Real Time System Final Note

aokblast

June 15, 2023

1 Homework

- Real time timing mechanism in Linux (e.g. clock_gettime(2))
- Linux Timer(timer_create(2), timer_settime(2), timer_delete(2)) and Signal(sigaction(2))
- pinCPU (sched_setaffinity(2))
- Scheduler changing aned priority mechanism (e.g. chrt(1))
- Pthread(pthread_create(3), pthread_mutex_lock/unlock(3), pthread_cond_signal/wait(3))
- Scheduler changing aned priority mechanism in C(pthread_setschedparam(3))
- Pthread based PIP, PCP, and NPP(pthread_mutexattr_setprotocol(3), pthread_mutexattr_setprioceiling(3))

2 In Class

2.1 Real Time Constrain

Hard Real Time Task missing deadline may cause catasrophic consequences on the system under control

Firm Real Time Task missing deadline does not cause any damage to the system but no output

Soft Real Time Task missing deadline has still some utility for the system but cause some performance

2.2 Process Utilization Bound

- Process Utilization Factor $U = \Sigma_{i=1}^{n} \frac{C_i}{T_i}$
- Process Utilization Bound $U_{ub}(\Gamma, A)$ = upper bound of U of a task set Γ under a given algorhtm A
- Least Upper Bound $U_{lub}(A) = min_{\Gamma} U_{ub}(\Gamma, A)$
- Fixed Priority Scheduling Algorithm $U = 2(2^{\frac{1}{2}} - 1)$
- Deadline Driven Scheduling Algorithm(EDF) $U = 1$
- RM $U_{lub} = ln2 \approx 0.69$

2.3 Scheduling Anomalies

Fast CPU doesn't means good scheduling performance The Scheduler may wrong scheduling in parallel core have higher speed that makes the total execution time worse than original case

2.4 Schedulability Analysis

1. $I_i = \Sigma_{j=1}^{i-1} \lceil \frac{D_i}{T_j} \rceil C_j$

2.5 Response Time Analysis

Through Iteration with following equations

1. $R_i^{k+1} = C_i + I_i^{(k)}$
2. $R_i = C_i + \Sigma_{j=1}^{i-1} \lceil \frac{R_i}{T_j} \rceil C_j$
3. $I_i^{(k)} = \Sigma_{j=1}^{i-1} \lceil \frac{R_i^{(k)}}{T_j} \rceil C_j$

2.6 Fixed Priority Server

- Background Scheduling
  - Two Queue, One for periodic Task and another for aperiodic Task.
  - Aperiodic Task will execute only when Periodic Task is empty.
  - Two queue can use different scheduling algorithm
- Polling Server
  - A periodic task will server aperiodic task
  - When no task can execute, the periodic task will sleep and waiting next period
- Deferrable Server
  - Like Polling Server but the server can sleep and execute task that issue in the period (Task can be executed in the middle period rather than period start)

2.7 Resource Access Protocal

Non-Preemptive protocal(NPP) Disallow any preemption by raise the task priority to the hightest one

Highest Locker Priority(HLP) Raise the task priority to the hightest task that share that resource

Priority Inheritance Protocal(PIP) Raise the task priority to the highest task when enter critical reghion

Priority Ceiling Protocal(PCP) Raise the task priority to the hightest task that share that resource when enter critical region

2.8 Hard to describe topic in note

Fault-tolerant Research that can tolerant some deadline missing C. Wang, C. Gill, and C. Lu, FRAME: Fault Tolerant and Real-Time Messaging for Edge Computing, IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019, pp. 976-985

Cyber-Physic Research that combile physical environment like sensor network G. Hackmann; W. Guo; G. Yan; Z. Sun; C. Lu; and S. Dyke, "Cyber-Physical Codesign of Distributed Structural Health Monitoring with Wireless Sensor Networks." IEEE Transactions on Parallel and Distributed Systems (Volume: 25, Issue: 1, Jan. 2014), pp. 63 - 72, DOI: 10.1109/TPDS.2013.30.

Mixed-Criticality Systems Research that combine different task criticality Vestal, Steve. "Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance." 28th IEEE International Real-Time Systems Symposium (RTSS 2007). IEEE, 2007.

Virtualization Virtualization on Real Time OS S. Xi, J. Wilson, C. Lu and C.D. Gill, RT-Xen: Towards Real-time Hypervisor Scheduling in Xen, ACM International Conference on Embedded Software (EMSOFT'11), October 2011.