Performance Analysis
of
Operating Systems

Project Progress Report

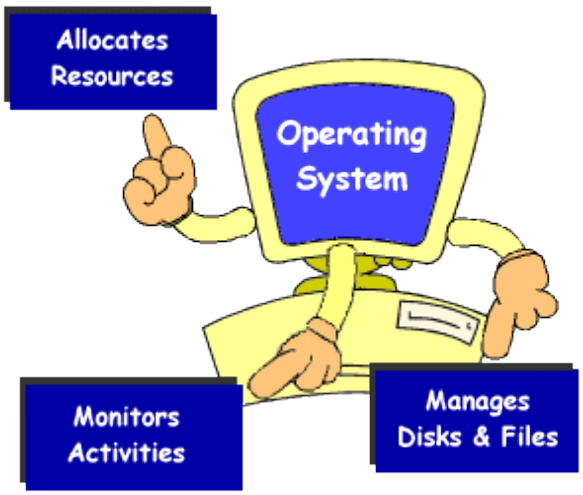
Sagar Shelke #820315776

Neetha Reddy #820855705

Sushma Kuchipudi #820850180

Introduction

Operating systems are broadly divided into two categories depending upon its applications: one is general purpose operating system (GPOS) and other is real time operating system (RTOS).



A GPOS is used for systems/ applications that are not time critical. Example: Windows, Linux, Raspbian etc.

A RTOS is on the other hand is used for time critical applications. Example: FreeRTOS, uCos etc. We can say an RTOS is supposed to give quick and predictable response.

These implementations differ in:

1.Task Scheduling

2.Latency Issues

3.Preemptible kernel

This project will present quantitative and qualitative results obtained from the comparison of performance analysis of RTOS and GPOS.

This report will present results obtained by running a real-time application on GPOS Raspbian which is widely used on Raspberry Pi. To mimic actual field condition, application is interrupted with external hardware interrupt.

Literature Survey

There are many efforts done in analyzing performances of different operating systems, which helps to choose one best for required application.

Two methods are widely used to measure performance of OS:

1. Analyzing performance counters of processor used to run OS and real rime application [1]
2. Using standard library(benchmarking platforms) to get system information [2]

Performance parameters achieved through first method are limited to number of registers in processor whereas second method gives more flexibility being pure software implementation.

We used a suite of micro-benchmarks to measure and compare specific aspects of system functionality. psutil (python system and process utilities) is standard cross-compile library which given detailed analysis of CPU, Memory, Network and timing in python.

Projects implemented so far compares two or more real time operating systems with each other whereas some projects are dedicated for comparing performance of desktop operating systems (GPOS). In this project, we compare GPOS with RTOS. Raspbian (distribution of Debian) is a microkernel based GPOS widely used on RPi . We will be selecting open source RTOS (One of the Free RTOS, Risc OS, Drops).

HW Components

Raspberry Pi 3 Model B:

Raspberry Pi is a single board computer, which can is used in embedded systems. It has following specifications:

A 1.2GHz 64-bit quad-core ARMv8 CPU

802.11n Wireless LAN

Bluetooth 4.1

Bluetooth Low Energy (BLE)

1GB RAM

40 GPIO Pins

Challenges:

Connecting Raspberry Pi directly to the computer creates problem (No IP address assigned) because IP address is assigned by DHCP server which is router.

Therefore, every time we need to connect RPi to router and then access its GUI through SSH client installed on laptop. (We need to make sure that SSH server is installed on RPi and enabled)

Changes:

Temperature sensor (TM 36) we bought gives analog output whereas RPi takes only digital input (unlike Arduino). We needed to use ADC or buy new sensor with digital output therefore we changed to software based RT application.

SW Components

Wnet Watcher (IP Address Scanner):

DHCP server (Router to which RPi is connected) allocates private IP to RPi for fixed amount of time and therefore it is dynamic. We check IP assigned to RPi using Wnet Watcher IP scanner.

Putty:

We have installed SSH server on RPi. Putty is used as SSH client for remote monitoring of RPi. After logging into RPi we install vnc- server.

VNC Viewer:

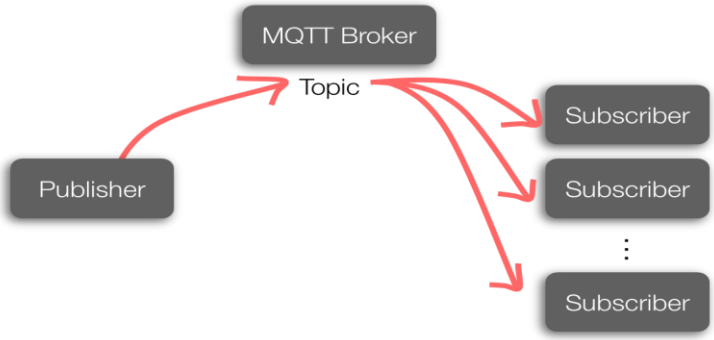
This is vnc-client used to view and control RPi desktop.

Raspbian:

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run .

MQTT and RT Application:

Real time application in this project is controlling devices connected to RPi from Smart phone using MQTT protocol. MQTT is widely used protocol in IoT.



In our application, MQTT broker is “iot.eclipse.org”. MQTT clients are installed on RPi and Smart Phone (Android application “MQTT Dashboard”). Both clients are subscribed to same topic (in our case “sagar/demo/led”). Whenever smart phone publishes something, client on RPi will subscribe it and proceed according to command.

Interrupt is given externally to mimic field environment, which causes a 3-bit counter to run by interrupting MQTT client program.

Callback function is used to detect for interrupt continuously.

Python and psutil:

Script of application is written in Python (high-level general- purpose programming language).

psutil (Python system and process utilities):

psutil (python system and process utilities) is a cross-platform library for retrieving information on running processes and system utilization (CPU, memory, disks, network) in Python. It is useful mainly for system monitoring, profiling and limiting process resources and management of running processes.

Experimental Results

Success Metrics:

CPU Performance Parameters (System and Process)

1. CPU timing in user and kernel mode.
2. Percentage of CPU used.
3. Number of context switches

Memory Performance Parameters (System and Process)

1. Memory information
2. Percent of memory used
3. Number of child processes

Timing Parameters (System and Process)

1. A-B Timing
2. ISR Latency
3. **Overhead due to commands**

Results:

Performance Parameters

PARAMETER	FUNCTION	RESULTS
System Parameters		
CPU Time	psutil.cpu_times()	Scputimes(user = 533.81, nice = 0.0, system=147.51, idle = 43837.67, iowait = 14.35, irq = 0.0, softirq = 1.16, steal = 0.0, guest = 0.0, guest_nice = 0.0)
CPU Percent	psutil.cpu_percent()	27.5
Memory Used	psutil.virtual_memory()	svmem(total=970485760L, available=809639936L, percent=16.6, used=322138112L, free=648347648L, active=216465408, inactive=70365184, buffer=28729344L, cached=132562944)
Memory Swapped	psutil.swap_memory	sswap(total=104853504L, used=0L, free=104853504L, percent=0.0,sin=0, sout=0)
Boot Time for Raspbian	psutil.boot_time()	1477788019.0
A-B Time	----------	0.185477972031
Process Parameters		
Main Process ID		2688
Main Process CPU Time	p.cpu_times()	pcputimes(user=0.57, system=0.04)
Main Process Context Switches	P.num_cxt_switches()	Pctxsw(voluntary=175, involuntary=29)
Main Process CPU Percent	p.cpu_percent()	0.0
Number Threads in Main Process	p.num_threads()	4
Main Process Memory information	p.memory_info()	pmem(rss=15257600, vms=50614272)
Main Process Memory Percent	p.memory_percent()	1.57216114124
Main Process Children	p.children(recursive=True)	0
ISR Latency	----------	0.11154794693
ISR Process ID		2688
ISR CPU Time	P_1.cpu_times()	pcputimes(user=0.7, system=0.11)
ISR Context switches	p_1.num_ctx_switches()	pctxsw(voluntary=223, involuntary=35)
ISR CPU Percent	p_1.cpu_percent()	0.0
Number of threads used in ISR	p_1.num_threads()	4

ISR process memory information	p_1.memory_info()	Pmem(rss=15368192, vms=51662848)
ISR process memory percent	p_1.memory_percent()	1.58355667166
ISR process children	p_1.children(recursive=True)	0

Functional Overhead:

Function	**Timing Overhead (sec)**
time.time()	8.106e-6
psutil.cpu_times()	0.0004141
psutil.cpu_percent()	0.000495910
psutil.virtual_memory()	0.00042700
psutil.swap_memory()	0.00052309
psutil.boot_times()	0.000407934
p_1.num_ctx_switches()	0.00045800
p_1.cpu_times()	0.000283900
p_1.cpu_percent()	0.00046300
p_1.num_threads()	0.000333070
p_1.meory_info()	0.000271790
p_1.memory_percent()	0.0002779969
p_1.memory_maps()	0.044970980
p_1.children()	0.045200824

Future Milestones

- Finding networking and hardware parameters on GPOS Raspbian.
  1. System wide network I/O statistics
  2. System wide socket connections etc.
- Installing open-source RTOS (Risc OS, Free RTOS, Drops or another)
- Running RT application on RTOS and performing same operations.
- Comparison and analysis of both operating systems.

Work Division

Sagar- RT Application and interrupt design. Timing analysis parameters

Neetha- CPU performance calculation

Sushma- Memory related performance parameters.

References

[1]. J. Bradley Chen, Yasuhiro Endo, Kee Chan, David Mazières ,Antonio Dias, Margo Seltzer, and Michael D. Smith, "The Measured Performance of Personal Computer Operating Systems" In *The Proceedings of the 15th ACM Symposium on Operating System Principles.*

[2]. Kevin Lai and Mary Baker "A Performance Comparison of UNIX Operating Systems on the Pentium" In *1996 USENIX Technical Conference, January 1996.*

[3]. Rafael V. Aroca, Glauco Caurin "A Real Time Operating Systems (RTOS) Comparison", Universidade de S˜ao Paulo (USP), Technical Report.

[4]. A.Barbalacel, A. Luchetta, G. Manduchi, M. Moro', A. Soppelsa and C. Taliercio "Performance Comparison of VxWorks, Linux, RTAI and Xenomai in a Hard Real-time Application" *IEEE 1-4244-0867-9/07*

[5]. Benjamin Ip "Performance Analysis of VxWorks and RTLinux", Technical Report, Columbia University, USA