Scheduler Multiprocessors Preemption Run

Problems:

These problems are addressed by

(1) They're

Problems with

It is generally assumed that semaphores are fair, in the sense that

function remove returns data :

procedure insert(d : data) :

shared

shared mutex : semaphore := 1

shared

procedure V(ref S : semaphore) :

procedure P(ref S : semaphore) :

Here is one possible implementation:

semaphore.

thread is de

initial value, and it keeps track of the excess (if any) of past V

In either case, a semaphore is a special sort of counter.  It has an

additional power (you can implement them trivially with binary

These are useful for certain algorithms, though they don't add any

We can extend this to

could perform acquire operations without blocking.

are called "binary" because we can think of them as a counter that

P as standing for "pause", though it

named V; these stand for words in Dutch.  (Mnemonically, I think of

locks

So

re

release(

return d

V(mutex)

next_full

buf

P(mutex)

next_empty

S.N +:= 1

enqueue(

scheduler_lock

S.N

disable_signals

sleep_on

sleep

yield:  

Disabling signals doesn't suffice:

Note that

Some other thread/process will move us to the ready list when we can

Now

Suppose

transfer(other):

As in

add multiple processors

add preemption

make uniprocessor run

Start with coroutines

- (as in MacOS 9

prereq

shelved

split

reassemble

or

Windows 3.1):  

We're just relinquishing the processor for the sake of

or

We're just preempting, in a way that we will discuss shortly.

In the worst case, a semaphore is a collection of backlogged Run

The

run_list

are pointers to

context blocks

In this case, a semaphore is a simple data structure:

Each

take the current thread and put it on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or

on the Run

or