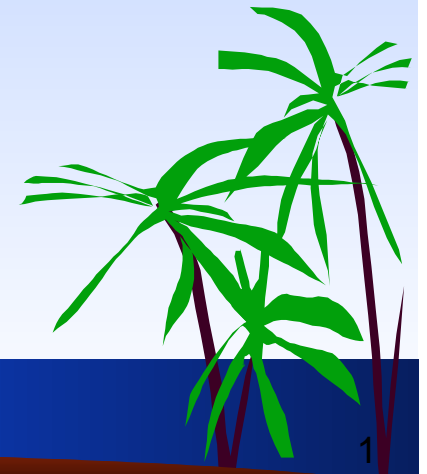
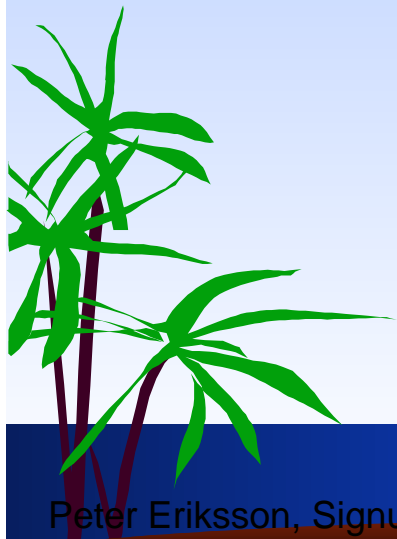


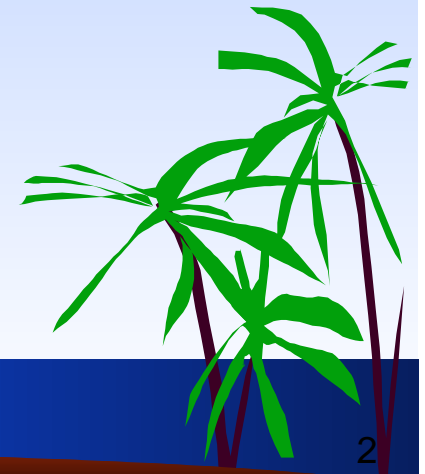
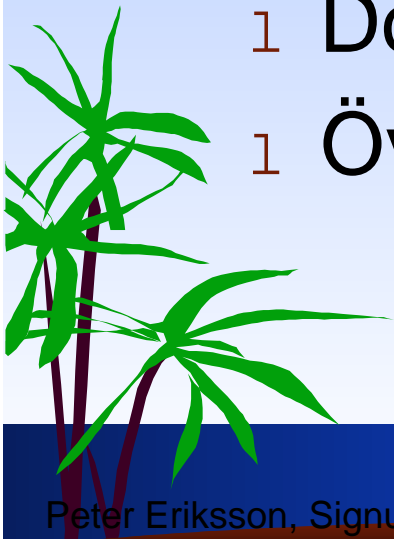
# Posix Threads

En studie i förvirring  
eller  
Halleluja!



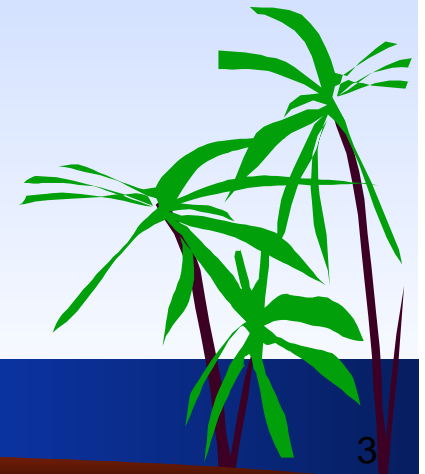
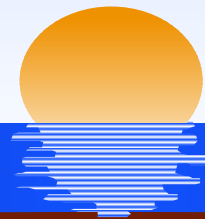
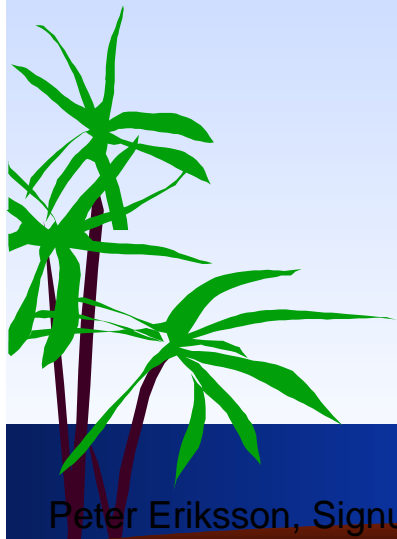
# Innehåll

- 1 Översikt
- 1 Programmering
- 1 Kompilering
- 1 Vanliga problem
- 1 Dokumentation
- 1 Övrigt



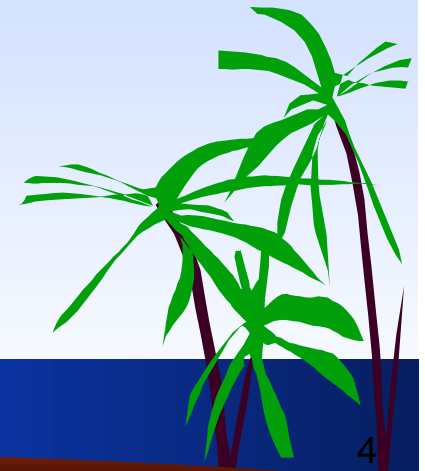
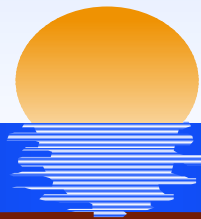
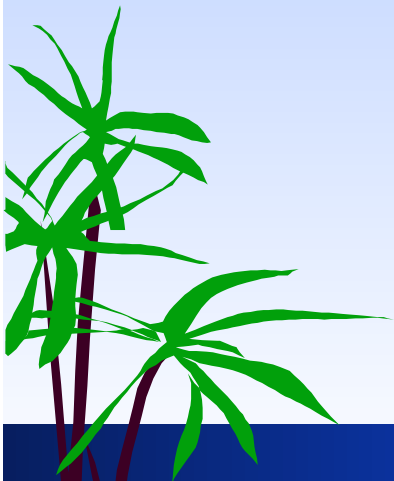
# Översikt

- 1 Introduktion
- 1 Implementationer
- 1 Operativsystem
- 1 Prestanda



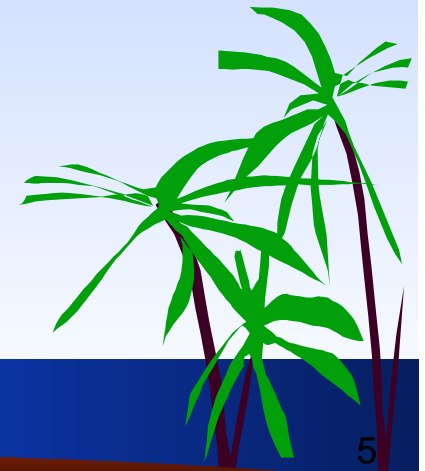
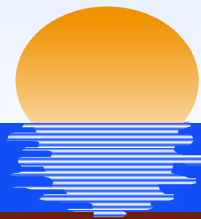
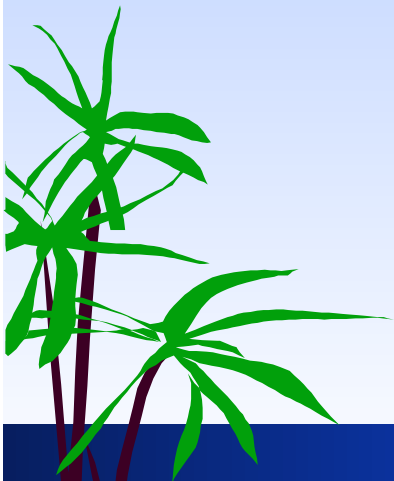
# Introduktion

- 1 Vad är en tråd?
- 1 Varför tråda ett program?
- 1 Fördelar
- 1 Nackdelar



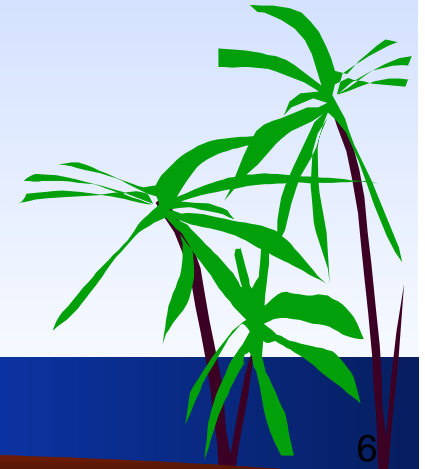
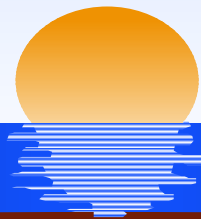
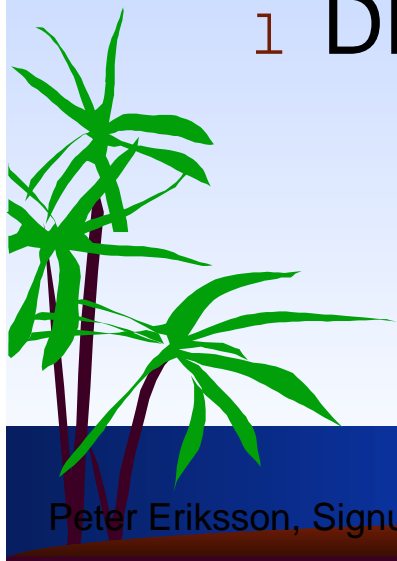
# Implementationer

- 1 Pthreads (POSIX P1003.1c)
- 1 DCE Threads
- 1 UI Threads
- 1 Windows NT



# Operativsystem

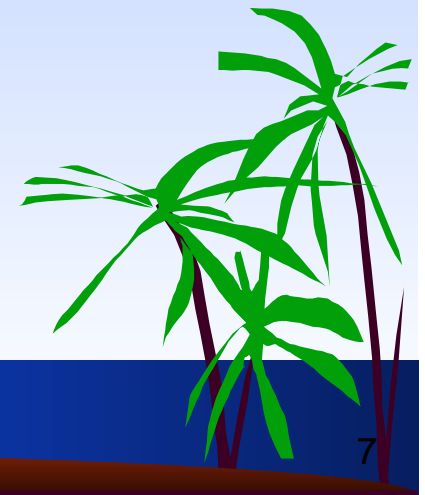
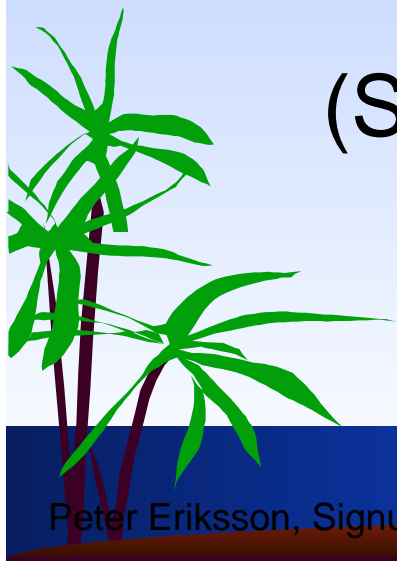
- 1 Sun Solaris 2.5
- 1 Redhat Linux 5.0
- 1 SGI IRIX 6.3
- 1 HP HP-UX 10.30
- 1 DEC DigitalUnix 4.0



# Prestanda

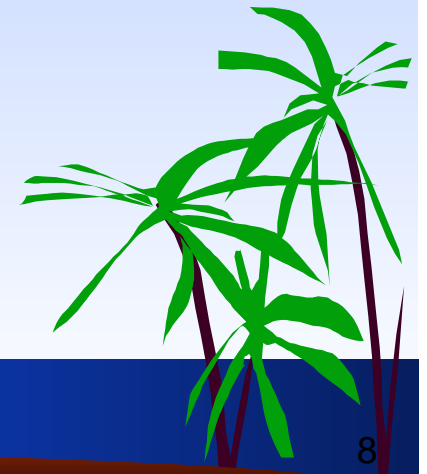
- 1 Skapa en tråd:  $\sim 52\mu\text{s}$
- 1 Skapa en process:  $\sim 1700\mu\text{s}$
- 1 Trådsynkronisering:  $\sim 66\mu\text{s}$
- 1 Processsynkronisering:  $\sim 200\mu\text{s}$

(SPARCstation 2, Solaris 2.4)



# Programmering

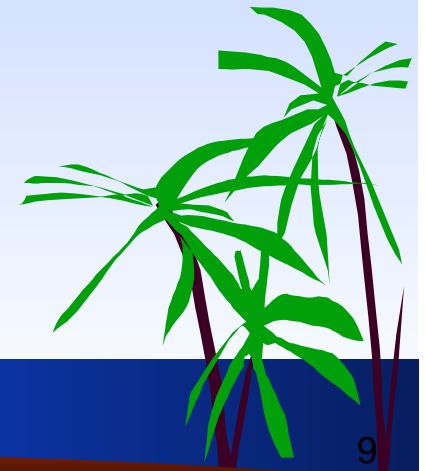
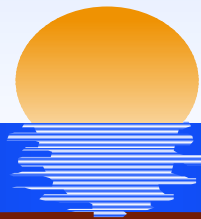
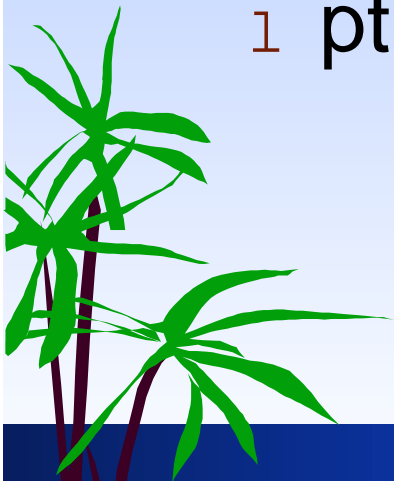
- 1 Att skapa trådar
- 1 Synkronisering
- 1 Biblioteksstöd
- 1 Signalhantering
- 1 TSD (trådlokala globala variabler)
- 1 Avancerad programmering





# Att skapa trädar

- 1 pthread\_create()
- 1 pthread\_exit()
- 1 pthread\_join()
- 1 pthread\_self()
- 1 pthread\_equal()



# Exempel

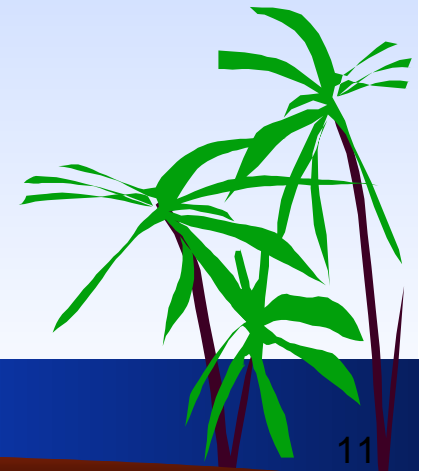
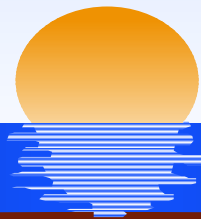
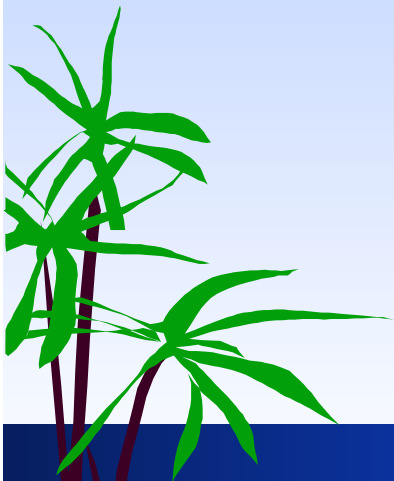
```
void *my_thread(void *arg) {
    puts(arg);
    return NULL;
}

int main(int argc, char *argv[]) {
    pthread_t tid;
    void *retval;

    pthread_create(&tid, NULL,
                  my_thread, "hello world");
    pthread_join(tid, &retval);
}
```

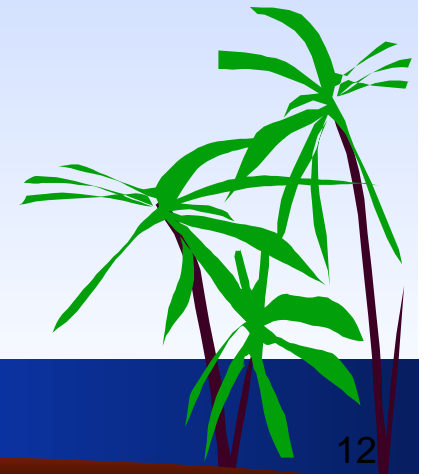
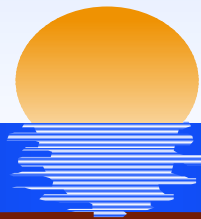
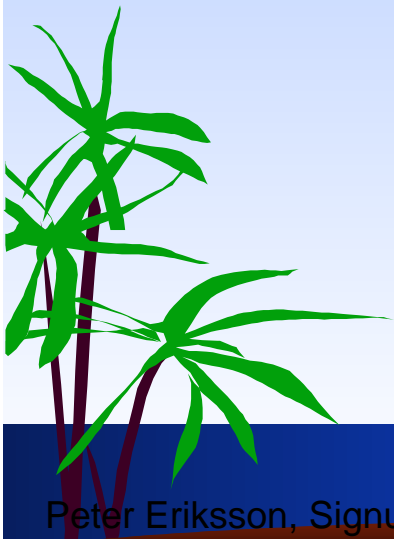
# Synkronisering

- 1 mutex
- 1 condition variables
- 1 Initialiseringskontroll
- 1 Är det allt?



# MUTual EXclusion Locks

- 1 `pthread_mutex_init()`
- 1 `pthread_mutex_lock()`
- 1 `pthread_mutex_unlock()`
- 1 `pthread_mutex_trylock()`

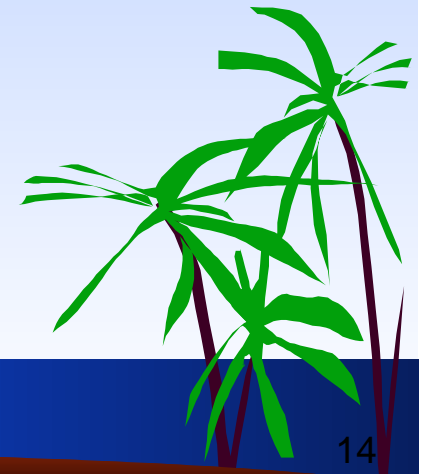
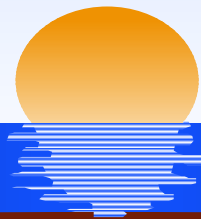
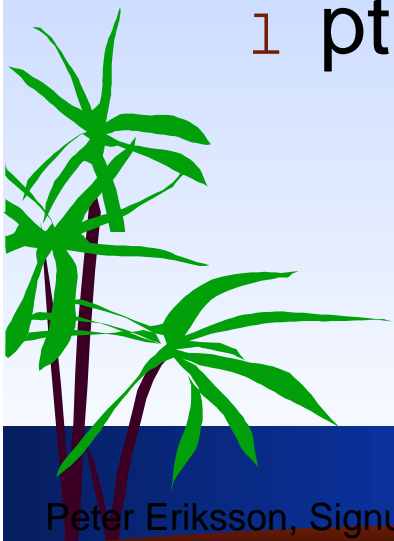


# Exempel

```
pthread_mutex_t  mtx;  
int              cnt = 0;  
  
int add_val(int val) {  
    int sum;  
    pthread_mutex_lock(&mtx);  
    sum = cnt += val;  
    pthread_mutex_unlock(&mtx);  
    return sum;  
}
```

# Condition variables

- 1 `pthread_cond_init()`
- 1 `pthread_cond_wait()`
- 1 `pthread_cond_timedwait()`
- 1 `pthread_cond_signal()`
- 1 `pthread_cond_broadcast()`

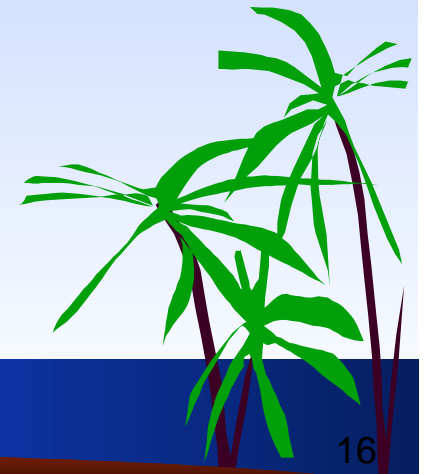
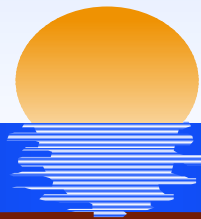
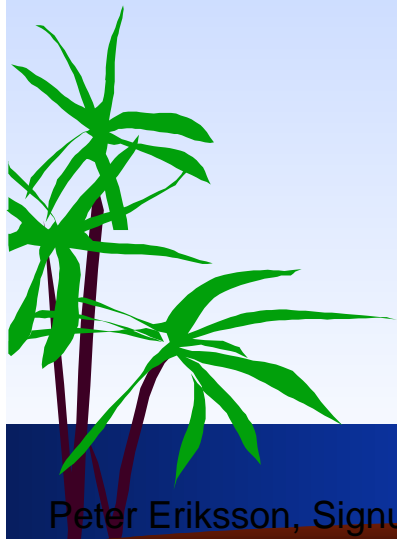


# Exempel

```
pthread_mutex_t    mtx;  
pthread_cond_t     cv;  
int active = 0;  
int max = 10;  
  
void test_init(void) {  
    pthread_mutex_init(&mtx, NULL);  
    pthread_cond_init(&cv, NULL);  
}
```

# Exempel, del 2

```
void *my_thread(void *arg) {  
    ...  
    pthread_mutex_lock(&mtx);  
    if (active == max)  
        pthread_cond_signal(&cv);  
    --active;  
    pthread_mutex_unlock(&mtx);  
    return NULL;  
}
```



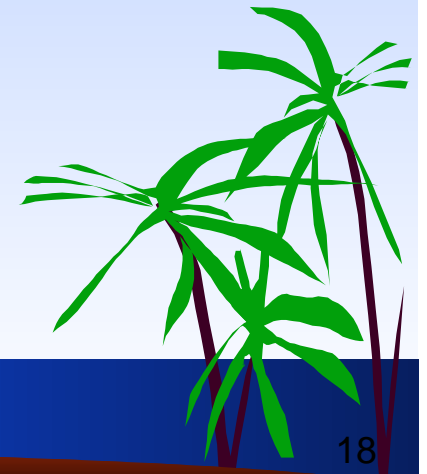
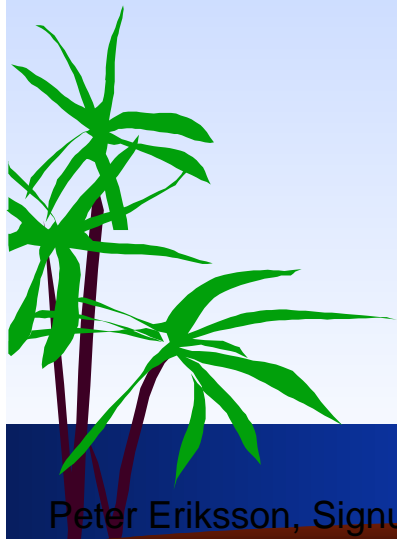


# Exempel, del 3

```
void start_new_thread(void) {  
    pthread_t tid;  
  
    pthread_mutex_lock(&mtx);  
    while (active >= max)  
        pthread_cond_wait(&cv, &mtx);  
    active++;  
    pthread_mutex_unlock(&mtx);  
    pthread_create(&tid, NULL, my_thread, NULL);  
}
```

# Initialiseringskontroll

1 pthread\_once()



# Exempel

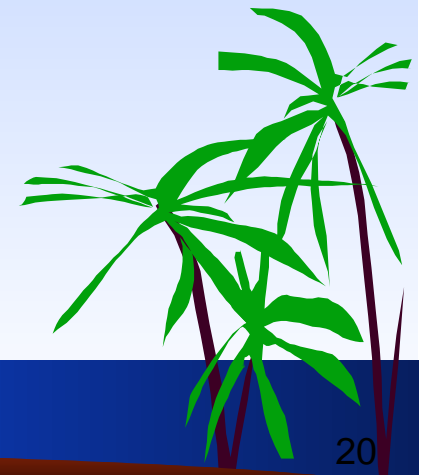
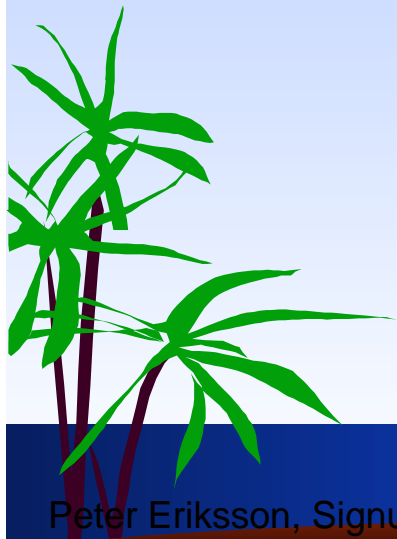
```
pthread_once_t once_cntrl = PTHREAD_ONCE_INIT;

static void my_init(void) {
    ... initialize something ...
}

int my_function(int arg) {
    pthread_once(&once_cntrl, my_init);
    ... do something ...
}
```

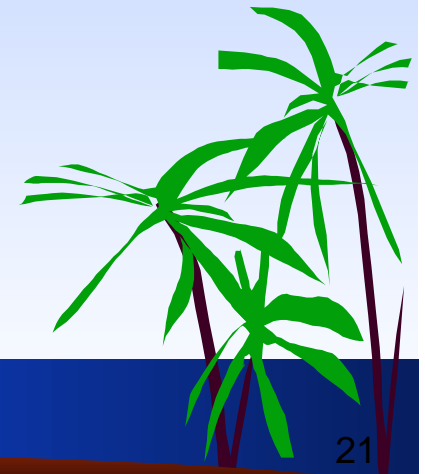
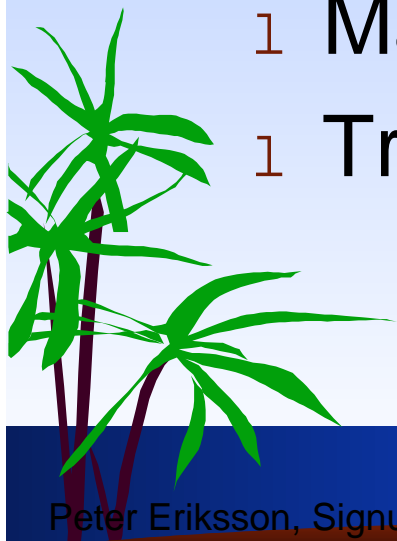
# Är det allt?

- 1 rwlocks
- 1 semaforer
- 1 spinlocks



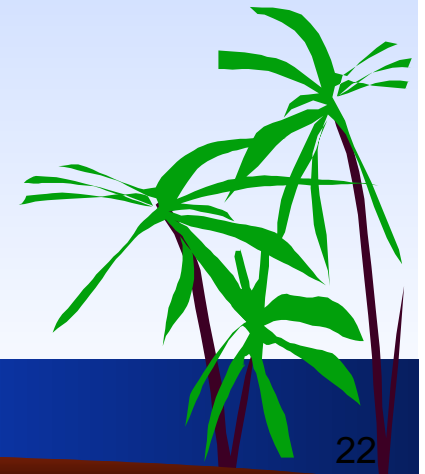
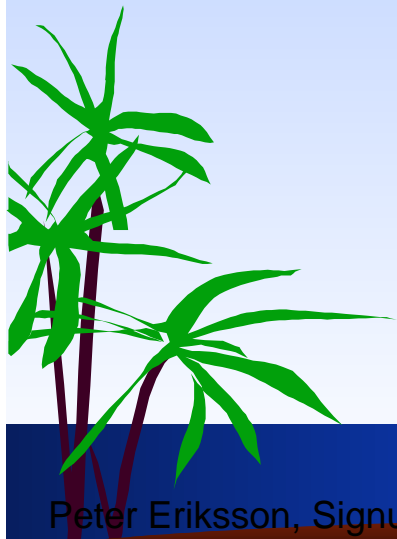
# Biblioteksstöd

- 1 Systemanrop
- 1 errno
- 1 fork() / pthread\_atfork()
- 1 Trådsäkra funktioner
- 1 Manuellt låsbara funktioner
- 1 Trådosäkra funktioner



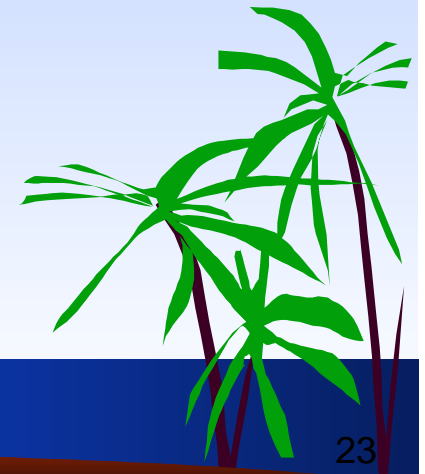
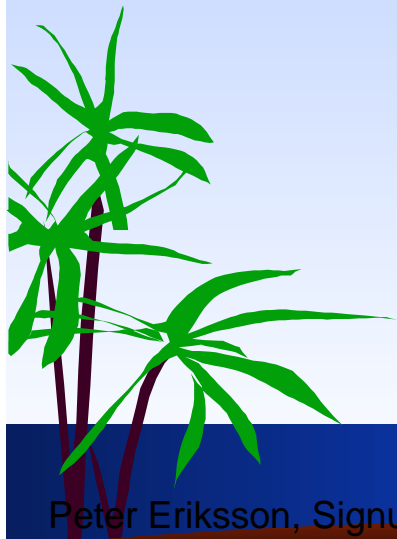
# Systemanrop

- 1 Trådsäkra
- 1 Blockar enbart tråden



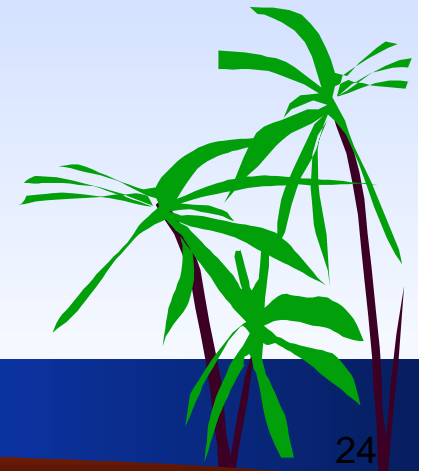
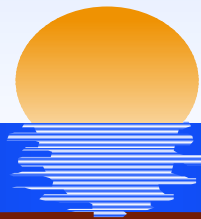
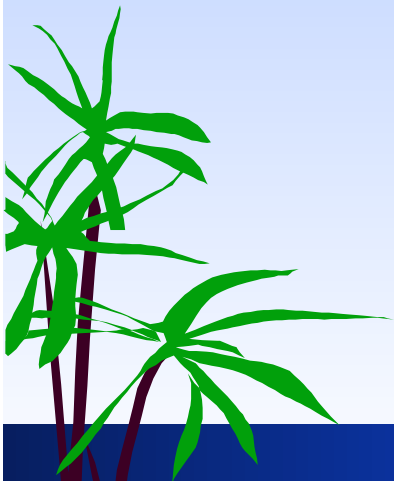
# errno

1 Måste inkludera `<errno.h>`



# Trådsäkra funktioner

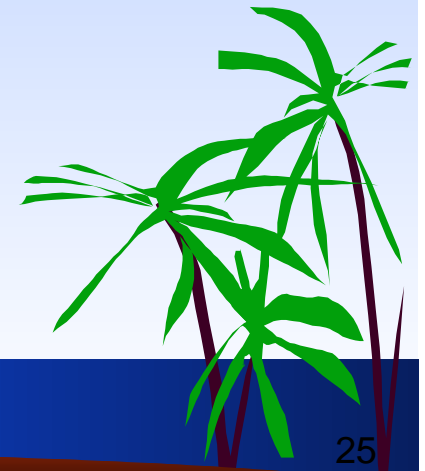
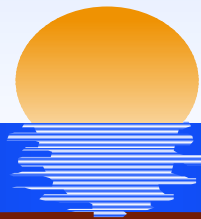
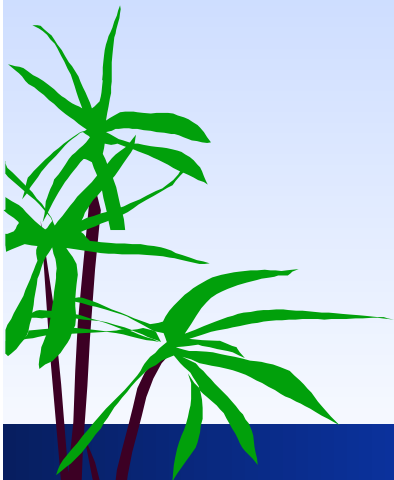
- 1 Läs manualsidan!
- 1 Thread-Safe vs Reentrant
- 1 POSIX trådsäkra funktioner (\_r)





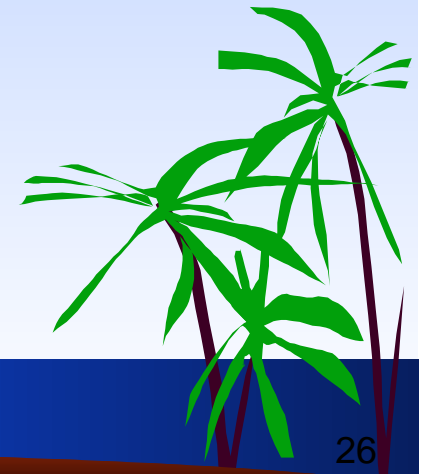
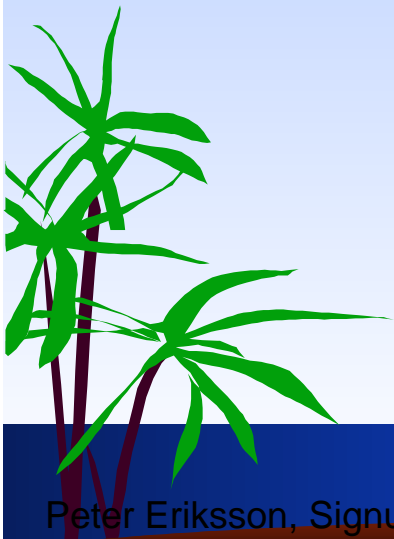
# Manuellt läsbara funktioner

- 1 flockfile() / ftrylockfile() / funlockfile()
- 1 getc\_unlocked() / getchar\_unlocked()
- 1 putc\_unlocked() / putchar\_unlocked()



# Trådosäkra funktioner

- 1 Använder static data, fel errno etc...
- 1 Undvik helst...
- 1 Om ej möjligt, anropa från main-tråden



# Signalhantering

- 1 pthread\_kill()
- 1 sigwait()
- 1 pthread\_sigmask()
- 1 Async-safe
- 1 longjmp()
- 1 Tips: All signalhantering i en tråd

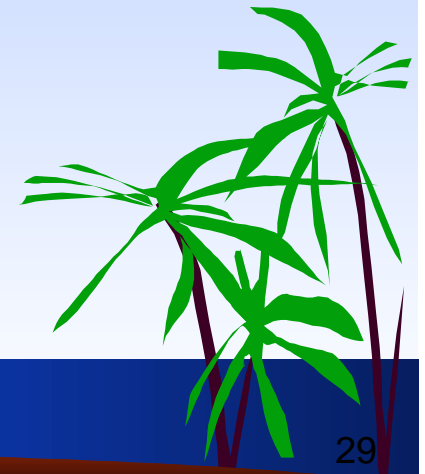
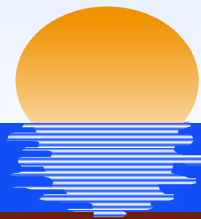
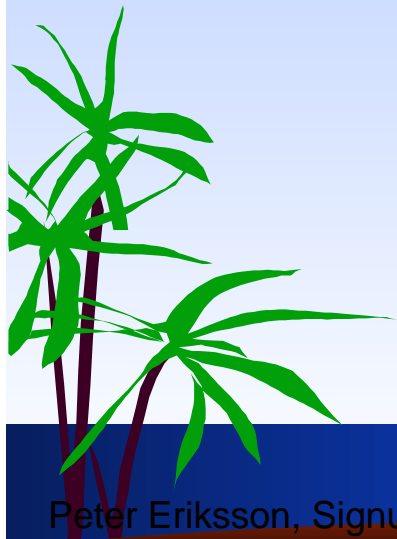
# Exempel

```
int main(int argc, char *argv[]) {
    sigset_t set; pthread_t tid; int sig;
    ...
    sigemptyset(&set);
    sigaddset(&set, SIGHUP);
    pthread_sigmask(SIG_BLOCK, &set, NULL);
    pthread_create(&tid, NULL, my_thread, NULL);

    while (sigwait(&set, &sig) == 0) {
        switch (sig) {
            case SIGHUP:
                do_something();
            ...
        }
    }
}
```

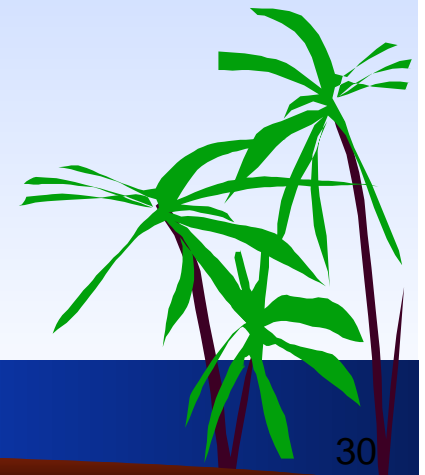
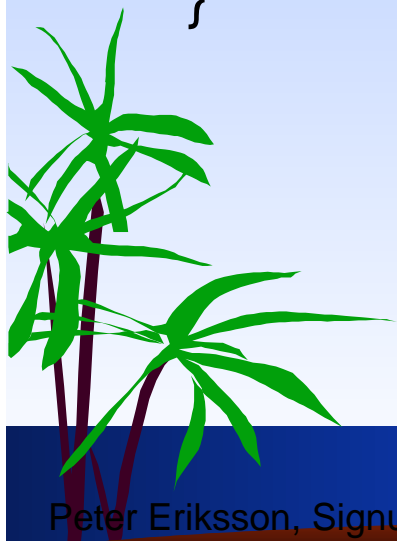
# TSD (trådlokala data)

- 1 pthread\_key\_create()
- 1 pthread\_key\_delete()
- 1 pthread\_setspecific()
- 1 pthread\_getspecific()



# Exempel

```
static pthread_key_t    my_key;  
static pthread_once_t  once = PTHREAD_ONCE_INIT;  
  
/* To be called via pthread_once_init() */  
static void my_init(void) {  
    pthread_key_create(&my_key, free);  
}
```



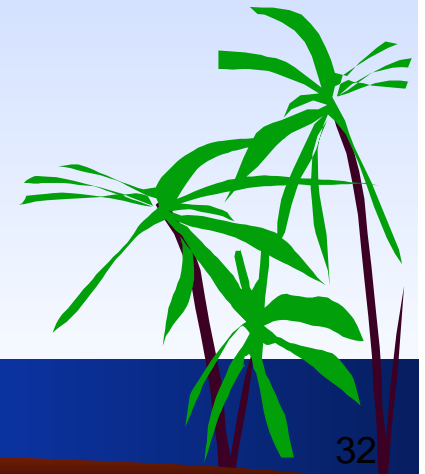
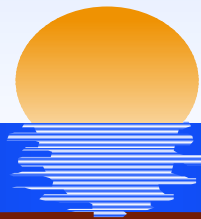
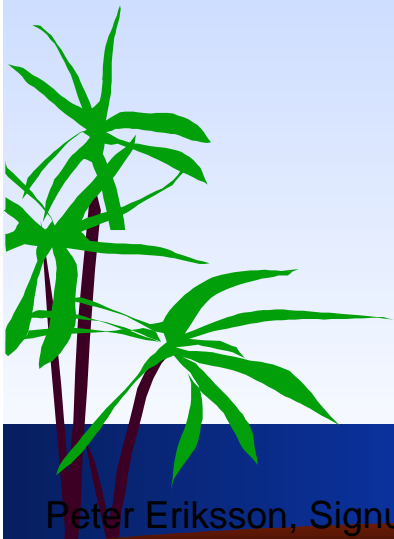
# Exempel, del 2

```
char *my_fun(int val) {
    char *res;

    pthread_once_init(&once, &my_init);
    res = pthread_getspecific(my_key);
    if (res == NULL) {
        res = malloc(4711);
        pthread_setspecific(my_key, res);
    }
    ...
    return res;
}
```

# Avancerad programmering

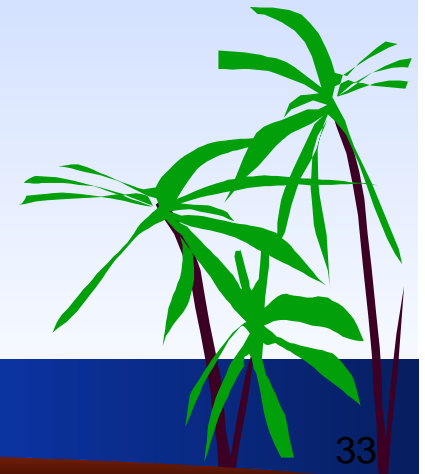
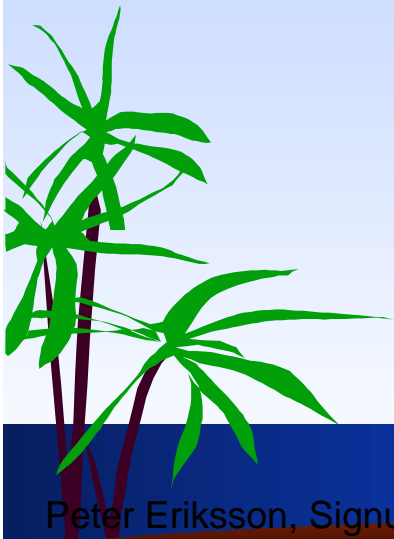
- 1 Trådcancellerig
- 1 Trådning av gamla program
- 1 Multipla processorer
- 1 Parallell beräkning





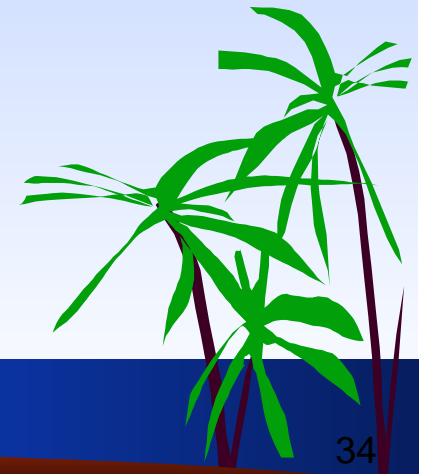
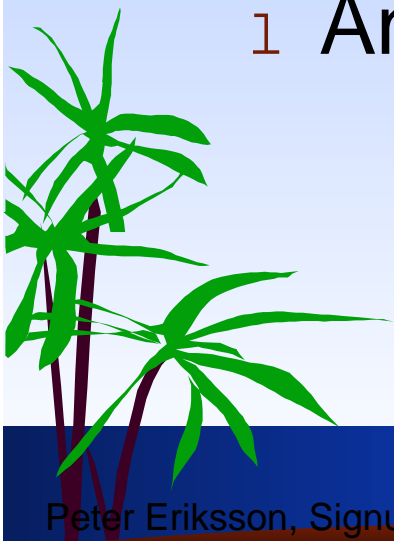
# Kompilering

- 1 #define \_POSIX\_C\_SOURCE=199506L
- 1 -lpthreads
- 1 #define \_REENTRANT



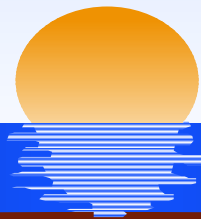
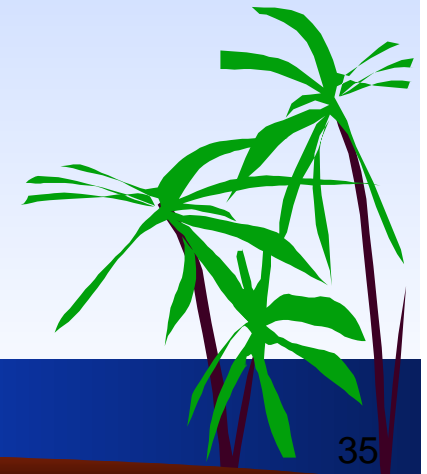
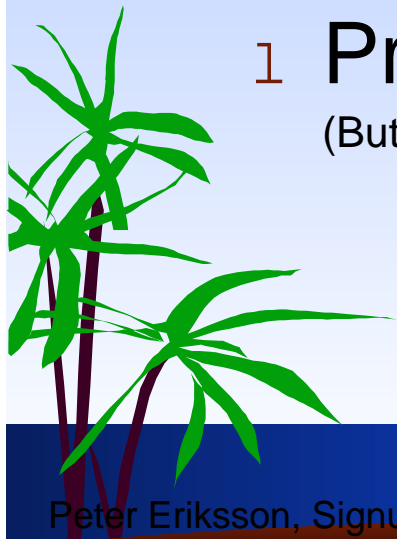
# Vanliga problem

- 1 “static”-deklarerade variabler
- 1 Globala variabler
- 1 Deadlocks
- 1 Djupt rekursiva funktioner
- 1 Använd “invariants” för felkontroll!



# Dokumentation

- 1 Programming with Threads  
(Kleiman, Shah & Smaalders, Prentice Hall, ISBN 0-13-172389-8)
- 1 Multithreaded Programming Guide  
(Sun Answerbook)
- 1 Threads Primer  
(Lewis & Berg, SunSoft Press, ISBN 0-13-443698-9)
- 1 Programming with POSIX Threads  
(Butenhof, Addison-Wesley, ISBN 0-201-63392-2)



# Övrigt

- 1 Avlusning (gdb, debugger, printf)
- 1 lock\_lint, tnfview
- 1 comp.programming.threads

