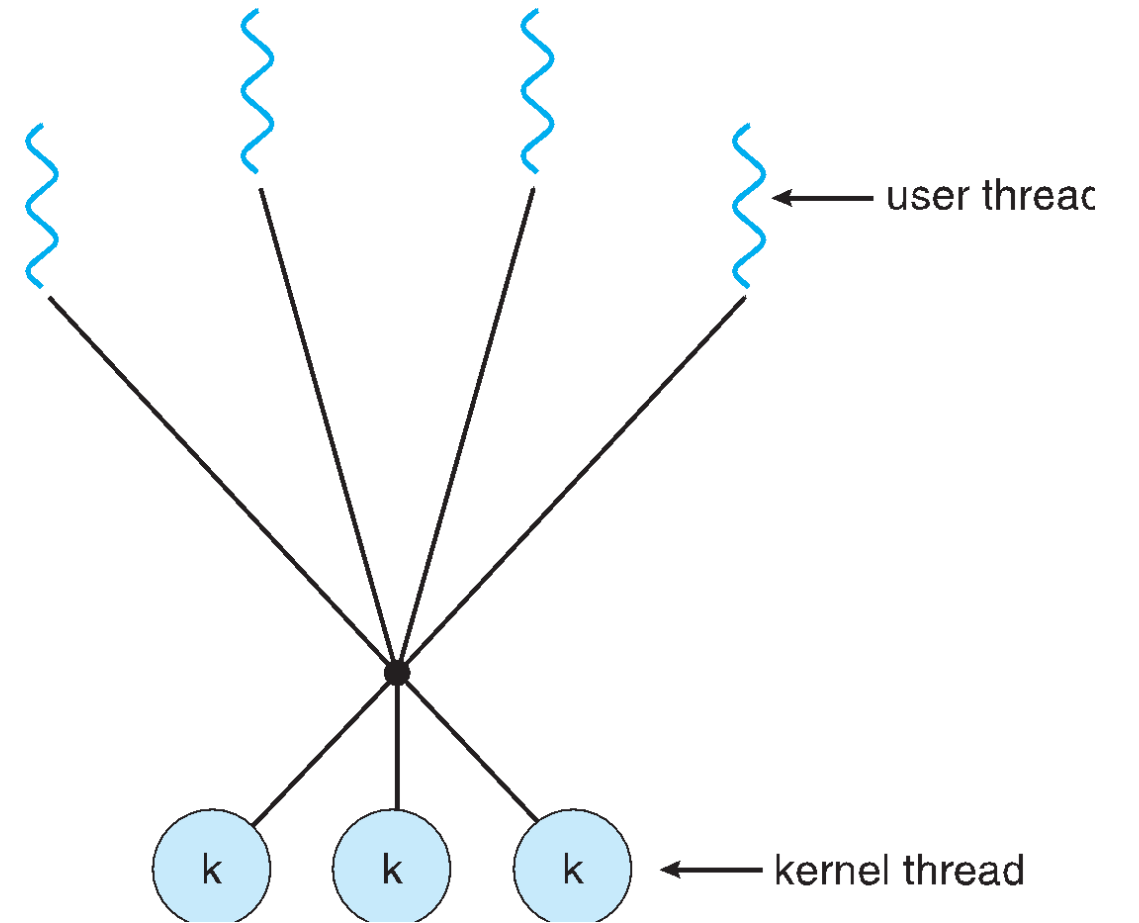
- n أمثلة
 - | ويندوز NT / XP / 2000
 - | لينكس
 - | و في وقت لاحق 9 سولاريس





العديد إلى العديد من نموذج

- n يسمح العديد من المواضيع مستوى المستخدم ليتم تعيينها إلى العديد من المواضيع النواة
- n يسمح نظام التشغيل لإنشاء عدد كاف من المواضيع النواة
- n 9 سولاريس قبل الإصدار
- n صفقة *ThreadFiber* مع NT / 2000 وويندوز



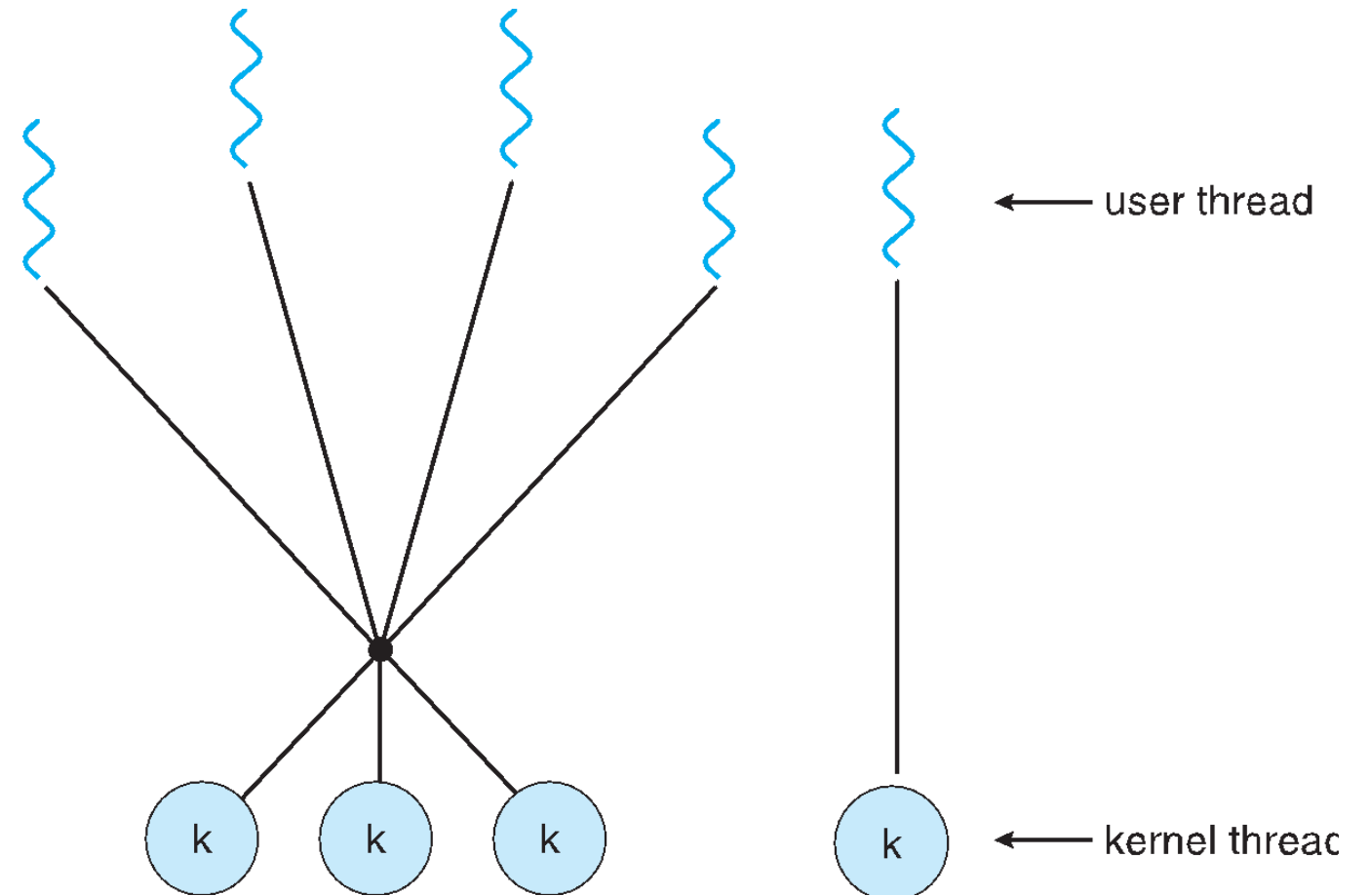


وهما على مستوى نموذج

n ، إلا أنه يسمح للمستخدم أن يكون موضوع مقيد إلى نواة موضوع M: M على غرار

n أمثلة

- | IRIX
- | HP-UX
- | يونيكس Tru64
- | وفي وقت سابق 8 سولاريس





المكتبات الموضوع

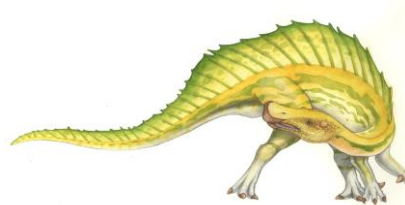
- n لإنشاء وإدارة المواضيع API مكتبة الموضوع يوفر مبرمج مع
- n طريقتين رئيسيتين لتنفيذ
 - | مكتبة تماما في الفضاء المستخدم
 - | مكتبة على مستوى النواة التي يدعمها نظام التشغيل





بثريديس

- n ويمكن توفير إما على مستوى المستخدم أو على مستوى النواة
- n لخلق موضوع وتزامن API (IEEE 1003.1c) معيار POSIX و
- n مواصفة، ليس التنفيذ
- n سلوك مكتبة موضوع والتنفيذ هو ما يصل الى تطوير المكتبة API يحدد
- n (Mac OS X سولاريس، لينكس، نظام التشغيل) مشتركة في يونيكس أنظمة التشغيل





بثريديس مثال

```
#include <pthread.h>
#include <stdio.h>

int sum; /* this data is shared by the thread(s) */
void *runner(void *param); /* threads call this function */

int main(int argc, char *argv[])
{
    pthread_t tid; /* the thread identifier */
    pthread_attr_t attr; /* set of thread attributes */

    if (argc != 2) {
        fprintf(stderr, "usage: a.out <integer value>\n");
        return -1;
    }
    if (atoi(argv[1]) < 0) {
        fprintf(stderr, "%d must be >= 0\n", atoi(argv[1]));
        return -1;
    }
}
```





(يتبع) بثريديس مثال

```
/* get the default attributes */
pthread_attr_init(&attr);
/* create the thread */
pthread_create(&tid,&attr,runner,argv[1]);
/* wait for the thread to exit */
pthread_join(tid,NULL);

printf("sum = %d\n",sum);
}

/* The thread will begin control in this function */
void *runner(void *param)
{
    int i, upper = atoi(param);
    sum = 0;

    for (i = 1; i <= upper; i++)
        sum += i;

    pthread_exit(0);
}
```

Figure 4.9 Multithreaded C program using the Pthreads API.





المواضيع 10 بثريديس مدونة الانضمام

```
#define NUM_THREADS 10

/* an array of threads to be joined upon */
pthread_t workers[NUM_THREADS];

for (int i = 0; i < NUM_THREADS; i++)
    pthread_join(workers[i], NULL);
```

Figure 4.10 Pthread code for joining ten threads.





C متعددة مؤشرات الترابط Win32 API برنامج

```
#include <windows.h>
#include <stdio.h>
DWORD Sum; /* data is shared by the thread(s) */

/* the thread runs in this separate function */
DWORD WINAPI Summation(LPVOID Param)
{
    DWORD Upper = *(DWORD*)Param;
    for (DWORD i = 0; i <= Upper; i++)
        Sum += i;
    return 0;
}

int main(int argc, char *argv[])
{
    DWORD ThreadId;
    HANDLE ThreadHandle;
    int Param;

    if (argc != 2) {
        fprintf(stderr, "An integer parameter is required\n");
        return -1;
    }
    Param = atoi(argv[1]);
    if (Param < 0) {
        fprintf(stderr, "An integer >= 0 is required\n");
        return -1;
    }
}
```





Win32 API وبرنامج (يتبع) C متعددة مؤشرات الترابط

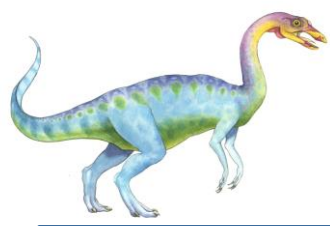
```
/* create the thread */
ThreadHandle = CreateThread(
    NULL, /* default security attributes */
    0, /* default stack size */
    Summation, /* thread function */
    &Param, /* parameter to thread function */
    0, /* default creation flags */
    &ThreadId); /* returns the thread identifier */

if (ThreadHandle != NULL) {
    /* now wait for the thread to finish */
    WaitForSingleObject(ThreadHandle, INFINITE);

    /* close the thread handle */
    CloseHandle(ThreadHandle);

    printf("sum = %d\n", Sum);
}
}
```





جافا المواضيع

- n JVM تدار المواضيع جافا من قبل
- n تنفذ عادة باستخدام نموذج عن المواضيع التي يقدمها نظام التشغيل الأساسي
- n يمكن إنشاء المواضيع جافا من قبل:
 - | توسيع الطبقة الموضوع
 - | Runnable تنفيذ واجهة

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





برنامج جافا متعددة مؤشرات الترابط

```
class Sum
{
    private int sum;

    public int getSum() {
        return sum;
    }

    public void setSum(int sum) {
        this.sum = sum;
    }
}

class Summation implements Runnable
{
    private int upper;
    private Sum sumValue;

    public Summation(int upper, Sum sumValue) {
        this.upper = upper;
        this.sumValue = sumValue;
    }

    public void run() {
        int sum = 0;
        for (int i = 0; i <= upper; i++)
            sum += i;
        sumValue.setSum(sum);
    }
}
```





(يتبع) برنامج متعددة مؤشرات الترابط جافا

```
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 {
                Sum sumObject = new Sum();
                int upper = Integer.parseInt(args[0]);
                Thread thrd = new Thread(new Summation(upper, sumObject));
                thrd.start();
                try {
                    thrd.join();
                    System.out.println
                        ("The sum of "+upper+" is "+sumObject.getSum());
                } catch (InterruptedException ie) { }
            }
        }
        else
            System.err.println("Usage: Summation <integer value>"); }
}
```





ضمني التخييط

- n تزايد شعبية كأرقام من المواضيع الزيادة، صحة البرنامج أكثر صعوبة مع المواضيع صريحة
- n إنشاء وإدارة المواضيع قام به المجمعين وتشغيل مكتبات وقت بدلا من المبرمجين
- n ثلاث طرق استكشاف
 - | حمامات الموضوع
 - | قانون الزواج
 - | جراند إيفاد الوسطى
- n صفة `java.util.concurrent` (TBB) وتشمل الأساليب الأخرى مايكروسوفت خيوط بناء كتل



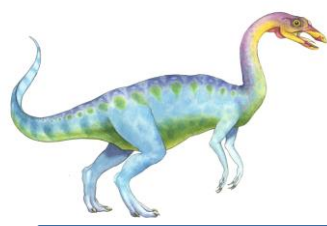


حمامات الموضوع

- n إنشاء عدد من المواضيع في بركة حيث ينتظرون العمل
- n مزايا:
 - | عادة ما تكون أسرع قليلا لخدمة الطلب مع موضوع موجود من إنشاء موضوع جديد
 - | لتكون ملزمة لحجم التجمع (ق) يسمح عدد من المواضيع في التطبيق
 - | فصل المهمة التي يتعين القيام بها من آليات خلق مهمة تسمح استراتيجيات مختلفة لإدارة مهمة
 - لتشغيل بشكل دوري `ieTasks` يمكن المقرر 4
- n يدعم حمامات الموضوع API ويندوز

```
DWORD WINAPI PoolFunction(AVOID Param) {  
    /*  
    * this function runs as a separate thread.  
    */  
}
```





قانون الزواج

- n ، ++C ، C API مجموعة من التوجيهات مترجم و FORTRAN
- n يقدم الدعم للبرمجة المتوازية في بيئات الذاكرة المشتركة
- n كتل من التعليمات البرمجية التي -يحدد المناطق الموازية يمكن تشغيلها في نفس الوقت

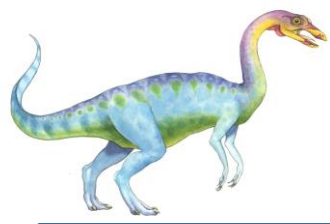
المرصد المغربي للسجون مواز `#pragma`
إنشاء العديد من المواضيع كما أن هناك النوى

```
#pragma المرصد المغربي للسجون  
(++؛ ط <N ؛ ط = 0 ؛ بالتوازي عن  
ج . [أ] + [ط] = [ط] ج  
}
```

الترشح للحلقة بالتوازي

```
#include <omp.h>  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    /* sequential code */  
  
    #pragma omp parallel  
    {  
        printf("I am a parallel region.");  
    }  
  
    /* sequential code */  
  
    return 0;  
}
```



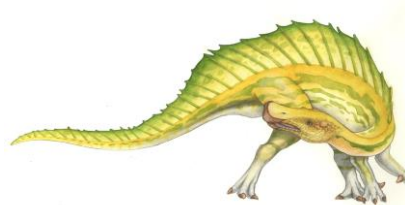


جراند إيفاد الوسطى

- n ودائرة الرقابة الداخلية Mac OS X التكنولوجيا أبل لأنظمة التشغيل
- n ، ومكتبة وقت تشغيل API لغة، C ++ ، C ملحقات ل
- n يسمح تحديد أقسام موازية
- n تدير معظم التفاصيل من خيوط
- n ؛ { printf ("أنا كتلة " - { } ^ "كتلة في
- n كتل وضعها في قائمة الانتظار إرسال
 - | المخصصة لموضوع متاح في ترابط التجمع عند إزالتها من قائمة الانتظار
- n :نوعين من طوابير ارسال
 - | ، الطابور هو في العملية، ودعا **طابور الرئيسي** FIFO بنات إزالتها من أجل -مسلسل
 - 4 يمكن للمبرمجين خلق طوابير مسلسل إضافية ضمن برنامج
 - | ولكن العديد يمكن إزالتها في وقت FIFO إزالتها من أجل -المتزامنة
 - 4 طوابير ثلاثة نظام اسعة مع أولويات منخفضة، والتقصير، وارتفاع

```
dispatch_queue_t queue = dispatch_get_global_queue  
(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
```

```
dispatch_async(queue, ^{ printf("I am a block."); });
```





قضايا خيوط

- n استدعاءات النظام () وإسبك () دلالات شوكة
- n معالجة الإشارة
 - | المتزامن وغير المتزامن
- n إلغاء الموضوع من موضوع الهدف
 - | غير المتزامن أو تأجيلها
- n التخزين الموضوع المحلية
- n جدولة التنشيط





()exec والذي () دلالات شوكة

- n تكرار فقط موضوع الدعوة أو كل المواضيع؟ () هل شوكة
 - | يكون نسختين من شوكة Unixes بعض من
- n استبدال عملية التشغيل بما في ذلك جميع المواضيع - عادة ما يعمل كالمعتاد () إكسيك

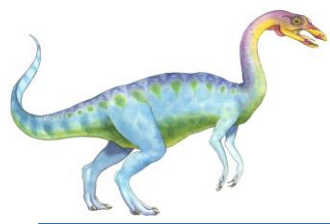




التعامل مع إشارة

- n إشارات وتستخدم في أنظمة يونيكس لإعلام عملية وقوع حدث معين
- n **معالج إشارة** يستخدم لمعالجة الإشارات
 - 1. يتم إنشاء إشارة من حدث معين
 - 2. يتم تسليم إشارة إلى عملية
 - 3. تتم معالجة إشارة من جانب واحد من اثنين من معالجات الإشارات
 - 1. الافتراضي
 - 2. تعريف المستخدم
- n كل إشارة لها **معالج الافتراضي** هذه الإشارة أشواط نواة عند التعامل مع
 - | **معالج إشارة المعرفة** يمكن تجاوز الافتراضي
 - | للترابط واحد، سلمت إشارة إلى معالجة
- n حيث ينبغي تسليم إشارة لمتعددة الخيوط؟
 - | تقديم إشارة إلى موضوع التي تنطبق عليها إشارة
 - | تقديم إشارة إلى كل موضوع في عملية
 - | تقديم إشارة إلى بعض المواضيع في عملية
 - | تعيين موضوع محدد لاستقبال جميع إشارات لعملية





إلغاء الموضوع

- n إنهاء الموضوع قبل انتهائه
- n الموضوع تلغى هو **موضوع الهدف**
- n نهجين العامة:
 - | إلغاء غير متزامن إنهاء موضوع الهدف فوراً
 - | إلغاء المؤجلة يسمح للموضوع المستهدف للتحقق بشكل دوري إذا كان ينبغي إلغاء
- n لإنشاء وإلغاء الموضوع Pthread كود:

```
pthread_t tid;  
  
/* create the thread */  
pthread_create(&tid, 0, worker, NULL);  
  
. . .  
  
/* cancel the thread */  
pthread_cancel(tid);
```





(يتبع) كاتب الموضوع إلغاء

n يطلب استدعاء إلغاء موضوع إلغاء، ولكن يعتمد على إلغاء الفعلي على الدولة موضوع

Mode	State	Type
Off	Disabled	–
Deferred	Enabled	Deferred
Asynchronous	Enabled	Asynchronous

n إذا موضوع تم تعطيل الإلغاء، لا يزال إلغاء انتظار حتى تمكن موضوع كان

n يتم تأجيل النوع الافتراضي

إلغاء يحدث فقط عندما يصل الموضوع **نقطة إلغاء**

4 `pthread_testcancel ()` أي

4 ثم **معالج تنظيف** تم استدعاء

n على أنظمة لينكس، ويتم التعامل مع إلغاء الموضوع من خلال إشارات

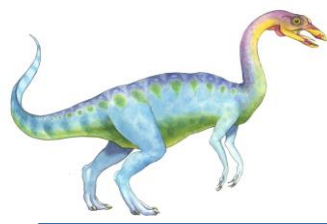




الخيط المحلي التخزين

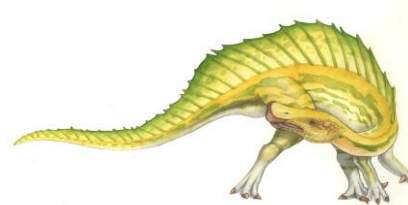
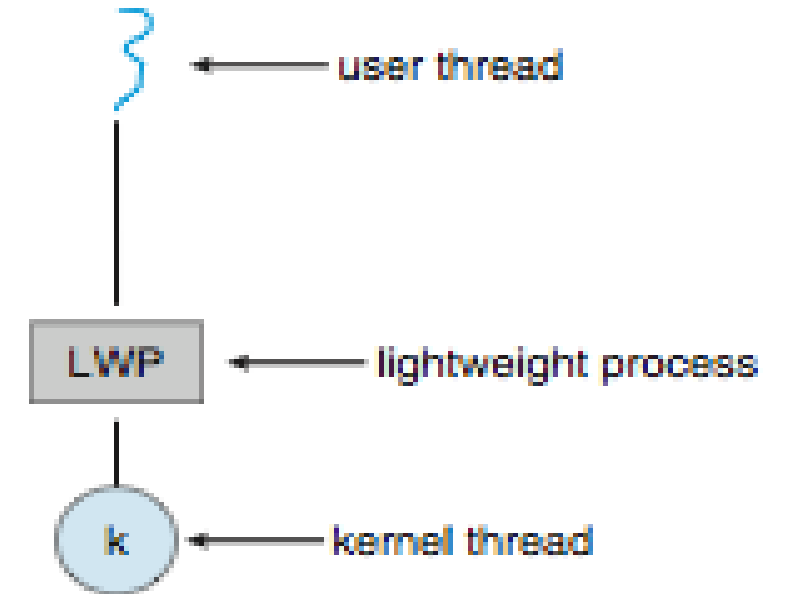
- n يسمح لكل موضوع لنسخته من البيانات (TLS) التخزين الموضوع المحلية
- n (أي عند استخدام تجمع مؤشرات ترابط) من المفيد عندما لم يكن لديك السيطرة على عملية إنشاء موضوع
- n تختلف عن المتغيرات المحلية
 - | المتغيرات المحلية مرئية فقط خلال وظيفة الاحتجاج واحد
 - | مرئيا عبر الدعاء وظيفة TLS
- n على غرار ساكن البيانات
 - | هي فريدة من نوعها في كل موضوع TLS

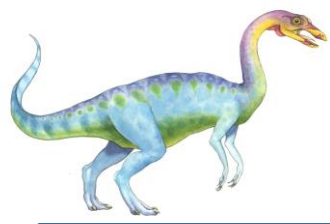




جدولة التنشيط

- n وعلى مستوى نموذجين الاتصالات للحفاظ على عدد مناسب من المواضيع نواة M تتطلب M كلاً: المخصصة للتطبيق
- n (LWP) عملية وخفيفة الوزن - عادة ما تستخدم بنية بيانات وسيط بين المستخدم ونواة المواضيع
 - | يبدو أن المعالج الظاهري على العملية التي يمكن أن الجدول الزمني موضوع المستخدم لتشغيل
 - | تعلق على نواة موضوع LWP كل
 - | لخلق؟ LWP كم عدد
- n في مكتبة موضوع upcall آلية التواصل من نواة ل معالج - upcalls توفر التنشيط جدول
- n هذا التواصل يتيح تطبيق للحفاظ على المواضيع نواة الرقم الصحيح

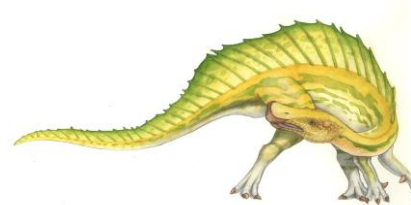




أمثلة نظام التشغيل

n ويندوز إكس بي المواضيع

n لينكس الموضوع

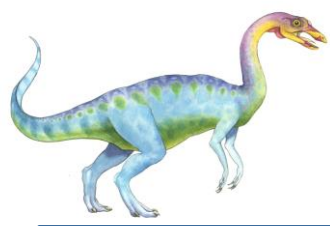




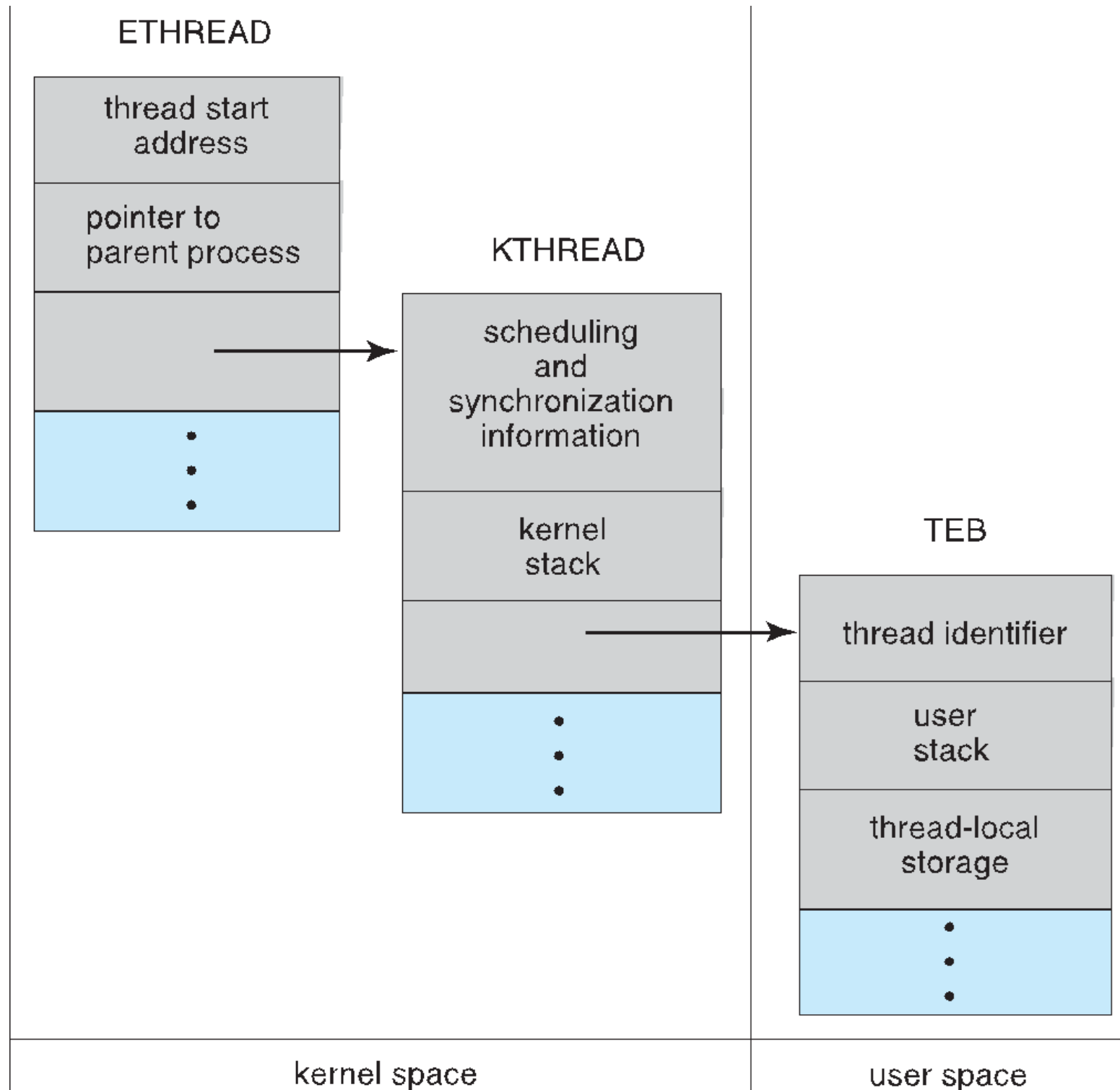
نوافذ المواضيع

- n 7، ويندوز XP، 2000، وين NT، وين 98 الرئيسي لفوز API - API تنفذ نوافذ ويندوز
- n تطبق الخرائط واحد واحد الى و، على مستوى النواة
- n يحتوي كل موضوع
 - | معرف موضوع
 - | تسجيل وضع دولة تمثل المعالج
 - | kernel منفصلة المستخدم ونواة المداخن لأنه عندما يعمل موضوع في وضع المستخدم أو وضع (DLL) خاصة منطقة تخزين البيانات المستخدمة من قبل مكتبات وقت التشغيل ومكتبات الارتباط الحيوي
- n مجموعة السجل، المداخن، ومنطقة تخزين خاصة تعرف باسم **سياق الكلام** في الموضوع
- n بنيات البيانات الأساسية لموضوع ما يلي
 - | ، في الفضاء النواة KTHREAD يشمل المؤشر إلى معالجة التي ينتمي موضوع و - (كتلة موضوع التنفيذي) ETHREAD
 - | ، في الفضاء النواة TEB كومة، مؤشر إلى kernel جدول وتزامن المعلومات، وضع - (نواة كتلة موضوع) KTHREAD
 - | موضوع الهوية، ووضع المستخدم كومة، التخزين موضوع المحلي، في الفضاء المستخدم - (كتلة بيئة موضوع) TEB





الهياكل ويندوز إكس بي المواضيع البيانات





لينكس المواضيع

- n يشير لينكس لهم كما المهام بدلا من المواضيع
- n استدعاء النظام () يتم إنشاء الموضوع من خلال استنساخ
- n (عملية)يسمح مهمة طفل للمشاركة في مساحة العنوان من مهمة الأم () استنساخ
أعلام ضبط السلوك |

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.

- n (مشتركة أو فريدة من نوعها)نقاط لمعالجة هياكل البيانات `task_struct` البنية

