Session
AD019

Threads vs. Events

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 718 463 1079
gezelter@rlgsc.com

Tuesday, November 12, 1996
10:00 am – 10:50 am
Room B2

Fall 1996 US DECUS Symposium
Anaheim Convention Center
Anaheim, California

Introduction

Robert Gezelter
Software Consultant

In recent years, Threads have been the rage.

Like most fads, Threads have strengths and weaknesses which must be considered when planning software development.

Robert Gezelter
Software Consultant

NOTES

Thread Model:

A process contains multiple threads: Each thread may computable or blocked at any instant in time.

Examples:

DECthreads, Windows NT

Robert Gezelter
Software Consultant

Event Model:

A process consists of the processing of a series of events. At any instant in time, a single event is being processed.

Examples:

OpenVMS AST, X-Windows

Robert Gezelter
Software Consultant

NOTES

Threads vs. Events – A Comparison

- ***Mathematically Equivalent***
- ***Significant Complexity Difference***
- ***Which is better?***

Robert Gezelter
Software Consultant

Threading:

If no interactions, simple application implementation.

If there are interactions between threads, then the interactions require locking and other mechanisms to prevent subtle programming errors from causing failures.

Robert Gezelter
Software Consultant

NOTES

Events:

Each event is treated as an independent transaction by the program. Pre-emption is not permitted, so there are no locks or synchronization required to access data structures during the processing of a single event.

Robert Gezelter
Software Consultant

Process Types

- ***Heavyweight***
- ***Lightweight***
- ***Featherweight***

Robert Gezelter
Software Consultant

NOTES

Heavyweight Processes

- ***Separate Register Set***
- ***Separate Stack***
- ***Separately Dispatchable***
- ***Separate Address Space***
- ***Expensive Creation***

Robert Gezelter
Software Consultant

Lightweight

- ***Separate Register Set***
- ***Separate Stack***
- ***Separately Dispatchable***
- ***Preemptable***
- ***Low resource consumption***

Robert Gezelter
Software Consultant

NOTES

Featherweight

- ***Shared Register Set***
- ***Shared (nested) Stack***
- ***Separate Address Space***
- ***Extremely inexpensive Creation***
- ***No preemption***
- ***Implicit synchronization***
- ***Extremely inexpensive***

Robert Gezelter
Software Consultant

When Threads/Events?

Use threads only when there a payoff.

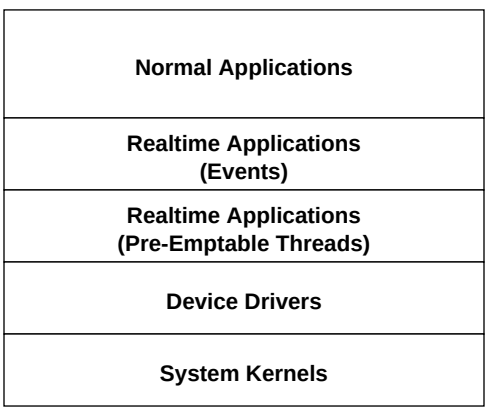
Pre-emption only pays when CPU saturation occurs.

Hierarchy of Effort

Robert Gezelter
Software Consultant

NOTES

Hierarchy of Effort



Robert Gezelter
Software Consultant

Threading Hazards

- ***synchronization***
- ***complexity***
- ***proper tool?***
- ***debugging***
- ***data structure locking***

Robert Gezelter
Software Consultant

NOTES

Threading Implementation Basics

- ***preemption model***
- ***priority model***
- ***debugging***
- ***application suitability***

Robert Gezelter
Software Consultant

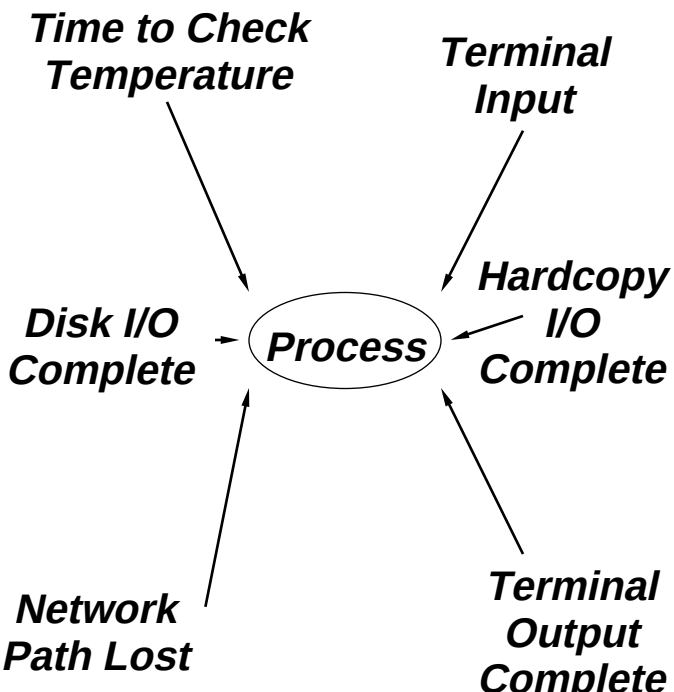
Events (Non-Preemptible Threads)

- ***FIFO***
- ***Non-preemptable***
- ***'Featherweight'***

Robert Gezelter
Software Consultant

NOTES

Typical Event Driven Computer Application



Robert Gezelter
Software Consultant

Event Processing —

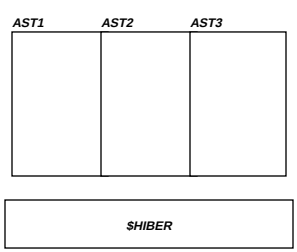
- ***External events control program***
- ***Programs need to be efficient***
- ***External event sequence is not under program control***
- ***No Dispatch Routine***

Robert Gezelter
Software Consultant

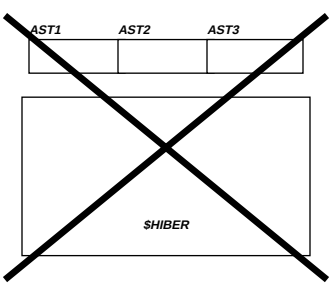
NOTES

Tricks to Getting It Right

Do ALL Processing in as Events



Good



Bad

Robert Gezelter
Software Consultant

Avoid Problems

- ***Kill bugs before they occur***
- ***DO NOT inhibit Events.***

Robert Gezelter
Software Consultant

NOTES

Summary:

The minimum needed level of complexity is desireable. If your application is not operating in a CPU saturated environment, then the performance benefits of pre-emptible threads are illusory at best, and represent a significant increase in the complexity of your application over the functionally equivalent application using events.

Questions?

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 718 463 1079
gezelter@rlgsc.com

+1 718 463 1079
35 – 20 167th Street, Suite 215, Flushing, New York 11358 – 1731 USA

NOTES