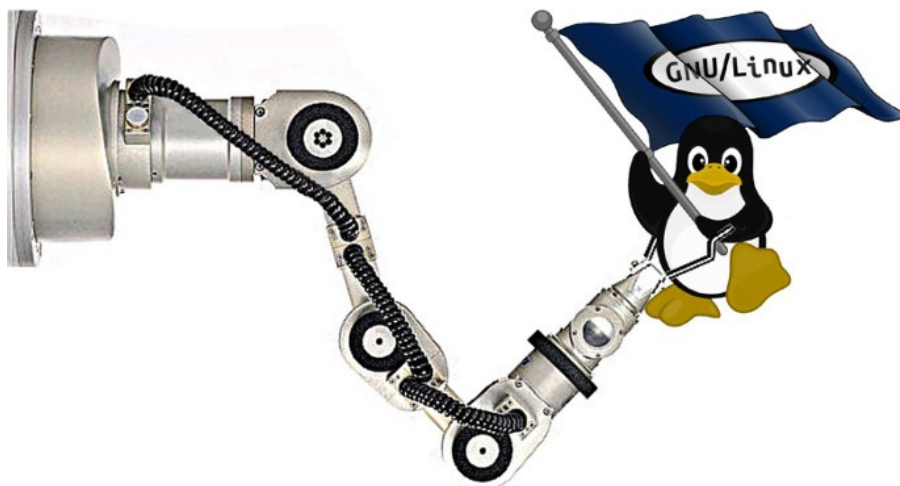


# Real-Time Linux



□ 投影片: [Making Linux do Hard Real-time](#)

## 簡介

---

「工業 4.0」反映了 real-time system (即時系統) 的需求，而自動控制工業大廠如 Siemens 和 KUKA 均投入研發資源在 GNU/Linux 為基礎的機器人設計，但尚未有圓滿地解決現有需求的單一方案，原因是 GNU/Linux 過於複雜而且得考慮到開發社群的多樣需求。

於是，在高速的發展過程中，邁向 hard real-time 作業系統的目標，伴隨著巨大的維護成本，其中 Linux 核心主流的解決方案就是 PREEMPT\_RT。本線上講座嘗試從排程、中斷處理、high resolution timer (HRT)，對稱多核心 (SMP) 架構特有設計等角度去探討 PREEMPT\_RT 的設計和實做議題。

## 1分鐘看完數位馬達

---

- [數位馬達](#)



一台機械手臂要價 120 萬貴嗎？其實很便宜

- 機械手臂的組裝速度，是人類的 3 倍
- 人類一天工作 8 小時，機械手臂可以 24 小時
- 人類一周休息 2 天，機械手臂工作 7 天
- 人類一年放假 130 天左右，機械手臂工作 365 天

整體算起來，機械手臂是人類的 12 倍產能

- 正常人一年的薪水最低算 30 萬就好，這樣換算起來，手臂的價格只要 1/3
- 10 隻手臂可以抵上百人，更省了管理問題 (上百人的公司，又要工會、福委會、育嬰室有的沒的)

所以建造機器人大軍絕對正確，千萬不要跟機器人競爭，時代在轉變，工作型態一定要轉變。

[ [source](#) ]

- 機械手臂結構大致有：
  - Cartesian: PPP ;
  - Cylindrical: RPP ;
  - Spherical: RRP ;
  - Articulated: RRR ;
  - SCARA (Selective Compliant Articulated Robot for Assembly) : RRP。
  - (P: prismatic ; R: revolute)
- 機械手的部份，又有Normal ; Sliding ; Approach等三軸。

搭配影片

- <https://www.youtube.com/watch?v=vA0k8i9VPNA>
- <https://www.youtube.com/watch?v=IfP6J9Bxt2w>

## Real-time Linux

---

Real-time Linux 不僅可拿來控制馬達，事實上應用範疇相當廣，早在 1997 年，Linux 被改造為 hard real-time 系統時，其中一個目的就是為了打造火箭控制系統 (請見 [RTLlinux](#), 研究人員來自美國新墨西哥大學 [NMT])，經過這近 20 年的演化，產生了相當多不同的組合。

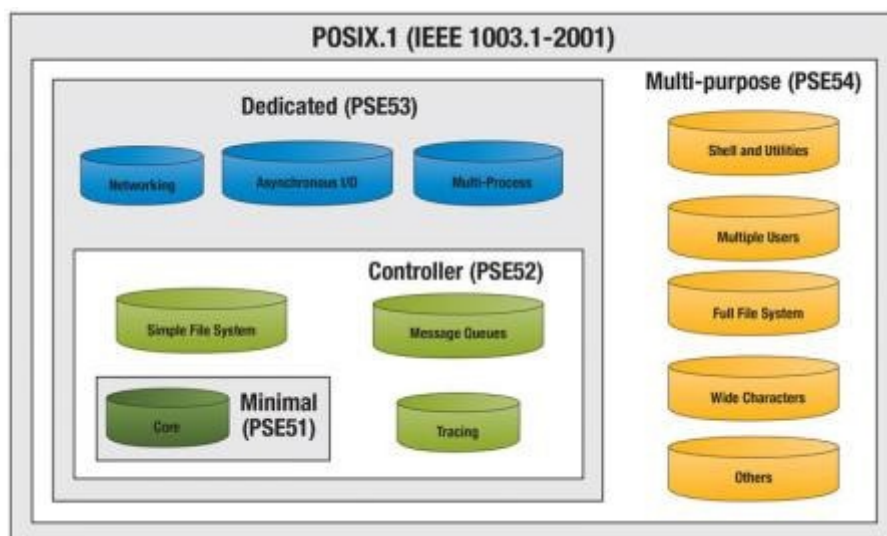




成功大學資訊工程系課程報告: [Xenomai](#)

## POSIX

- [POSIX – 25 Years of Open Standard APIs](#)
  - 用圖表說明了 PSE51, PSE52, PSE53, PSE54 , 以及用於即時處理應用的 IEEE 1003.13-2003 (POSIX.13) Profiles



- 1986 年，IEEE 指定了一個委員會制定了一個開放作業系統的標準，稱為 POSIX (Portable Operating Systems

Interface), 最後加上個 "X" 是因為本質上是 UNIX 的標準

### API standards for Open Systems

- IEEE POSIX Standards for Real-time
- The PASC Real-time System Services Working Group (SSWG-RT) has developed a series of standards that amend IEEE Std 1003.1-1990 and a profile standard (IEEE Std 1003.13-1998).
- The Real-time amendments to IEEE Std 1003.1-1990 are as follows:
  - IEEE Std 1003.1b-1993 Realtime Extension
  - IEEE Std 1003.1c-1995 Threads
  - IEEE Std 1003.1d-1999 Additional Realtime Extensions
  - IEEE Std 1003.1j-2000 Advanced Realtime Extensions
  - IEEE Std 1003.1q-2000 Tracing
- Note that RTLinux from FSMLabs turns the 1003.13 hierarchy upsidedown, with the smaller PSE51/52 realtime threads in control, and the full-figured Linux system (similar in functionality to PSE54) as just another thread under control of the realtime threads.
  - This is the opposite of what the PASC SSWG-RT had imagined when 1003.13-1998 was written, but it nonetheless works.
- [POSIX module on top of Xenomai](#)

## NOTE

---

- RPi3 拿來做機械手臂是綽綽有餘的，不管是電腦視覺、通訊（內建無線網路晶片）或是運算，是有可能在其上建構出低成本機械手臂的，這也是我們今年預計執行的計畫，之後會有募款，歡迎有興趣的人共襄盛舉。
- 機械手臂之間需要相互通訊，Linux 在通訊方面及低耗方面都可以和商用的作業系統競爭。
- 我們要建構的是一個 Hard Real-Time System，一個小小的失誤很可能就會造成整個系統出錯，甚至是出人命的公共安全危機都有可能。
  - 這也是為什麼這些機械設備要賣這麼貴的原因，因為很有可能要拿來打官司。
- Soft Real-Time 的例子：聲音和影像。沒寫好可能頂多造成品質不佳。
- Hard Real-Time 還應用在交通系統, Google 自走車。
- 去年 Linux Foundation 宣佈 [Advance Real-Time Linux Project](#)，目的在加強 Linux 在 Real-Time 方面的處理能力。

- 裡面出現了 Altera, ARM, Intel, IBM, TI 等公司，但有趣的是連 Google 都出現在這個計畫中，顯示這個時代很大部份是由軟體公司去參與甚至主導工業控制相關設備的製造。
- Linux 要做到 Hard Real-Time 要想辦法降低 Latency，而不是只是讓速度變快而已。
- Real-Time Linux 在追求的目標是能做到一般 Linux 做的事，並且能夠有效降低 Latency。（反應時間小於 10 微秒）
- Real-Time 可以接受慢，但必須要平均，不可忽快忽慢。
- 即便是商用的 Real-Time OS 也無法保證在每個實做議題都 Deterministic
- 以 printf() 來說，如果今天要印的是一個超級長，長達一百萬個字元的字串，假設印一個 "Hello World" 的時間是一個單位時間，要印出超長的字串有可能需要 20 萬倍的單位時間，也有可能只要一個單位時間。
  - File I/O 的部份是 non-deterministic 的典型例子。
  - buffering
- Linux kernel 開發過程主要透過 mailing-list 交流討論，基本上很少繪製示意圖，而是用文字去分析、提案，並往返討論下一個階段的 Linux 要做哪些功能，所以要參與開發 Linux 的話，要培養自己對英文文字的敏感度。
- Interrupt Handling & Context Switch
  - Context Switch 基本上就是講狀態的轉換。
  - Nested Interrupt
  - Interrupt 都涉及了 Context 的保存
  - 發生 Context Switch 的時候，我們要做很多事
  - 很多的排程演算法都是需要 Clock 去做切割的
- Latency in Kernel
  - 在 Real-Time 中，我們在意的是能不能在指定的時間內完成
  - 而 Real-Time 中的 Latency 指的是 Interrupt 開始後，到重要的 task 可以被執行的這段時間
  - 你對著你的智慧型手機螢幕用手往上滑就是觸發了一個 interrupt，到畫面真正動的這段時間就是 Latency
- Interrupt Handler
  - Preemption 的重點就是讓 Priority 高的 Task 能夠優先被執行
- Full Preemptive
  - 為了要達到 Full Preemptive，Spin Lock 在這裡是可以被打斷的。
  - PREEMPT\_RT
    - Spin Lock 在此被當作 Mutex Lock



- Threading Interrupt

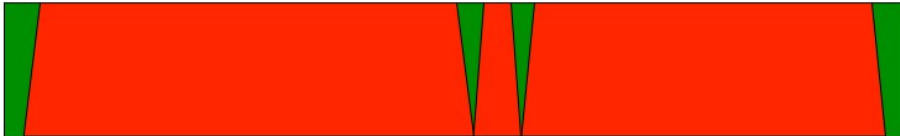
## Preemptible Kernel

---

- [Making Linux do Hard Real-time](#)
- [SMP/Linux Real-time Analysis](#)
  - 解釋 resolution 前，要先搞懂 [Spatial frequency](#)

(綠色: preemptible; 紅色: non-preemptible)

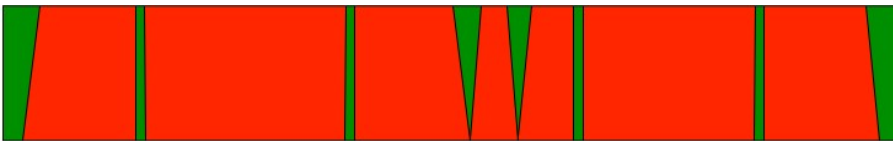
### Non-Preemptive



[ CONFIG\_PREEMPT\_NONE ]

- Preemption is not allowed in Kernel Mode
- Preemption could happen upon returning to user space

### Preemption Points in Linux Kernel



[ CONFIG\_PREEMPT ]

- Implicit preemption in Kernel
- preempt\_count
  - Member of thread\_info
  - Preemption could happen when preempt\_count == 0

### Fully Preemptive



[ CONFIG\_PREEMPT\_RT\_BASE ] / [ CONFIG\_PREEMPT\_RT\_FULL ]

- Difference appears in the interrupt context
- Goal: Preempt Everywhere except
  - Preempt disable
  - Interrupt disable
- Reduce non-preemptible cases in kernel
  - spin\_lock
  - Interrupt

- Process Management
  - Page 20
- Exploit the Advantages and Resolve the Challenges of Multicore Technology with Linux

ELC 2016 重點提示

=> <http://events.linuxfoundation.org/events/embedded-linux-conference>

☐ Xenomai 3: An Overview of the Real-Time Framework for Linux - Jan Kiszka, Siemens AG applications

- PLC, machine control system
- printing machines (manroland)
- network switches (Ruggedcom)
- Magnetic resonance tomograph (Siemens Health care)

symbol wrapping

- pthread\_mutex\_lock => \_\_wrap\_pthread\_mutex\_lock

preserve Linux service for Cobalt service

- system call
- trap

RTDM

- character device
  - \* UDD (analogous to UIO), memory
- protocol device

tool

- ipipe latency tracer

valgrind / Helgrind

- not supported because of unknown system call

SIGDEBUG (SIGXPCPU): enable when RT thread enter RT time-critical phrases

functional limit

changes to critical subsystem regularly cause /7A

"Dovetail"

- co-kernel extension
- sharing of CPU traps

☐ SCHED\_DEADLINE: A Status Update - Juri Lelli, ARM Ltd

not only about deadline

since 3.14

real-time scheduling policy: higher priority than NORMAL and FIFO/RR

enables predictable task scheduling

- allow explicit per-task latency constraints
- avoid starvation (tasks can not eat all available CPU time)



- enrich scheduler's knowledge about QoS requirements
- policies
- EDF (Earliest Deadline First)
  - CBS (constant bandwidth server)
- resource (CPU) servation mechanism
- Q time unit (runtime) in every interval of length P (period)
- EDF + CBS provides temporal isolation
- load balancing + inheritance
- work with PREEMPT\_RT? orthogonal
- QoS
- under discussing
- bandwidth reclaiming

## ☐ [Understanding a Real-Time System - Steven Rostedt, Red Hat](#)

real fast vs. real time

- hot cache: look ahead features
- paging: TLB
- least interruptions
- optimize the most likely case: transactional memory

branch prediction

- deadline\_test

NUMA

- memory speeds dependent on CPU
- need to organize the tasks

Hyper-threading

- recommended to disable on RT

system management interrupt (SMI)

- put processor into system management mode (SMM)

RT kernel

- threaded interrupt
- system management threads
- high resolution timer

threaded interrupt

- don't poll network task (higher prio) on network interrupt

software interrupt

- network irq will run network softirqs

except for softirqs raised by real hard interrupt

- RCU
- timer
- run in ksoftirqd

timer

- setitimer(): requires ksoftirqd to run (on PREEMPT\_RT)
- timer\_create() / timer\_settime()

## NO\_HZ

- when CPU is idle, turn off timers
- let CPU go into deep sleep
- great for power saving

## NO\_HZ\_FULL

- works if only one task

## priority inheritance locking

- prevent unbounded latency
- pthread set protocol

## real-time vs. multi-process

- migrateion clear caches (memory and TLB)

# Keywords

---

- [Linux CNC](http://linuxcnc.org), <http://linuxcnc.org>
  - 3D Printer ([mails on LinuxCNC talking about the expired patent](#))
- [ROS](#), Movie: [Trajectory planning for 6 DOF Manipulator. using ROS \(HUBO Arm\)](#)
  - ROS.Taipei FB <[台北社群連結](#)>
  - ROS 2016年 八歲 紀念影片 <[youtube](#)>
- Xenomai, <https://xenomai.org/>
  - real-time linux, 馬達轉動補償, 消除抖動
- Real Time
  - Determinism
    - 行為可預測
  - Hard real-time
    - missing deadline = system failure
  - Soft real-time
    - values degrade after deadline
    - Linux 內建
  - Linux已支援Soft real-time
    - 例：影、音同步。
  - Real-Time Linux Workshops
- [Programmable Logic Controller-PLC](#)
  - [Modbus](#)
- OSADL
- Kernel Latency延遲
- Preemption
- Interrupt Handling
- Context Switch

- ISR ( Interrupt service routine )
- tickless
  - [what is a tickless OS?](#)
  - it's an OS that is immune to tickling.
- Preemptive Kernel
- SGI IRIX



SY J [\*It's a Unix System. I know This!\*](#)

- XFS
- OpenGL
- CFS
  - [https://en.wikipedia.org/wiki/Completely\\_Fair\\_Scheduler](https://en.wikipedia.org/wiki/Completely_Fair_Scheduler)
- POSIX (The **P**ortable **O**perating **S**ystem **I**nterface)
  - <https://en.wikipedia.org/wiki/POSIX>
  - posix 1003.1b - realtime extensions
- rcu preempt
  - [Real-Time Preemption and RCU](#)
- 眾數
- PDF - probability density function