CMPE 220

Class 25 – Real Time Operating Systems

Types of Operating Systems

- Batch / Single-Process (early computers – 1940s-50s)
- Multi-Programming Systems ("modern" computers – last 1960s)
- Multi-Processor Systems
- Distributed Operating Systems
- Network Operating Systems
- Embedded Systems
- Real-Time Systems
- Cloud Systems
- Mobile Systems

San José State University

What is a Real-Time Operating System (RTOS)?

- Designed for applications that have critically defined time constraints
- *Predictability* is often more important than *speed*
- Deterministically meeting stated performance characteristics is more important than system throughput

San José State University

The first RTOS?

- IBM’s Basic Executive, released in 1962
  - Ran on the IBM 1710
  - A minimal system featuring an interrupt handler, I/O drivers, and a scheduler/dispatcher
  - Added FORTRAN support by 1963
  - “The Fortran Executive System provides the user with the ability to direct process control operations with programs written in the Fortran language”

    – IBM Systems Reference Library

San José State University

IBM’s Fortran Executive – circa 1963

Download the manual in Canvas

RTOS Classifications

- **Hard** - A hard real-time system causes an entire system to fail if it misses its deadline or time constraint.
  - For example, a flight control system with an unacceptable latency may cause an aircraft to crash.
- **Firm** - With this type of app, a missed deadline is tolerable but causes significant degradation in quality.
  - For example, in video conferencing, latency may degrade the quality of a call, but the system is still usable.
- **Soft** - With these apps, results degrade after their deadline, whether the deadline is met or not.
  - A video game is an example of a soft real-time system. Video games rely on user input and have limited time to process; degradation is sometimes expected for this reason.

San José State University

Can Real-Time Response be Guaranteed?

- Yes - self driving automobile
  - There are a fixed number of inputs / events
- No - web server
  - A potentially unlimited number of inputs / events
    - Denial-Of-Service attacks

Dedicated Applications

- Embedded systems
  - Self-driving automobile
  - Flight control system
  - IoT
- Command and Control
  - Factory automation

- *Single application system*

RTOS Architectures

- **Microkernel** – applications request services from an underlying operating system
- **Monolithic** – the application and the OS are integrated; the OS is essentially just a library that is built into the application

San José State University

Types of Failures & Mitigation

- Load (input/event limit exceeded)
    - Scalable (Cloud) Resources
        - Limits to scalability
- Hardware failure
    - Redundant hardware
- Unexpected software error
    - Recovery algorithm
- Deadlocks
    - Abort and restart processes
- Catastrophic (non-recoverable) failure
    - Auto-Restart system

- All
    - Operate in degraded mode

Scheduling Approaches

- Round-Robin (aka timeshare)
    - Response time guaranteed
    - < number of inputs * maximum service time
- Event-Driven (aka priority scheduling)
    - Switches tasks only when a higher-priority event (interrupt) occurs

Scheduling Algorithms

- **fixed priority preemptive scheduling** - the scheduler ensures that at any given time, the processor executes the highest priority task of all those tasks that are currently ready to execute
- **Non-preemptive** – tasks run to completion
- **Deferred preemption** – portions of the code cannot be preempted
- **Earliest deadline first** - when a scheduling event occurs (task finishes, new task released, etc.) the queue will be searched for the process closest to its deadline
  - Optimum strategy for CPU utilization; upper bound = 100%

Memory Management

- Dynamic memory allocation is avoided
  - Lack of predictability
  - Possibility of memory leaks
- Virtual memory / swapping is avoided
  - Disks have long and unpredictable latency
- Implications:
  - Limited memory
  - Small OS
  - Small applications
  - Code optimized for size
  - Critical sections optimized for performance

San José State University

Jitter

- The amount of error in the timing of a task over subsequent iterations of a program or loop is referred to as *jitter*
- Real-time operating systems are optimized to provide a low amount of jitter
- A hard real time system has less jitter than a soft real time system

San José State University

GPOS versus RTOS

General Purpose Operating System	Real Time Operating System
Example: desktop computer	Example: flight control system
Applications: general mix	Applications: dedicated
Response time: unbounded	Response time: guaranteed
Scheduling: optimized for throughput and “fairness”	Scheduling: optimized for predictability
Memory model: manages large amounts of dynamic memory	Memory model: small, fixed footprint
Mass storage: disk	Mass storage: semiconductor or none
Jitter: Milliseconds to seconds	Jitter: A few to tens of microseconds

San José State University

In-House versus Commercial

- Because an RTOS is small and often tightly coupled with the application, it’s possible to develop a custom OS in-house
- Risks of In-House Development:
  - Poor understanding of real time considerations
  - Poor understanding of security considerations
  - Maintenance expenses

San José State University

Selected RTOS

System	Released	Uses	Vendor
VxWorks	1987	Embedded / C&C	Wind River Systems
embOS	1992	Embedded	embOS
FreeRTOS	2003	Embedded / C&C	MIT / Amazon Web Services (AWS)
LynxOS	1986	Embedded / Avionics	Lynx Software
PikeOS	2005	Virtualization platform	Sysgo
Azure	1997	Embedded	Azure
Neutrino RTOS	1983	Embedded	Quantum Software Systems
SafeRTOS	2007	Embedded	Wittenstein High Integrity Systems (WHIS)
Zephyr	2016	Embedded / C&C	Open source

San José State University

POSIX and RTOS

- POSIX (Portable Operating System Interface) - a *family of standards* specified by the IEEE Computer Society for maintaining compatibility between operating systems
- POSIX defines both the system and user-level application programming interfaces (APIs), along with command line shells and utility interfaces
- In order to be compliant, an operating system must pass a suite of tests

POSIX and RTOS

- Embedded systems typically have space and resource limitations, and an operating system that includes all the features of POSIX may not be appropriate
- The **POSIX 1003.13** profile standard was defined to address these types of systems.13 POSIX 1003.13 does not contain any additional features; instead it groups the functions from existing POSIX standards into units of functionality

San José State University

POSIX and RTOS – Real Time Extensions

- POSIX 1003.1b defines extensions useful for development of real-time systems
  - Timers: periodic timers, delivery is accomplished using POSIX signals
  - Priority scheduling: fixed priority preemptive scheduling with a minimum of 32 priority levels
  - Real-time signals: additional signals with multiple levels of priority
  - Semaphores: named and memory counting semaphores
  - Memory queues: message passing using named queues
  - Shared memory: named memory regions shared between multiple processes
  - Memory locking: functions to prevent virtual memory swapping of physical memory pages

Benchmarking

- GPOS benchmarks attempt to represent a *typical task*, or a typical *mix of tasks*
- RTOS benchmarks may be less meaningful, because the system behavior is tightly coupled with the application
- RTOS benchmarks focus in specific system characteristics or services

San José State University

Typical RTOS Metrics

- Interrupt latency
- Task switch time
- Preemption time
- Deadlock break time
- Jitter

- System restart time

San José State University

Security

- RTOS systems and applications are often used for *critical systems*, making security an important consideration
- SafeRTOS is an example of a security hardened & certified real time operating system

Real Time Data Analysis

- A growing field
- Real time data monitoring, analysis, reporting, and control
- **Soft** systems
- Example: California power crisis
  - System automatically load-balanced
  - Governor Newsom asked Californians to cut power use
  - Demand dropped by 4%

Assignment 10

- A few questions – due next Monday

San José State
UNIVERSITY