? ? ? ? ? ? : RTlinux? ? ? ? ? ?

walker 04-12-02, 00:20

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ?

? ?

? , ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? , ? Intel, MIPS, ? ? ? ? ? ? ? ? ? ? 32? ?

? ? Linux? , ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? ? ? ? ? ? ? ? (Desktop Computing) , ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? , Linux ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? StrongARM , MIPS ? PowerPC? ? ? , Linux? ? ? ? , ? ? Linux ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? , ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? , ? Linux ? ? ? ? ? ? ? ? ,? ? ? ? ? ? ? ? ? ? , ?

? ? ? ? ? ? PDA ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? GUI? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Palm ? ? ? ? ? ? ? ? ? Windows CE ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ?

? ? ? ? ? ? ? Linux ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? , ? Linux? , ? ? ? ? ? ? ? ? ? ? ? Linux? ?

? Linux? RTLinux? ? ? ? ? ? , ? ? ? ? ? ? RTLinux? RTLinux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? : Linux; RTLinux; ? ? ? ? ; ? ? ? ? ? ?

? ?

Abstract i

? ? Ii

? ? III

? ? ? ? ? ? ? v

? ? ? VI

? ? ? ? ? ? ? ? ? ? ? ? 1

1.1 ? ? ? ? ? ? ? 1

1.1.1 ? ? ? ? ? ? ? ? ? ? 1

1.1.2 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? 2

1.1.3 ? ? ? ? ? ? ? ? 3

1.2 ? ? ? ? ? ? 4

1.2.1 ? ? ? ? ? ? ? 4

1.2.2 ? ? ? ? ? ? ? ? ? ? 4

? ? ? ? ? ? ? ? ? ? ? ? ? 6

2.1 LINUX? ? ? ? ? 6

2.2 LINUX? ? ? ? ? 8

2.2.1 ? ? ? ? ? ? 8

2.2.2 ? ? ? ? ? ? ? 11

2.3 ? ? ? ? ? ? ? ? ? ? 12

? ? ? ? ? ? ? ? Linux? ? RTLinux? ? ? ? ? ? 16

3.1 RTLinux? ? ? 16

3.2 ? ? ? ? 17

3.3 ? ? ? ? 19

3.3.1 ? ? ? ? ? ? ? ? ? 20

3.3.2 ? ? ? ? ? ? ? ? ? ? 21

3.4 ? ? ? ? 21

3.4.1 ? ? ? ? ? ? 21

3.4.2 ? ? ? ? ? ? ? ? ? ? 22

3.5 ? ? 22

3.5.1 ? ? ? ? ? ? 23

3.6 ? ? ? ? ? 23

3.6.1 FIFO? ? 23

3.6.2 ? ? ? ? 24

3.6.3 mbuff? ? ? ? 25

? ? ? rtlinux? ? ? ? ? ? 26

4.1 ? ? ? ? 26

4.2 ? ? API..... 26

4.2.1 POSIX? ? ? ? ? ? 26

4.2.2 ? ? ? ? ? ? 28

4.2.3 ? ? ? ? ? ? 29

4.3 ? ? ? ? 30

4.3.1 ? ? ? ? 30

4.3.2 ? ? ? ? ? 34

4.3.3 ? ? ? ? ? ? ? 35

? ? ? rtlinux? ? ? ? ? 37

? ? ? ? ? ? ? ? ? 38

? ? ? ? 40

? ? 42

? ? A..... 43

? ? B..... 59

? ? ? ? ? ? ?

? 2.1 ? ? ? ? ? ? ? ? ? ? ? 9

? 2.2 ? ? ? ? ? 10

? 3.1 RTLINUX? ? ? ? ? 17

? 4.1 ? ? ? ? ? 26

? 4.2 ? ? ? ? ? ? ? 30

? 2.1 ? ? ? ? ? ? ? ? ? 10

? 2.2 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? 11

? 2.3 ? ? ? ? ? ? ? 12

? 5.1 ? ? ? ? ? ? ? ? 37

? ? 2.1 ? ? ? ? ? ? ? 6

? ? 3.1 “ ? ” cli, sti? iret..... 18

? ? 3.2 rtl_thread_struct? ? 21

? ? ?

API Application Program Interface

ATM Asynchronous Transfer Mode

CPU Central Processor Unit

DMA Direct Memory Access

EDF Eariest Deadline First

FIFO First-In-First-Out

GUI Graphical User Interface

I/O Input/Output

IPC Interprocess Communication

ISR Interrupt Service Routine

IST Interrupt Service Thread

MCU Micro-Controller Unit

OS Operating System

PC Personal Computer

PDA Personal Data Assistant

POS Point Of Sells

POSIX Portable Operating System Interface for computer Environments

RAM Random-Access Memory

ROM Read-Only Memory

RT Real Time

RTOS Real Time Operating Systems

TLB Translation Lookaside Buffer

? ? ? ? ? ? ? ? ? ? ? ? ?

1.1 ? ? ? ? ? ? ? ?

1.1.1 ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? 30? ? ? ? ? ? , ? 90? ? ? , ? , ? , ? , ? Internet? ? ? ? ? ? ? ? , ? ? ? ? ? (Consumptive electron),? ? ? (Computer),? ? (Communication),3C? ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? 4? ? ? [1]?

I ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? : ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? , ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ?

I ? ? ? ? ? ? ? ? ? CPU? : CPU? ? ? ? , ? ? ? ? ? ? ; ? ? ? ? ? , ? ? ? ; ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ; ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ; ?

I ? : ? , ? ? ? ? ; ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ; ? ; ? ? ? ? ? ? ? ? ? ? ? ? (API) , ? ? ? ? ? ? ? ? ? ; ? ? ? ? ? ? ? ? ? ? ?

I ? ? ? ? ? ? ? ? Internet? ? ? ? ? ? ? ? ? , ? Internet? ? , ? ? ? Internet? ? ? ? ? Internet? , ? ? ? ? ? ? Internet? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

1.1.2 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? “ ? ? ” ? (MCU)? ? ? ? ? ? ? ? ? CPU? ? ? ? 8? ? 16? ? , ? 32? ? 64? ? ? ? ? ? ? ? ? ? , ? (ROM? RAM?) ? ? ? ? ? ? IC? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? (Flash Memory) ?

? ? ? ? , ? , ? , ?

? PC? ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ? , ? , ? ? ? ? ? ? ? ? ? POS/ATM? ? IC? ? ; ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? WebTV? ; ? , ? , ? ? ? ? ? ? ? ? ?

1.1.3 ? ? ? ? ? ? ? ?

? , ? ? 4? ? ? ? ? ? ? WinCE[18]? Palm OS[19]? EPOC[20]? Linux[21]? ? ? ? ? , ?

? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? , 2000? ? 2004? ? ? ? ? ? ? ? ? 77.4% , ? ? ? ? ? ? ? (PDA) ? , ?

? , ? WinCE? ? ? ? ? ? ? ? ? ? ? ? ? ? Palm OS? ? ? ? ? ? ? ? PDA? ? 70% ? ? ? ? ? ? ; ? ? ? ? 3COM? IBM? ? ? ? ? ? ? ? ? ? ? ? EPOC? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? 3? , ? ? ? ? ? ? ? ? ? ? ? ; ? 3? ? ? ? ? ? ? ? ? , EPOC? ? ? ? ? ? , ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ?

? ? , ? 3? ? ? ? ? ? ? ? ? ? , Linux? ? ? ? ? ? ? ? ? ? ? ; ? ? Linux? ? ? ? , ? ? ? ? ? ? ? ? , ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? Linux? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ? , ? ? ? ? ? Linux? ? ? ? ? ? , ? Linux? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? , Linux? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? ? CPU, ? ? Intel? MIPS? ASIC? ALPHA? 68K? POWER PC? ? ? ? Linux? , ? ? ? ? ? ? ? , ? ? ? ? ?

1.2 ? ? ? ? ? ?

1.2.1 ? ? ? ? ? ? ?

? , ? ? ? ? ? ? ? , ? , ?

? ? ? ? [5]? ? ? ? ? ? ? ? ? ? ? , ? , ? , ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? “ ? ? ” ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? (hard real-time task) ? ? ? ? ? ? ? ? ? ? ? ? , ? (soft real-time task) ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ?

? , ? ? ? ? ? ? ? ? ? , ? (aperiodic task) ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? (periodic task) , ? ? ? ? ? ? ? "? ? ? ? T? ? "? ? "? ? T? ? ? "?

1.2.2 ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? [5]:

l ? ? ? ? : ?

l ? ? ? : ?

l ? ? ? ? : ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? (? ? ? ? ? ? ? ? ? ? ? ?) ?

l ? ? ?

l ? ? ? ? ? ? : ?

? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? (RTOS)? ? , ? RTOS? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? RTOS, ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? RTOS? ? ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? UNIX? UNIX? ? , ? ? Windows NT, ? ? ? POSIX.1003? ? ? ? , ? ? ? ? ? ? ? ? ? UNIX? POSIX1003.1b? ? ? ? ? ? ? ? ? , UNIX? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? UNIX? ? ? , ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

2.1 LINUX? ? ? ? ?

UNIX? ? ? ? ? ? ? ? ? ? ? ? ? ? [17]? LINUX? ? UNIX? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? (? ? 2.1) ?

```
#define DELAY 10000

main()

{

int i;

while (1) {

for (i=0;i

speaker_on();
```

```
for (i=0;i
speaker_off();
}
}
```

? ? 2.1 ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? on? off? , ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? LINUX? ? ? , ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? 412MHz Celeron? ? ? ? Linux? , ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? , ? ? X-Windows, ? , ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

Linux? ? ? ? ? ? ? ? ? ? ? ? UNIX? ? ? ? [12]? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? , ? ? ? ? ? ? , ?

? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? [4]?

? ? ? ? UNIX? ? ? ? , ? ? ? Linux? ? ? ? , ? CPU? ? ? ? [16]? CPU? ? , ? ? /? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

Linux? ? ? ? ? ? ? ? ? (time slices)? ? ? CPU? , ? ? ? ? ? ? CPU, ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? I/O? , ?

Linux? ? ? ? POSIX? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? (SCHED_FIFO)? ? ? ? ? ? ? ? (SCHED_RR)? ? SCHED_RR? ? ? , ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? SCHED_FIFO? ? ? ? ? ? ? ? ? ? SCHED_FIFO? ? ? ? ? ? ? ? , ? SCHED_RR? ? ? ? ? ? ? ? ?

? sleep()? ? ? ? ? ? 1? ? ? ? , ? , ? ? , ?

? ? ? ? Linux? ? ? ? ? ? ? ? ? ? [10]? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ?

? , ? ? ? ? ? ? ? ? ? ? , ? (semaphores)? ? ? ? ? (spinlocks)? ? ? ? ? ? ? ? ? ? ? ? ? ? , ?

Linux? ? ? ? ? ? ? ? ? ? ? ? [10]? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? RAM? , ? ? ? ? ? ? ? ? ? ? ? ? RAM? , ? ? ? ? ? ? ? ? ? , ?

? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? Linux? ?

2.2 Linux? ? ? ?

? ? ? Linux? ? ? ? ? ? ? ? ? ? ? , ? ? Linux? , ? ? ? ? ? Linux? ? ? ? ? ? ? ? ?

2.2.1 ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? : ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? 2.1? ?

? 2.1 ? ? ? ? ? ? ? ? ? ? ?

? , ? Linux? , ? ? ? ? ? ? ? ? ? ? ? /? ? ? , ? ? ? ? ? ? ? ? __cli()/__sti()? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? __cli()/__sti()? ? ? ? ? ? ? ? ? ? ? __cli()? , ? ? ? ? ? ? ? , ? ? __sti()? Linux? ? ? ? ? ? ? ? ? 2.2? ? :

? ? ? ? ? ? ? ? ? ? ? ? ? __cli()/__sti()? , ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? (? ? ? ? ? ? ? ? ? ? ? ? A)

? ? Linux? ? ? ? ? 3? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? 500? ? ? ? 2.1? 2.2? ? ? ? ? ?

? 2.2 ? ? ? ? ?

? 2.1 ? ? ? ? ? ? ? ? ?

? 2.2 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? 496? ? , ? ? ? ? ? ? ? ? ? ? 250? ? ? 300? , ? ? ? ? ? ? ? ? ? ? ? ? : Linux? ?

2.2.2 ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? (? ? ? ? ? B) ? ? ? ? ? ? , ? UNIX? , ? 2000? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? : ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? , ? 2.3 ? ? , Y? ? ? ? ? ? ? ? , X? ? ? ? ? ? ? ? , size? ? ? ? ? ? ? ?

? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? , ? 16k? ? ? ? ? ? ? ? , ? , ? ? 16k? , ? ? ? ? ? ? ? , ? ? ? ? ? ? 64k? , ? ? ? ? ? ? 300? ? ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? 300? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? 2.3 ? ? ? ? ? ? ? ?

2.3 ? ? ? ? ? ? ? ? ?

? ?

? [12]? ? ? ? ? ? MINIX? round-robin? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? MINIX? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? UNIX? ? ? ? ? POSIX.1b-1993? , ? ? ? ? ? ? ? ? ? , ? ? ? ? , ? ? ? IPC? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? UNIX? ? ? ? ? ? ? ?

Linux? ? ? ? ? POSIX.1b? ? ? ? ? ? ? [12]? ? 1997? 5? 1? ? , ? Linux? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , POSIX.1b? ? ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? POSIX.1b? ? ? ? ? ? QNX[26]? QNX? 4? ? ? : ? ? ? ? , ? ? ? ? ? , ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? (? ? 7KB? ?) ? ? ? ? ? ? ?

QNX? ? POSIX 1003.1? ? (? ? ? ?) ? POSIX 1003.2? ? (? ? ? ? ? ? ? ?) ? ? ? ? ? UNIX? ? ? ? ? ? , ? ? ? ? ? ? ? QNX? ? ? ? ? UNIX? ? : ? ? ? , ? ? ? , X-Windows, ? TCP/IP?

? (? QNX? ?) , ? , QNX? ? ? ? ? 100KB? ? ? ? ? ROM? ? ? , ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? , ? IPC? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ?

? , ? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? [2], ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? VxWorks[27]? VxWorks? ? ? ? ? ? ? ? ? ? , ? ? ? ? /? ? ? ? ? ? ? ? ? UNIX? wind? VxWorks? TCP/IP? ? ? ? ?

? ? VxWorks? ? ? UNIX? ? , ? ? ? ? ? ? ? ? ? POSIX? ? ? ? ? ? ? , ? ? ? POSIX.1b? ? ? ? ? ? ? ? ? ? ? VxWorks API? ? ? ? ?

? VxWorks? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? , ? C? ? ? shell? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? , ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

REAL/IX? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? UNIX? ? ? ? ? ? ? , ? ? ? UNIX System V, ? , ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? sleep/wakeup? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

REAL/IX? POSIX.1003? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? UNIX? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? I/O, ? ? ? IPC? ? ? ? , ?

? ? ? ? ? ? ? ? ? ? Windows NT? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Windows, ? ? , ? : ? ? ? ? , ? ? , ? Win32 API? Microsoft? ? ? ? ? ? ? ? ? ? ? ?

? ? ? [25]? ? ? ? ? , Windows NT? ? ? ? ? ? ? ? ? ? ? ? ? : Win32 API? ? ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? , Windows NT? ? ? ? ? ? ? ? ? ? ? ? ?

Microsoft? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Windows CE? ? ? ? ? ? ? ? ? ? , Windows CE? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Windows CE? ? ? ? ? ROM? , ? ROM? ? , ? ? ? ? ? ? ? RAM? ? ? ? ? Windows CE? ? ? ? ? ? ? ? ? : ? ? ? ? ? ? ? ? ? ? ? ? ? ISR(interrupt service routine); ? ? ? ? ? ? ? ? ? ? IST(interrupt service thread)? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? IST? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? IST? ? ? ? ? ? ? ? ? ? ? ? ? Win32 API? ? ? ? ? ? ? ? ? ? ? Windows? ? ? ? ? ? ? ? ? Windows? ? ?

? ? ? ? ? ? ? ? LINUX? ? RTLINUX? ? ? ? ? ?

? ? ? ? ? ? ? , ? ? ? ? ? ? Linux? ? ? ? ? ? ? ? ? , ? , Linux? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? , ? ? ? Linux? ? ? ? ? , ? ? ? ? ? ? ? RTLinux? ? ? ? ? ? ? RTLinux? ? ? ? ? ?

3.1 RTLinux? ? ?

RTLinux? Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? , ? ? Linux ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? ? ? ? ? ? ? , ? Linux? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? Linux ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? FIFOs? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ; ? ? ? ? ? ? ? ? ? ? RMS? EDF? ? ?

? ? ? ? ? ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? , ? CPU? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? : ? RTLinux? ? ? ? ? ? ? ? 3.1? ? ?

? 3.1 RTLinux? ? ? ? ?

3.2 ? ? ? ?

? ? ? ? Linux? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? Linux? (i486? ? ? ? cli? sti? ? ? ?) ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? Linux? , ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? Linux? , ? ? ? ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? [6]? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? ? ? cli, sti, ? iret(iret: ? ? ? ? ? ? ?) ? ? ? ? ? ? ? ? ? : S_CLI, S_STI? S_IRET? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

```
/* These are macros */

S_CLI: movl $0, SFIF

S_IRET: push %ds
pushl %eax
pushl %edx
```

```
movl $KERNEL_DS, %edx
mov %dx, %ds
cli
movl SFREQ, %edx
andl SFMASK, %edx
bsrl %edx, %eax
jz not_found
movl $0, SFIF
sti
jmp SFIDT ( , %eax, 4)
not_fount:
movl $1, SFIF
sti
popl %edx
popl %eax
pop %ds
iret

S_STI: pushfl
pushl $KERNEL_CS
pushl $done_STI
S_IRET
done_STI:
```

? ? 3.1 “ ? ” cli, sti? iret

? ? ? ? ? ? , ? , ? (Linux? ? ? ? ? ?) , Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? ? ? ? , ?

? ? Linux? ? ? ? ? ? ? ? ? ? ? , Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

S_CLI? S_STI? S_IRET? ? ? ? 3.1? ? ? ? ? ? ? ? ? GNU? ? ? ? ? S_CLI? ? ? ? ? ? ? ? ? ? ? , ? ? Linux? ? ? ? ? S_STI? ? ? ? ? ? ? ? ? ? ? ? ? ? S_IRET? ? ? ? ? ? ? ? S_IRET? ? ? ? ? ? ? ? iret? ? ? ? ? ? ? ? ? ? ? ? ?

S_IRET? , ? ? ? ? ? ? ? ? ? , ? , ? ? ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? , ? , ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? S_STI? ? S_IRET? ? ? ? ? ? ? ? ? ?

? Linux? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ?

3.3 ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? , ?

? 80x86? ? ? ? ? ? ? ? ? ? [10]? ? ? ? ? ? ? ? ? ? ? , ?

? ? ? ? ? ? ? ? ? ? , ? ? ? TLB? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux ? ? ? ? ? ? ? ? ? : ? , ? : init_module()? cleanup_module()? init_module()? ? ? ? ? ? ? ? ? ? ? , cleanup_module()? Linux? ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? RTLinux? : ? C? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ?

? TLB? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? C? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? C? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ?

? ?

3.3.1 ? ? ? ? ? ? ? ?

```
struct rtl_thread_struct

struct rtl_thread_struct {

int *stack;

int fpu_initialized;

RTL_FPU_CONTEXT fpu_regs;

int uses_fp;

int *kmalloc_stack_bottom;

struct rtl_sched_param sched_param; /* ? ? ? ? ? ? ?  */

struct rtl_thread_struct *next; /* ? ? ? ? ? ? ? ? ?  */

int cpu; /* ? ? ? CPU?  */

hrtime_t resume_time; /* ? ? ? ?  */
```

```
hrtime_t period; /* ? ? ? ?  */
hrtime_t timeval;
struct module *creator; /* ? ? ? ? ?  */
void (*abort)(void *);
void *abortdata;
int threadflags;
rtl_sigset_t pending;
rtl_sigset_t blocked;
void *user[4];
int errno_val;
struct rtl_cleanup_struct *cleanup;
int magic;
struct rtl_posix_thread_struct posix_data;
void *tsd [RTL_PTHREAD_KEYS_MAX];
};
```

? ? 3.2 rtl_thread_struct? ?

3.3.2 ? ? ? ? ? ? ? ? ?

? , ? ? ? ? ? ? ? ? ? ? ? ? ? RTLinux? , ? ? ? ? ? ? ? Linux ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

3.4 ? ? ? ?

? [4]? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ?

? RTLinux? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

3.4.1 ? ? ? ? ? ?

? , ? , ? CPU?

? (? ? ? ?) ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? (rate monotonic scheduling algorithm, RMS)

[5]? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? n? ? ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? (deadline):

? ? , Ci? ? ? i? ? ? ? ? ? ? ? ? ? ? ? ? , Ti? ? ? i? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? [5]?

? ? ? ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? Linux? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? (Earliest Deadline First, EDF)? (deadline)? ? ? ? ? ? ? ? ?

3.4.2 ? ? ? ? ? ? ? ? ? ?

RTLinux? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? :

l ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ;

l ? ? ? ? ? ? ? ? ? ? ? ? ;

l ? ? ? ? ? ? ? ? ? ? ;

l ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ;

RTLinux? ? ? ? ? ? ? rtl_sched.c? rtl_schedule.h? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? pthread_attr_setschedparam? ? ? ? ? ? ? pthread_setchedparam? ? ? ? ? ? ? scheduler? ? ? ? ? ? ? ? ? -1, ? ? ? ? ? ? ? ? ? ? ? ? ? ? 0, ? rtl_schedule(? ? ? ?) ? ? ; ? ? ? ? ? ? ? ? ? ? : schedule_t? rtl_thread_struct; ? ? ? ? ? task_queue *list, task_queue *destlist?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

3.5 ? ?

? , ? ? ? ? ? ? ? ? ? (task release jitter)[24]? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? [24]? ?

? Linux? ? ? ? ? ? ? IBM? ? ? PC? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? 100Hz? ? ? ? ? , ? ? ? ? ? ? ? 10? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? VxWorks? REAL/IX? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? RTLinux? , ? CPU? ? ? ? , ? Intel 8354? ? ? ? ? ? ? ? interrupt-on-terminal-count? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? 1? ?

? Linux? : ? ? ? ? ? ? ? ? ? ? ? , ? , ? ? ? sti? ? ? , ? ? ? ? ? ? ? ? ?

3.5.1 ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

3.6 ? ? ? ? ?

? ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? (IPC)? ? ? ? RTLinux? ? ? ? ? ? ? ? ? ?

3.6.1 FIFO? ?

RTLinux? FIFO? ? ? ? Linux? ? ? ? Linux? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? (real-time FIFOs) , ? ? ? UNIX IPC? ? ? ? ? ? ?

RT-FIFO? RT-FIFOs? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? ? RT-FIFOs? ? ? ? ? ? ? ? ? ? ? ? FIFO? ? FIFO? ? ? ? ? ? ? ? ? ? , ?

? Linux? ? ? , ? RT-FIFOs? ? ? ? ? ? ? ? ? , ? (API)? ? ? ? ? ? Linux? ? ? ? ? ? ? ? ? ? ? ? , ? ? : open, close, read? write?

```
struct file_operations rtf_fops

static struct file_operations rtf_fops =

{

rtf_llseek,

rtf_read,

rtf_write,

NULL,

rtf_poll,

NULL,

NULL,

rtf_open,

NULL,

rtf_release,

NULL,

NULL,

NULL,

NULL,

NULL

};
```

? ? ? ? ? ? ? ? ? ? :

```
int rtf_create(unsigned int minor, int size)

int rtf_destroy(unsigned int minor)
```

```
int rtf_put(unsigned int minor, void *buf, int count)

int rtf_get(unsigned int minor, void *buf, int count)
```

? ? ? ? FIFO? ? ? ? ? ?

```
int rtf_create_handler(unsigned int minor,

int (*handler) (unsigned int fifo))
```

? ? ? ? ? ? ? ? ? , ? ? ? ? FIFO? ? ? ? ? ? ? ? ? ?

3.6.2 ? ? ? ?

? RTLinux? ? ? ? ? , ? ? ? ? ? ? ? ? mem? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? RTLinux? ? ? ? ? /dev/mem? ? ? ? ? ? ? ? ? ? , Linux? , ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? ? ? ?

3.6.3 mbuff? ? ? ?

? ? ? Tomasz Motylewski? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? mbuff? ? ? mbuff_alloc()? ? ? ? ? ? ? ? ? ? ? ? ? ? , mbuff? RTLinux? Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

```
#include

void * mbuff_alloc(const char *name, int size);

void mbuff_free(const char *name, void * mbuf);
```

? ? ? ? ? mbuff_alloc? , ? ? ? ? ? ? , ? 1? NULL? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? 1?

? ? ? RTLinux? ? ? ? ? ?

4.1 ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? : ? ? ? ? ? ? ? ? ? ? ? [2]? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ; ? ? ? ? ? ? ? ? ? ? ? , ? , ?

? 4.1? ?

? 4.1 ? ? ? ? ?

4.2 ? ? API

4.2.1 POSIX? ? ? ? ? ?

? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? RTLinux ? , ? ? ? ? ? ? ? Linux? ? ? ? ? ? ?

int pthread_create (pthread_t *thread, pthread_attr_t * attr, void * (* start_routine)(void *), void *arg)

? ? RTLinux? ? ? POSIX? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? start_routine? ? ? ? ? , arg? attr? ? ? ? , ? ? ? ? ? ? ? ? ? CPU? ? ? ? ? ? ? ? ? ? ? ? ? ? ? NULL, ? ? ? ? ? ? ? ? ? ? ? 0? ? ? ? ? ? ? ? ? , ? ? ? ? ? thread? ? ? ? ? ? ; ? ? ? 0? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? CPU? ? ? ? ? (pthread_attr_setcpu_np), ? ? ? ? FPU(pthread_attr_setfp_np)?

int pthread_attr_init (pthread_attr_t *attr)

? ? ? ? ? ? ? ? ? ? ?

int pthread_ attr_setschedparam (pthread_attr_t *attr, const struct sched_param *param)? int pthread_ attr_setschedparam (const pthread_attr_t *attr, struct sched_param *param)

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? attr? ? ? /? ? ? ? ? ? ? ? ? ? ? param? ? ? ? ? SCHED_FIFO? SCHED_RR? ? ? ? ? ? ? ? ?

int pthread_attr_setcpu_np (pthread_atte_t *attr, int cpu)?

int pthread_attr_getcpu_np (pthread_atte_t *attr, int cpu)

? ? /? ? ? ? ? ? ? CPU? ? ? SMP? ? ? ? ? ? ? ? ? ? ? ? ? ? ? CPU? ? ? ?

int pthread_cancel (pthread_t thread)

? ? ? ? ? ? ? ? ? ?

int pthread_delete_np (pthread_t thread)

? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? 0? ? ? ? ? ? ? , ? 0? ? ? ? ? ? ? ?

pthrad_t pthread_self (void)

? ? ? ? ? ? ? ? ? ? ? ? ? ?

clockid_t rtl_getschedclock (void)

? ? ? ? ? ? ? ? ? ? ? ? ?

int rtl_setclockmode (clockid_t clock, int mode, hrtime_t mode_param)

? ? ? ? ? ? ? ? ? , mode=RTL_CLOCK_MODE_ONESHOT? ? ? ? ? (? ? ?) ? ? mode_param? ? ? ? ;

mode=RTL_CLOCK_MODE_PERIODIC? ? ? ? ? ? , mode_param? ? ? ? ? ? ? ? ? (? ? ? ? ? ? ? ? 3.4? ? ? ?)

int pthread_wait_np (void)

? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? 0?

4.2.2 ? ? ? ? ? ?

RTLinux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? , ? ? TSP(timestamps)? ?

? ? ? ? ? ? ? ? ? ? ? :

/* #include */

int clock_gettime(clockid_t clock_id, struct timespec *ts);

hrtime_t clock_gethrtime(clockid_t clock);

struct timespec {

time_t tv_sec; /* ? */

long tv_nsec; /* ? ? */

};

clock_gettime: ? ? ? ? ? ? ? , ? ? ? clock_id? ? ? ? ? ? ?

clock_gethrtime: ? ? ? ? ? ? , ? ? ? ? ? ? 64? (hrtime_t)? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? , ?

? ? ? ? ? ? :

/* #include */

hrtime_t timespec_to_ns(const struct timespec *ts); /* timespec? ? ? ? ? ? */

struct timespec timespec_from_ns(hrtime_t t); /* ? ? ? ? timespec? ? */

const struct timespec * hrt2ts(hrtime_t value); /*

? ? ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? :

l CLOCK_MONOTONIC: POSIX? ? , ? ? ? ? ? ? ? ; ? ? ? ? ? ? ?

l CLOCK_REALTIME: ? ? POSIX? ? ? ? ? ? ? ? CLOCK_MONOTONIC? ? ? ?

l CLOCK_RTL_SCHED: ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? :

l CLOCK_8254: ? x86? ? ? ? ? ? ? ? ? ? ? ? ? ?

l CLOCK_APIC: ? ? SMP x86? ? ? ? ?

4.2.3 ? ? ? ? ? ?

RTLinux? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? RTLinux? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? RTLinux? ? ? ? ? ? ? API:

int pthread_setschedparam (pthread_t thread, int policy, const struct sched_param *param)

? ? ? ? ? ? ? ? ? ? ? , ? policy? sched_param? ? ? ? ? ? thread? ? ? ? ? ? ? :

policy=SCHED_RR: ? ? Round-Robin? ? ? ?

policy=SCHED_FIFO: ? ? ? ? ? ? ? ? ? ? ?

? ? 0? ? ? ? ? ? , ? 0? ? ? ? ?

int pthread_getschedparam (pthread_t thread, int policy, const struct sched_param *param)

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? policy? sched_param? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

int pthread_make_periodic_np (pthread_t thread, hrtime start_time, hrtime_t period)

? ? ? ? ? ? thread? ? ? ? ? ? ? ? ? ? ? start_time? ? ? ? ? ? , ? ? ? ? ? ? ? ? period? ? ?

int pthread_wait_np (void)

pthread_wait_np? pthread_make_periodic_np? ? ? ? ? ? ? ? ?

int sched_get_priority_max (int policy)?

int sched_get_priority_min (int policy)

? ? sched_priority? ? ? ? ?

4.3 ? ? ? ?

? ? ? ? ? RTLinux? ? ? API, ? ? ? ? ? ? ? ? ? ? ? RTLinux? ? ? ? ? ? ? ? ? ? ? ? RTLinux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? FIFO? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ?

? 4.2 ? ? ? ? ? ? ?

4.3.1 ? ? ? ?

init_module? ? ? ? ? ? ? ? ? ? ? cleanup_module? ? ? ? ? ? ? ? ? ? ?

/* * RTLinux scheduling accuracy measuring example */

```
#include
#include
#include
#include
#include
#include
#include
#include
#include
#include "common.h"

int ntests=500;
int period=1000000;
int bperiod=3100000;
int mode=0;
int absolute=0;
int fifo_size=4000;
int advance=0;

MODULE_PARM(period,"i");
MODULE_PARM(bperiod,"i");
MODULE_PARM(ntests,"i");
MODULE_PARM(mode,"i");
MODULE_PARM(absolute,"i");
MODULE_PARM(advance,"i");

pthread_t thread;
int fd_fifo;
```

```
void *thread_code(void *param)
{
hrtime_t expected;
hrtime_t diff;
hrtime_t now;
hrtime_t last_time = 0;
hrtime_t min_diff;
hrtime_t max_diff;
struct sample samp;
int i;
int cnt = 0;
int cpu_id = rtl_getcpuid();

rtl_printf ("Measurement task starts on CPU %d\n", cpu_id);
if (mode) {
int ret = rtl_setclockmode (CLOCK_REALTIME, RTL_CLOCK_MODE_PERIODIC, period);

if (ret != 0) {
conpr("Setting periodic mode failed\n");
mode = 0;
}
} else {

rtl_setclockmode (CLOCK_REALTIME, RTL_CLOCK_MODE_ONESHOT, 0);
}

expected = clock_gethrtime(CLOCK_REALTIME) + 2 * (hrtime_t) period;


fd_fifo = open("/dev/rtf0", O_NONBLOCK);
if (fd_fifo < 0) {
```

```
rtl_printf("/dev/rtf0 open returned %d\n", fd_fifo);

return (void *) -1;

}



if (advance) {

rtl_stop_interrupts(); /* Be careful with this! The task won't be preempted by anything else. This is probably only appropriate for small high-priority tasks. */

}



/* first cycle */

clock_nanosleep (CLOCK_REALTIME, TIMER_ABSTIME, hrt2ts(expected - advance), NULL);

expected += period;

now = clock_gethrtime(CLOCK_MONOTONIC);

last_time = now;



do {

min_diff = 2000000000;

max_diff = -2000000000;



for (i = 0; i < ntests; i++) {

++cnt;

clock_nanosleep (CLOCK_REALTIME, TIMER_ABSTIME, hrt2ts(expected - advance), NULL);



now = clock_gethrtime(CLOCK_MONOTONIC);

if (absolute && advance && !mode) {

if (now < expected) {

rtl_delay (expected - now);

}

now = clock_gethrtime(CLOCK_MONOTONIC);

}

if (absolute) {

diff = now - expected;

} else {

diff = now - last_time - period;
```

```
if (diff < 0) {
diff = -diff;
}
}
if (diff < min_diff) {
min_diff = diff;
}
if (diff > max_diff) {
max_diff = diff;
}

expected += period;
last_time = now;
}

samp.min = min_diff;
samp.max = max_diff;
write (fd_fifo, &samp, sizeof(samp));
} while (1);
return 0;
}

pthread_t background_threadid;

void *background_thread(void *param)
{
hrtime_t next = clock_gethrtime(CLOCK_REALTIME);
while (1) {
hrtime_t t = gethrtime ();
next += bperiod;
/* the measurement task should preempt the following loop */
while (gethrtime() < t + bperiod * 2 / 3);
clock_nanosleep (CLOCK_REALTIME, TIMER_ABSTIME, hrt2ts(next), NULL);
}
```

```
}



int init_module(void)
{
pthread_attr_t attr;
struct sched_param sched_param;
int thread_status;
int fifo_status;

rtf_destroy(0);
fifo_status = rtf_create(0, fifo_size);
if (fifo_status) {
rtl_printf("RTLinux measurement test fail. fifo_status=%d\n",fifo_status);
return -1;
}



rtl_printf("RTLinux measurement module on CPU %d\n",rtl_getcpuid());
pthread_attr_init (&attr);
if (rtl_cpu_exists(1)) {
pthread_attr_setcpu_np(&attr, 1);
}
sched_param.sched_priority = 1;
pthread_attr_setschedparam (&attr, &sched_param);
rtl_printf("About to thread create\n");
thread_status = pthread_create (&thread, &attr, thread_code, (void *)1);
if (thread_status != 0) {
rtl_printf("failed to create RT-thread: %d\n", thread_status);
return -1;
} else {
rtl_printf("created RT-thread\n");
}
```

```
if (bperiod) {
pthread_create (&background_threadid, NULL, background_thread, NULL);
}
return 0;
}


void cleanup_module(void)
{
rtl_printf ("Removing module on CPU %d\n", rtl_getcpuid());
pthread_cancel (thread);
pthread_join (thread, NULL);
close(fd_fifo);
rtf_destroy(0);
if (bperiod) {
pthread_cancel (background_threadid);
pthread_join (background_threadid, NULL);
}
}
```

4.3.2 ? ? ? ? ?

? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? /dev/rtf0? ? ? ? ? ? ? ? ? ? ?

```
#include
#include
#include
#include
#include
#include
#include
#include
#include "common.h"
```

```
int main()
{
int fd0;
int n;
struct sample samp;
if ((fd0 = open("/dev/rtf0", O_RDONLY)) < 0) {
fprintf(stderr, "Error opening /dev/rtf0\n");
exit(1);
}


while (1) {
n = read(fd0, &samp, sizeof(samp));
printf("min: %8d, max: %8d\n", (int) samp.min, (int) samp.max);
fflush(stdout);
}

return 0;
}
```

4.3.3 ? ? ? ? ? ? ?

? ? ? Celeron 412MHz，196MB? ? ，RTLinux3.1? ? ? ? ? ? ? ? ? Makefile? ? :

```
all: rt_process.o irqsema.o monitor histplot

include ../../rtl.mk

monitor: monitor.c
$(CC) ${USER_CFLAGS} ${INCLUDE} -Wall -O2 -o monitor monitor.c

clean:
rm -f *.o monitor histplot periodic_monitor gnuplot.out
```

include $(RTL_DIR)/Rules.make

? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? :

min: 0, max: 24480

min: 0, max: 10720

min: 0, max: 10912

min: 0, max: 10976

min: 0, max: 11072

min: 0, max: 10656

min: 0, max: 10944

min: 0, max: 11200

min: 0, max: 11200

min: 0, max: 11008

min: 0, max: 10912

.....

? ? ? RTLINUX? ? ? ? ?

? ? 2.2? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? RTLinux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Celeron 412MHz, 196MB? ? , RTLinux3.1+Linux-2.2.19(? ? ? ? ? ? Linux? ? ? ?) ? ? ? ? ? ? ? ? ? ? (? ? ? ? ? ? ? ? 2) ? ? ? ? ? ? ? ? 5.1 ? ? :

? ? ? ?

? ? ?

? ? ?

? ? ?

? ? ?

2.36

2.10

15.50

? ? ? ? ? ?

3.20

2.10

19.40

? ? ? ?

2.56

2.20

14.50

? 5.1 ? ? ? ? ? ? ? ? (? ? : ? ?)

n ? ? ? : ? ? ? ? ? ? ? ?

n ? ? ? ? ? ? : ? ? ? ? ? ? ? ? shell? ? ? ? ?

n ? ? ? ? : ? ? ? ? ? ? ? ? ? ? ? C? ?

? ? ? ? ? ? , RTLinux? ? ? ? ? ? ? ? ? ? ? ? ? Linux? 19.40? ? , ? ? ? ? ? ? ? ? , ? ? Linux? ? ? ? ? ? 500? ? ?

? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? 10? ? ? ? ?
? ? ? ? ? , RTLinux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ?

? ? , ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? , ? ? ? ?

? ? ? ? ? ? ? ? , ? ! ? , ? ? ? ? ? ? ? ? ? ?

? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ?

? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? Windows CE, PalmOS ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? , ? , ?

? ? ? ? ? ? ? ? ? ? ? ? ? Linux, ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux:

l ? , ?

l ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? , ? ? ? ? ? ? ? ?

l ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? Windows CE ? ? ? ? ? ? ? Linux? ? ? ?

? ? , ? ? ? ? ? ? ? ? ? ? Linux ? , ? ? ? ? ? Linux ? , ? Linux ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? :

l Linux ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? CPU ? ? ? ? ? ? ? ? ? ? ? ? ? ? CPU ? , ? ? ? ? ? ? ? ? ? ? Linux ? ? , ? ? ? ? ? ? ? ? ? ? ?

l ? ? ? ? ? Linux ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ?

l ? ? ? ? ? ? ? ? Linux ? ? ? ? ? ? ? ? , ?

l ? ? ? ? Linux ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ?

? ? ? ?

[1] ? ? ? ? ? ? ? ? ? ? ? ? . ? ? ? ? ? . 2000? ? 49?

[2] ? ? ? . ? ? ? Linux? ? ? ? ? . ? ? ? ? ? ? ? . 2002.1

[3] ? ? ? , ? ? ? . ? ? ? ? Linux? ? ? ? ? ? ? ? ? ? . ? ? ? ? ? ? ? ? . 2001.7: 64-85.

[4] D. M. Dhamdhere. Systems Programming and Operating Systems. ? ? ? ? ? ? ? . 2001.9

[5] William Stallings. Operating Systems Internals and Design Principles. ? ? ? ? ? ? ? . 2001.6

[6] H. Lyckama & L. Bayer. UNIX time-sharing system: The MERT Operating System. Bell System Technical Journal. 57(6):2049-2086, 1978.

[7] Alessandro Rubini. Linux? ? ? ? . ? ? ? ? ? ? ? . 2000.4

[8] ? ? ? ? ? ? . ? ? ? ? ? ? . 2000.11

[9] ? ? ? , ? ? ? ? ? . 2001? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? . ? ? ? ? ? ? ? ? ? ? ? . 2001.10

[10] Daniel P.Bovet & Marco Cesati. ? ? ? ? Linux? ? . ? ? ? ? ? ? ? . 2001.10

[11] Andrew S. Tanenbaum & Albert S. Woodhull. Operating Systems Design and Implementation. ? ? ? ? ? ? ? . 2001.4

[12] Scott Maxwell. Linux? ? ? ? ? ? ? . ? ? ? ? ? ? ? . 2000.6

[13] Jean J.Labrosse. uC/OS the Real-Time Kernel. R&D Publications,1992

[14] Michael Barabanov. A Linux-based Real-Time Operating System[D]. Master Paper. New Mexico Institute of Technology, 1997.6

[15] Victor Yodaiken, Michael Barabanov. A Real-Time Linux. New Mexico Institute of Technology

[16] Maurice J.Bach. The Design of the UNIX Operating sytem. ? ? ? ? ? ? ? . 2000.4

[17] Dennis W.Ritchie & Ken Thempson. The UNIX time-sharing System. Communications of the Association for computing Machinery, 17(1):365-375, July 1974.

[18] Gabriel A.Wainer. Implementing real-time Service in MINX. Operating Systems Review, 29(3):75-84, July 1995.

[19] http://www.microsoft.com/

[20] http://www.palmos.com/

[21] http://www.epoccity.com/

[22] http://www.linux.org/

[23] http://www.xlinux.com.cn/

[24] http://www.rtlinux.org/

[25] http:// www.realtime-info.be/

[26] http://www.qnx.com

[27] http://www.vxworks.com

[28] http://www.aero.polimi.it

[29] http://www.ittc.ukans.edu/kurt

[30] http://www.minigui.org/

[31] http://www.gnu.org/

[32] http://www.linuxdoc.org/

[33] http://www.linuxaid.com.cn

? ? A
? ? ? ? ? ? ? ? ? :
? ? ? ? ? ? (patch)? ? , ? ? ? Linux-2.2.12? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? Linux? ? ? ? :

```
# cd /usr/src/linux-2.2.12
# patch ????1 < /usr/src/linux-2.2.12/interrupt-latency-2.2.12-patch
```

? ? ? ? ? ? ? Linux? ? :

```
# cd /usr/src/linux-2.2.12
# make config # ? ? ? ?
# make dep
# make bzImage # ? ? ? ?
# make modules # ? ? ? ?
# make modules_install # ? ? ? ?
# cp arch/i386/boot/bzImage /boot/interrupt-lattency
```

? ? LILO, ? ? /etc/lilo.conf, ? ? ? ? ? ? ? ? ? :

```
image=/boot/interrupt-latency
label=interrupt-lattency
read-only
root=/dev/hda1
```

? ? ? LILO:

```
# /sbin/lilo
```

? ? ? ? LILO? ? ? ? ? ? :

```
LILO: interrupt-latency
```

? ? ? ? ? ? ? ? ? /? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ? ? ? ? ? ? ? (view.c, ? ? ?)? ? ? ? ? ? ? ? ? ? ? ? ? ? ?

interrupt-latency-2.2.12-patch? ? ? ? :

```
diff --exclude=version.h --exclude=config.h -Nru linux-2.2.12/arch/i386/kernel/Makefile linux-2.2.12-
interrupt/arch/i386/kernel/Makefile
--- linux-2.2.12/arch/i386/kernel/Makefile Wed Jan 20 10:18:53 1999
+++ linux-2.2.12-interrupt/arch/i386/kernel/Makefile Mon Mar 6 11:04:25 2000
@@ -14,7 +14,9 @@

O_TARGET := kernel.o
O_OBJS := process.o signal.o entry.o traps.o irq.o vm86.o \
- ptrace.o ioport.o ldt.o setup.o time.o sys_i386.o
+ ptrace.o ioport.o ldt.o setup.o time.o sys_i386.o \
+ intr_blocking.o
+
OX_OBJS := i386_ksyms.o
MX_OBJS :=

diff --exclude=version.h --exclude=config.h -Nru linux-2.2.12/arch/i386/kernel/entry.S linux-2.2.12-
interrupt/arch/i386/kernel/entry.S
--- linux-2.2.12/arch/i386/kernel/entry.S Fri Apr 30 08:13:37 1999
+++ linux-2.2.12-interrupt/arch/i386/kernel/entry.S Mon Mar 6 11:05:10 2000
@@ -96,6 +96,7 @@
movl %dx,%es;

#define RESTORE_ALL \
+ incl intrData; \
popl %ebx; \
popl %ecx; \
popl %edx; \
@@ -562,6 +563,8 @@
.long SYMBOL_NAME(sys_ni_syscall) /* streams1 */
.long SYMBOL_NAME(sys_ni_syscall) /* streams2 */
.long SYMBOL_NAME(sys_vfork) /* 190 */
+
+ .long SYMBOL_NAME(sys_get_intrData) /* 191 */

/*
* NOTE!! This doesn't have to be exact - we just have
diff --exclude=version.h --exclude=config.h -Nru linux-2.2.12/arch/i386/kernel/head.S linux-2.2.12-
interrupt/arch/i386/kernel/head.S
--- linux-2.2.12/arch/i386/kernel/head.S Thu Jan 14 22:57:25 1999
+++ linux-2.2.12-interrupt/arch/i386/kernel/head.S Mon Mar 6 11:14:26 2000
@@ -316,6 +316,7 @@
movl %ax,%es
pushl $int_msg
call SYMBOL_NAME(printk)
+ incl intrData
popl %eax
popl %ds
popl %es
diff --exclude=version.h --exclude=config.h -Nru linux-2.2.12/arch/i386/kernel/intr_blocking.c linux-2.2.12-
interrupt/arch/i386/kernel/intr_blocking.c
--- linux-2.2.12/arch/i386/kernel/intr_blocking.c Wed Dec 31 16:00:00 1969
+++ linux-2.2.12-interrupt/arch/i386/kernel/intr_blocking.c Mon Mar 6 11:41:50 2000
@@ -0,0 +1,322 @@
+#include <asm/system.h>
+
+
+/**** platform ****/
+#define readclock(low) \
+ __asm__ __volatile__ ("rdtsc" : "=a" (low) : : "edx")
+
+/**** configure ****/
```

```
+#define NUM_LOG_ENTRY 4
+#define INTR_IENABLE 0x200
+
+/**** data structure ****/
+struct IntrData {
+ /* count interrupt and iret */
+ int breakCount;
+
+ /* the test name */
+ const char * testName;
+
+ /* flag to control logging */
+ unsigned logFlag; /* 0 - no logging; 1 - logging */
+
+ /* panic flag - set to 1 if something is realy wrong */
+ unsigned panicFlag;
+
+ /* for synchro between start and end */
+ unsigned syncFlag;
+
+ /* we only log interrupts within certain range */
+ unsigned rangeLow;
+ unsigned rangeHigh;
+
+ /* count the total number interrupts and intrs in range*/
+ unsigned numIntrs;
+ unsigned numInRangeIntrs;
+
+
+ /* error accounting */
+ unsigned skipSti;
+ unsigned skipCli;
+ unsigned syncStiError;
+ unsigned syncCliError;
+ unsigned stiBreakError;
+ unsigned restoreSti;
+ unsigned restoreCli;
+
+ struct {
+ /* worst blocking time */
+ unsigned blockingTime;
+
+ const char * startFileName;
+ unsigned startFileLine;
+ unsigned startCount;
+
+ const char *endFileName;
+ unsigned endFileLine;
+ unsigned endCount;
+ } count[NUM_LOG_ENTRY];
+};
+
+struct IntrData intrData = {
+ 0,
+ "interrupt latency test (4 distinctive entries)",
+ 0,
+ 0,
+ 0,
+
+ 1,
+ 0xffffffff,
+
+ 0,
+ 0,
+
+ 0,
+ 0,
```

```
+ 0,
+ 0,
+ 0,
+ 0,
+ 0
+};
+
+
+/**** functions ****/
+#if 0
+void intr_check_int(int x)
+{
+
+ unsigned flag;
+ __intr_save_flags(flag);
+
+ if ((flag & INTR_IENABLE) != 0) {
+ switch(x) {
+ case 0 :
+ intrData.count[0].blockingTime ++;
+ break;
+ case 1 :
+ intrData.count[0].startFileLine ++;
+ break;

+ case 2 :
+ intrData.count[0].startCount ++;
+ break;
+ case 3 :
+ intrData.count[0].endFileLine ++;
+ break;
+ case 4 :
+ intrData.count[0].endCount ++;
+ break;
+ default :
+ intrData.count[0].startFileName = "Wrong check number";
+ break;
+ }
+ } else {
+ switch(x) {
+ case 0 :
+ intrData.count[1].blockingTime ++;
+ break;
+ case 1 :
+ intrData.count[1].startFileLine ++;
+ break;
+ case 2 :
+ intrData.count[1].startCount ++;
+ break;
+ case 3 :
+ intrData.count[1].endFileLine ++;
+ break;
+ case 4 :
+ intrData.count[1].endCount ++;
+ break;
+ default :
+ intrData.count[1].startFileName = "Wrong check number";
+ break;
+ }
+ }
+}
+
+#endif
+
+
+static inline void intr_SetPanic(unsigned x, const char *fname, unsigned l)
+{
```

```
+ if (intrData.panicFlag != 0) {
+ /* double error; impossible */
+ intrData.panicFlag = 99;
+ return;
+ }
+ intrData.panicFlag = x;
+ intrData.count[0].startFileName = fname;
+ intrData.count[0].startFileLine = l;
+}
+
+static const char *intrStartFileName;
+static unsigned intrStartFileLine;
+static unsigned intrStartCount;
+
+/* strategy :
+ * if it is true "cli", i.e., clearing the IF, we remember
+ * everything, and clear breakCount.
+ */
+void intr_cli(const char *fname, unsigned lineno)
+{
+ unsigned flag;
+ __intr_save_flags(flag);
+
+ __intr_cli();
+
+ /* if we are not logging or we have an error, do nothing */
+ if ((intrData.logFlag == 0) || ( intrData.panicFlag != 0)) {
+ return;
+ }
+
+ /* do nothing we had IF cleared before we call this function */
+ if ((flag & INTR_IENABLE) == 0) {
+ intrData.skipCli ++;
+ return;
+ }
+
+ /* debug */
+ if (intrData.syncFlag == 1) {
+ intrData.syncCliError ++;
+ }
+ intrData.syncFlag = 1;
+
+ intrData.breakCount = 0;
+
+ /* Read the Time Stamp Counter */
+ intrStartFileName = fname;
+ intrStartFileLine = lineno;
+ readclock(intrStartCount);
+}
+
+
+/* strategy:
+ * we do a count only if
+ * 1. syncFlag is 1 (a valid cli() was called)
+ * 2. breakCount is 0 (no iret is called between cli() and this sti()
+ */
+void intr_sti(const char *fname, unsigned lineno)
+{
+ unsigned flag;
+ unsigned endCount;
+ unsigned diff;
+ int i;
+
+ __intr_save_flags(flag);
+
+ /* if we are not logging or we have an error, do nothing */
+ if ((intrData.logFlag == 0) || ( intrData.panicFlag != 0)) {
```

```
+ __intr_sti();
+ return;
+ }
+
+
+ /* check if this is a real sti() */
+ if ((flag & INTR_IENABLE) != 0) {
+ intrData.skipSti ++;
+ __intr_sti();
+ return;
+ }
+
+ /* check 1*/
+ if (intrData.syncFlag != 1) {
+ intrData.syncStiError ++;
+ __intr_sti();
+ return;
+ }
+
+ /* check 2 */
+ if (intrData.breakCount != 0) {
+ intrData.stiBreakError ++;
+ __intr_sti();
+ return;
+ }
+
+ /* read count again */
+ readclock(endCount);
+
+ intrData.syncFlag = 0;
+
+ diff = endCount - intrStartCount;
+
+ if ((diff >= intrData.rangeLow) && (diff <= intrData.rangeHigh)) {
+ unsigned lowest=0xffffffff;
+ unsigned lowestIndex;
+ unsigned sameIndex = 0xffffffff;
+
+ intrData.numInRangeIntrs++;
+
+ /* check if we need to log this event */
+ for (i=0; i< NUM_LOG_ENTRY; i++) {
+
+ if (lowest > intrData.count[i].blockingTime) {
+ lowest = intrData.count[i].blockingTime;
+ lowestIndex = i;
+ }
+
+ if ( (lineno == intrData.count[i].endFileLine) &&
+ (intrStartFileLine == intrData.count[i].startFileLine) &&
+ (fname[0] == intrData.count[i].endFileName[0]) &&
+ (intrStartFileName[0] == intrData.count[i].startFileName[0]) ) {
+ /* if the line numbers are same, the first chars in
+ * both file names are same, we consider it is the same
+ * entry. */
+ sameIndex = i;
+ }
+ }
+
+ if (sameIndex == 0xffffffff) {
+ i = lowestIndex;
+ } else {
+ i = sameIndex;
+ }
+
+ if (diff > intrData.count[i].blockingTime) {
+ intrData.count[i].blockingTime = diff;
```

```
+ intrData.count[i].endFileName = fname;
+ intrData.count[i].endFileLine = lineno;
+ intrData.count[i].endCount = endCount;
+ intrData.count[i].startFileName = intrStartFileName;
+ intrData.count[i].startFileLine = intrStartFileLine;
+ intrData.count[i].startCount = intrStartCount;
+ }
+ }
+
+ intrData.numIntrs++;
+ __intr_sti();
+}
+
+
+void intr_restore_flags(const char *fname, unsigned lineno, unsigned x)
+{
+ unsigned flag;
+
+ /* if we are not logging or we have an error, do nothing */
+ if ((intrData.logFlag == 0) || ( intrData.panicFlag != 0)) {
+ __intr_restore_flags(x);
+ return;
+ }
+
+ __intr_save_flags(flag);
+
+ if (((flag & INTR_IENABLE) == 0) &&
+ ((x & INTR_IENABLE) != 0) ) {
+ intrData.restoreSti ++;
+ intr_sti(fname, lineno);
+ }
+
+ if ( ((flag & INTR_IENABLE) != 0) &&
+ ((x & INTR_IENABLE) == 0) ) {
+ intrData.restoreCli ++;
+ intr_cli(fname, lineno);
+ }
+
+ __intr_restore_flags(x);
+}
+
+#include <asm/uaccess.h>
+
+asmlinkage int sys_get_intrData(void ** ptr)
+{
+ return put_user(&intrData, ptr);
+}
diff --exclude=version.h --exclude=config.h -Nru linux-2.2.12/include/asm-i386/system.h linux-2.2.12-interrupt/include/asm-
i386/system.h
--- linux-2.2.12/include/asm-i386/system.h Mon Oct 11 21:28:12 1999
+++ linux-2.2.12-interrupt/include/asm-i386/system.h Mon Mar 6 11:08:02 2000
@@ -174,13 +174,33 @@
#define wmb() __asm__ __volatile__ ("": : :"memory")

/* interrupt control.. */
+#if 0
#define __sti() __asm__ __volatile__ ("sti": : :"memory")
#define __cli() __asm__ __volatile__ ("cli": : :"memory")
#define __save_flags(x) \
__asm__ __volatile__("pushfl ; popl %0":"=g" (x): /* no input */ :"memory")
#define __restore_flags(x) \
__asm__ __volatile__("pushl %0 ; popfl": /* no output */ :"g" (x):"memory")
+#endif

+#define __intr_sti() __asm__ __volatile__ ("sti": : :"memory")
+#define __intr_cli() __asm__ __volatile__ ("cli": : :"memory")
+#define __intr_save_flags(x) \
```

```
+__asm__ __volatile__("pushfl ; popl %0":"=g" (x): /* no input */ :"memory")
+#define __intr_restore_flags(x) \
+__asm__ __volatile__("pushl %0 ; popfl": /* no output */ :"g" (x):"memory")
+
+/* jsun */
+extern void intr_cli(const char *, unsigned);
+extern void intr_sti(const char *, unsigned);
+extern void intr_restore_flags(const char *, unsigned, unsigned);
+extern void intr_sync_flag(const char *, unsigned lineno);
+
+#define __cli() intr_cli(__FILE__, __LINE__)
+#define __sti() intr_sti(__FILE__, __LINE__)
+#define __save_flags(x) \
+__asm__ __volatile__("pushfl ; popl %0":"=g" (x): /* no input */ :"memory")
+#define __restore_flags(x) intr_restore_flags(__FILE__, __LINE__, x)

#ifdef __SMP__

@@ -197,9 +217,8 @@

#define cli() __cli()
#define sti() __sti()
-#define save_flags(x) __save_flags(x)
+#define save_flags(x) __save_flags(x)
#define restore_flags(x) __restore_flags(x)
-
#endif

/*

? ? ? ? ? ? view.c? ? :

#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <fcntl.h>
#include <assert.h>

/****************** CONFIG ****************/
#define NUM_LOG_ENTRY 4
#define CLOCK_FREQUENCY 266 /* cycles per microsecond */
#define SYSCALL_NUMBER 191

/****************** END OF CONFIG ***************/

#define CMD_EXIT 0
#define CMD_DISPLAY 1
#define CMD_START 2
#define CMD_STOP 3
#define CMD_CONTINUE 4
#define CMD_SET_RANGE 5
#define CMD_LAST 6

struct IntrData {
/* count interrupt and iret */
int breakCount;

/* the test name */
const char * testName;

/* flag to control logging */
unsigned logFlag; /* 0 - no logging; 1 - logging */

/* panic flag - set to 1 if something is realy wrong */
unsigned panicFlag;

/* for synchro between start and end */
```

```
unsigned syncFlag;

/* we only log interrupts within certain range */
unsigned rangeLow;
unsigned rangeHigh;

/* count the total number interrupts and intrs in range*/
unsigned numIntrs;
unsigned numInRangeIntrs;


/* error accounting */
unsigned skipSti;
unsigned skipCli;
unsigned syncStiError;
unsigned syncCliError;
unsigned stiBreakError;
unsigned restoreSti;
unsigned restoreCli;

struct {
/* worst blocking time */
unsigned blockingTime;

const char * startFileName;
unsigned startFileLine;
unsigned startCount;

const char *endFileName;
unsigned endFileLine;
unsigned endCount;
} count[NUM_LOG_ENTRY];
};



unsigned pData;
struct IntrData data;
int kmem;
char buf[81];

unsigned int GetInt(const char *prompt)
{
unsigned int i;
printf("%s", prompt);
scanf("%d", &i);
return i;
}

unsigned int GetHex(const char *prompt)
{
unsigned int i;
printf("%s", prompt);
scanf("%x", &i);
return i;
}

unsigned GetCommand()
{
for(;;) {
unsigned cmd;
printf("\n");
printf("Command Menu \n");
printf("=============\n");
printf("0-Exit 1-Display 2-Start logging 3-Stop logging 4-Continue\n");
printf("5-Set range\n");
printf("\n");
```

```
printf("Your choice : ");
scanf("%u", &cmd);
if ((cmd >= 0) && (cmd < CMD_LAST)) {
return cmd;
} else {
printf("Invalid choice!!!!\n\n");
}
}
}

unsigned GetKmemInt(unsigned offset)
{
off_t seekError;
ssize_t size;
unsigned num=0;

assert((offset & 3) == 0);

seekError = lseek(kmem, offset, SEEK_SET);
assert(seekError != (off_t)-1);

size = read(kmem, &num, sizeof(num));
assert(sizeof(num) == 4);
assert(size == 4);
return num;
}

char * GetKmemString(unsigned offset)
{
off_t seekError;
ssize_t size;

buf[80]=0;
buf[0]=0;

if (offset == 0) return buf;

seekError = lseek(kmem, offset, SEEK_SET);
assert(seekError != (off_t)-1);

size = read(kmem, buf, 80);
assert(size == 80);
return buf;
}

void GetKmemBlock(unsigned offset, void * buf, unsigned bufSize)
{
off_t seekError;
ssize_t size;

seekError = lseek(kmem, offset, SEEK_SET);
assert(seekError != (off_t)-1);

size = read(kmem, buf, bufSize);
assert(size == bufSize);
}

void SetKmemBlock(unsigned offset, void *buf, unsigned bufSize)
{
off_t seekError;
ssize_t size;

seekError = lseek(kmem, offset, SEEK_SET);
assert(seekError != (off_t)-1);

size = write(kmem, buf, bufSize);
assert(size == bufSize);
```

```
}

void SetKmemInt(unsigned offset, unsigned x)
{
off_t seekError;
ssize_t size;

seekError = lseek(kmem, offset, SEEK_SET);
assert(seekError != (off_t)-1);

size = write(kmem, &x, sizeof(x));
assert(size == sizeof(x));
}


void Display()
{
unsigned i;

GetKmemBlock(pData, &data, sizeof(data));
printf("%s : \n", GetKmemString((unsigned)data.testName));
printf("breakCount : %d\n", data.breakCount);
printf("logFlag : %d\n", data.logFlag);
printf("panicFlag : %d\n", data.panicFlag);
printf("syncFlag : %d\n", data.syncFlag);

printf("range : [%u(0x%x) : %u(0x%x)]\n",
data.rangeLow, data.rangeLow,
data.rangeHigh, data.rangeHigh);

printf("numIntrs : %u\n", data.numIntrs);
printf("numInRangeInts: %u\n", data.numInRangeIntrs);

printf("skipSti skipCli syncSti syncCli stiBreak restSti restCli\n");
printf("%d\t%d\t%d\t%d\t%d\t%d\t%d\t\n",
data.skipSti,
data.skipCli,
data.syncStiError,
data.syncCliError,
data.stiBreakError,
data.restoreSti,
data.restoreCli);


for (i=0; i< NUM_LOG_ENTRY; i++) {
printf("log entry : %d\n", i);
printf("\tblockingTime : %u (%u us)\n",
data.count[i].blockingTime,
data.count[i].blockingTime / CLOCK_FREQUENCY);
printf("\tstartFileName : %s\n",
GetKmemString((unsigned)data.count[i].startFileName));
printf("\tstartFileLine : %u\n", data.count[i].startFileLine);
printf("\tstartCount : %u\n", data.count[i].startCount);
printf("\tendFileName : %s\n",
GetKmemString((unsigned)data.count[i].endFileName));
printf("\tendFileLine : %u\n", data.count[i].endFileLine);
printf("\tendCount : %u\n", data.count[i].endCount);
}
printf("\n");
}

void StartLogging()
{
unsigned i;

/* init first */
data.breakCount = 0;
```

```
data.logFlag = 0;
data.panicFlag = 0;
data.syncFlag = 0;

data.numIntrs = 0;
data.numInRangeIntrs = 0;

data.skipSti =
data.skipCli =
data.syncStiError =
data.syncCliError =
data.stiBreakError =
data.restoreSti =
data.restoreCli = 0;

for (i=0; i< NUM_LOG_ENTRY; i++) {
data.count[i].blockingTime = 0;
data.count[i].startFileName = 0;
data.count[i].startFileLine = 0;
data.count[i].startCount = 0;
data.count[i].endFileName = 0;
data.count[i].endFileLine = 0;
data.count[i].endCount = 0;
}

SetKmemBlock(pData, &data, sizeof(data));

/* turn the logging flag */
SetKmemInt(pData + ((unsigned)&data.logFlag - (unsigned)&data), 1);
}

void EndLogging()
{
/* turn the logging flag */
SetKmemInt(pData + ((unsigned)&data.logFlag - (unsigned)&data), 0);
}

void ContinueLoggin()
{
/* turn the logging flag */
SetKmemInt(pData + ((unsigned)&data.logFlag - (unsigned)&data), 1);
}

void SetRange()
{
data.rangeLow = GetInt("Input lower bound (decimal) : ");
printf("\tlower bound is %u(0x%x)\n", data.rangeLow, data.rangeLow);

data.rangeHigh = GetInt("Input upper bound (decimal) : ");
printf("\tupper bound is %u(0x%x)\n", data.rangeHigh, data.rangeHigh);

SetKmemInt(pData + ((unsigned)&data.rangeLow - (unsigned)&data),
data.rangeLow);
SetKmemInt(pData + ((unsigned)&data.rangeHigh - (unsigned)&data),
data.rangeHigh);
}


main()
{
unsigned int cmd;
unsigned long offset;

kmem = open("/dev/kmem", O_RDWR);
assert(kmem > 0);

if (syscall(SYSCALL_NUMBER, &pData) != 0) {
```

```
printf("failed to get jsunData address through syscall 191!\n");
pData = GetHex("Input mannually the address of jsunData : ");
}

GetKmemBlock(pData, &data, sizeof(data));

for(;;) {
cmd = GetCommand();

switch (cmd) {
case CMD_DISPLAY:
Display();
break;

case CMD_START:
StartLogging();
break;

case CMD_STOP:
EndLogging();
break;

case CMD_CONTINUE:
ContinueLoggin();
break;

case CMD_SET_RANGE:
SetRange();
break;

case CMD_EXIT:
close(kmem);
return;

default:
assert(0 == 1);
}
}
}

? ? B
? ? ? ? ? ? ? ? ? lat_ctx.c:

/*
* lat_ctx.c - context switch timer
*
* usage: lat_ctx [-s size] #procs [#procs....]
*
* Copyright (c) 1994 Larry McVoy. Distributed under the FSF GPL with
* additional restriction that results may published only if
* (1) the benchmark is unmodified, and
* (2) the version in the sccsid below is included in the report.
* Support for this development by Sun Microsystems is gratefully acknowledged.
*/
char *id = "$Id$\n";

#include "bench.h"

#if defined(sgi) && defined(PIN)
#include <sys/sysmp.h>
#include <sys/syssgi.h>
int ncpus;
#endif

#define MAXPROC 2048
#define CHUNK (4<<10)
#define TRIPS 5
```

```
#ifndef max
#define max(a, b) ((a) > (b) ? (a) : (b))
#endif

int process_size, *data; /* size & pointer to an array that big */
int pids[MAXPROC];
int p[MAXPROC][2];
double pipe_cost(int p[][2], int procs);
int ctx(int procs, int nprocs);
int sumit(int);
void killem(int procs);
void doit(int p[MAXPROC][2], int rd, int wr);
int create_pipes(int p[][2], int procs);
int create_daemons(int p[][2], int pids[], int procs);

int
main(int ac, char **av)
{
int i, max_procs;
double overhead = 0;

if (ac < 2) {
usage: printf("Usage: %s [-s kbytes] processes [processes ...]\n",
av[0]);
exit(1);
}

/*
* Need 4 byte ints.
*/
if (sizeof(int) != 4) {
fprintf(stderr, "Fix sumit() in ctx.c.\n");
exit(1);
}

/*
* If they specified a context size, get it.
*/
if (!strcmp(av[1], "-s")) {
if (ac < 4) {
goto usage;
}
process_size = atoi(av[2]) * 1024;
if (process_size > 0) {
data = (int *)calloc(1, max(process_size, CHUNK));
BENCHO(sumit(CHUNK), sumit(0), 0);
overhead = gettime();
overhead /= get_n();
overhead *= process_size;
overhead /= CHUNK;
}
ac -= 2;
av += 2;
}

#if defined(sgi) && defined(PIN)
ncpus = sysmp(MP_NPROCS);
sysmp(MP_MUSTRUN, 0);
#endif
for (max_procs = atoi(av[1]), i = 1; i < ac; ++i) {
int procs = atoi(av[i]);
if (max_procs < procs) max_procs = procs;
}
max_procs = create_pipes(p, max_procs);
overhead += pipe_cost(p, max_procs);
max_procs = create_daemons(p, pids, max_procs);
fprintf(stderr, "\n\"size=%dk ovr=%.2f\n", process_size/1024, overhead);
```

```
for (i = 1; i < ac; ++i) {
double time;
int procs = atoi(av[i]);

if (procs > max_procs) continue;

BENCH(ctx(procs, max_procs), 0);
time = usecs_spent();
time /= get_n();
time /= procs;
time /= TRIPS;
time -= overhead;
fprintf(stderr, "%d %.2f\n", procs, time);
}

/*
* Close the pipes and kill the children.
*/
killem(max_procs);
for (i = 0; i < max_procs; ++i) {
close(p[i][0]);
close(p[i][1]);
if (i > 0) {
wait(0);
}
}
return (0);
}


int
ctx(int procs, int nprocs)
{
int msg;
int i;
int sum;

/*
* Main process - all others should be ready to roll, time the
* loop.
*/
for (i = 0; i < TRIPS; ++i) {
if (write(p[nprocs - procs][1], &msg, sizeof(msg)) !=
sizeof(msg)) {
perror("read/write on pipe");
exit(1);
}
if (read(p[nprocs-1][0], &msg, sizeof(msg)) != sizeof(msg)) {
perror("read/write on pipe");
exit(1);
}
sum = sumit(process_size);
}
return (sum);
}

void
killem(int procs)
{
int i;

for (i = 1; i < procs; ++i) {
if (pids[i] > 0) {
kill(pids[i], SIGTERM);
}
}
}
```

```
void
doit(int p[][2], int rd, int wr)
{
int msg, sum = 0 /* lint */;

signal(SIGTERM, SIG_DFL);
if (data) bzero((void*)data, process_size);
for ( ;; ) {
if (read(p[rd][0], &msg, sizeof(msg)) != sizeof(msg)) {
perror("read/write on pipe");
break;
}
sum = sumit(process_size);
if (write(p[wr][1], &msg, sizeof(msg)) != sizeof(msg)) {
perror("read/write on pipe");
break;
}
}
use_int(sum);
exit(1);
}

int
doit_cost(int p[][2], int procs)
{
static int k;
int msg = 1;
int i;

for (i = 0; i < TRIPS; ++i) {
if (write(p[k][1], &msg, sizeof(msg)) != sizeof(msg)) {
perror("read/write on pipe");
exit(1);
}
if (read(p[k][0], &msg, sizeof(msg)) != sizeof(msg)) {
perror("read/write on pipe");
exit(1);
}
if (++k == procs) {
k = 0;
}
}
return (msg);
}

/*
* The cost returned is the cost of going through one pipe once in usecs.
* No memory costs are included here, this is different than lmbench1.
*/
double
pipe_cost(int p[][2], int procs)
{
double result;

/*
* Measure the overhead of passing a byte around the ring.
*/
BENCH(doit_cost(p, procs), 0);
result = usecs_spent();
result /= get_n();
result /= TRIPS;
return result;
}

int
create_daemons(int p[][2], int pids[], int procs)
```

```
{
int i;
int msg;

/*
* Use the pipes as a ring, and fork off a bunch of processes
* to pass the byte through their part of the ring.
*
* Do the sum in each process and get that time before moving on.
*/
signal(SIGTERM, SIG_IGN);
for (i = 1; i < procs; ++i) {
switch (pids[i] = fork()) {
case -1: /* could not fork, out of processes? */
procs = i;
break;

case 0: /* child */
#if defined(sgi) && defined(PIN)
sysmp(MP_MUSTRUN, i % ncpus);
#endif
doit(p, i-1, i);
/* NOTREACHED */

default: /* parent */
;
}
}

/*
* Go once around the loop to make sure that everyone is ready and
* to get the token in the pipeline.
*/
if (write(p[0][1], &msg, sizeof(msg)) != sizeof(msg) ||
read(p[procs-1][0], &msg, sizeof(msg)) != sizeof(msg)) {
perror("write/read/write on pipe");
exit(1);
}
if (data) bzero((void*)data, process_size);
return procs;
}

int
create_pipes(int p[][2], int procs)
{
int i;
/*
* Get a bunch of pipes.
*/
morefds();
for (i = 0; i < procs; ++i) {
if (pipe(p[i]) == -1) {
return i;
}
}
return procs;
}

/*
* Bring howmuch data into the cache, assuming that the smallest cache
* line is 16 bytes.
*/
int
sumit(int howmuch)
{
int done, sum = 0;
register int *d = data;
```

```
#if 0
#define A sum+=d[0]+d[4]+d[8]+d[12]+d[16]+d[20]+d[24]+d[28]+\
d[32]+d[36]+d[40]+d[44]+d[48]+d[52]+d[56]+d[60]+\
d[64]+d[68]+d[72]+d[76]+d[80]+d[84]+d[88]+d[92]+\
d[96]+d[100]+d[104]+d[108]+d[112]+d[116]+d[120]+d[124];\
d+=128;
#define TWOKB A A A A
#else
#define A sum+=d[0]+d[1]+d[2]+d[3]+d[4]+d[5]+d[6]+d[7]+d[8]+d[9]+\
d[10]+d[11]+d[12]+d[13]+d[14]+d[15]+d[16]+d[17]+d[18]+d[19]+\
d[20]+d[21]+d[22]+d[23]+d[24]+d[25]+d[26]+d[27]+d[28]+d[29]+\
d[30]+d[31]+d[32]+d[33]+d[34]+d[35]+d[36]+d[37]+d[38]+d[39]+\
d[40]+d[41]+d[42]+d[43]+d[44]+d[45]+d[46]+d[47]+d[48]+d[49]+\
d[50]+d[51]+d[52]+d[53]+d[54]+d[55]+d[56]+d[57]+d[58]+d[59]+\
d[60]+d[61]+d[62]+d[63]+d[64]+d[65]+d[66]+d[67]+d[68]+d[69]+\
d[70]+d[71]+d[72]+d[73]+d[74]+d[75]+d[76]+d[77]+d[78]+d[79]+\
d[80]+d[81]+d[82]+d[83]+d[84]+d[85]+d[86]+d[87]+d[88]+d[89]+\
d[90]+d[91]+d[92]+d[93]+d[94]+d[95]+d[96]+d[97]+d[98]+d[99]+\

d[100]+d[101]+d[102]+d[103]+d[104]+\
d[105]+d[106]+d[107]+d[108]+d[109]+\
d[110]+d[111]+d[112]+d[113]+d[114]+\
d[115]+d[116]+d[117]+d[118]+d[119]+\
d[120]+d[121]+d[122]+d[123]+d[124]+d[125]+d[126]+d[127];\
d+=128; /* ints; bytes == 512 */
#define TWOKB A A A A
#endif

for (done = 0; done < howmuch; done += 2048) {
TWOKB
}
return (sum);
}


/* bench.h */

#ifndef _BENCH_H
#define _BENCH_H

#ifdef WIN32
#include <windows.h>
typedef unsigned char bool_t;
#endif

#include <assert.h>
#include <ctype.h>
#include <stdio.h>
#ifndef WIN32
#include <unistd.h>
#endif
#include <stdlib.h>
#include <fcntl.h>
#include <signal.h>
#include <errno.h>
#ifndef WIN32
#include <strings.h>
#endif
#include <sys/types.h>
#ifndef WIN32
#include <sys/mman.h>
#endif
#include <sys/stat.h>
#ifndef WIN32
#include <sys/wait.h>
#include <time.h>
```

```
#include <sys/time.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <sys/resource.h>
#define PORTMAP
#include <rpc/rpc.h>
#endif

#ifndef HAVE_uint
typedef unsigned int uint;
#endif

#ifdef HAVE_uint64_t
typedef uint64_t uint64;
#else
typedef unsigned long long uint64;
#endif

#define NO_PORTMAPPER /* needs to be up here, lib_*.h look at it */
#include "stats.h"
#include "timing.h"
#include "lib_tcp.h"
#include "lib_udp.h"
#include "lib_unix.h"


#ifdef DEBUG
# define debug(x) fprintf x
#else
# define debug(x)
#endif
#ifdef NO_PORTMAPPER
#define TCP_SELECT -31233
#define TCP_XACT -31234
#define TCP_CONTROL -31235
#define TCP_DATA -31236
#define TCP_CONNECT -31237
#define UDP_XACT -31238
#define UDP_DATA -31239
#else
#define TCP_SELECT (u_long)404038 /* XXX - unregistered */
#define TCP_XACT (u_long)404039 /* XXX - unregistered */
#define TCP_CONTROL (u_long)404040 /* XXX - unregistered */
#define TCP_DATA (u_long)404041 /* XXX - unregistered */
#define TCP_CONNECT (u_long)404042 /* XXX - unregistered */
#define UDP_XACT (u_long)404032 /* XXX - unregistered */
#define UDP_DATA (u_long)404033 /* XXX - unregistered */
#define VERS (u_long)1
#endif

#define UNIX_CONTROL "/tmp/lmbench.ctl"
#define UNIX_DATA "/tmp/lmbench.data"
#define UNIX_LAT "/tmp/lmbench.lat"

/*
* socket send/recv buffer optimizations
*/
#define SOCKOPT_READ 0x0001
#define SOCKOPT_WRITE 0x0002
#define SOCKOPT_RDWR 0x0003
#define SOCKOPT_PID 0x0004
#define SOCKOPT_REUSE 0x0008
#define SOCKOPT_NONE 0

#ifndef SOCKBUF
#define SOCKBUF (1024*1024)
#endif
```

```
#ifndef XFERSIZE
#define XFERSIZE (64*1024) /* all bandwidth I/O should use this */
#endif

#if defined(SYS5) || defined(WIN32)
#define bzero(b, len) memset(b, 0, len)
#define bcopy(s, d, l) memcpy(d, s, l)
#define rindex(s, c) strrchr(s, c)
#endif
#define gettime usecs_spent
#define streq !strcmp
#define ulong unsigned long

#ifdef USE_RAND
#define srand48 srand
#define drand48() ((double)rand() / (double)RAND_MAX)
#endif
#ifdef USE_RANDOM
#define srand48 srand
#define drand48() ((double)rand() / (double)RAND_MAX)
#endif

#ifdef WIN32
#include <process.h>
#define getpid _getpid
int gettimeofday(struct timeval *tv, struct timezone *tz);
#endif

#define SMALLEST_LINE 32 /* smallest cache line size */
#define TIME_OPEN2CLOSE

#define GO_AWAY signal(SIGALRM, exit); alarm(60 * 60);
#define REAL_SHORT 50000
#define SHORT 1000000
#define MEDIUM 2000000
#define LONGER 7500000 /* for networking data transfers */
#define ENOUGH REAL_SHORT

#define TRIES 11

typedef struct {
int N;
uint64 u[TRIES];
uint64 n[TRIES];
} result_t;
void insertinit(result_t *r);
void insertsort(uint64, uint64, result_t *);
void save_median();
void save_minimum();
void save_results(result_t *r);
void get_results(result_t *r);


#define BENCHO(loop_body, overhead_body, enough) { \
int __i, __N; \
double __oh; \
result_t __overhead, __r; \
insertinit(&__overhead); insertinit(&__r); \
__N = (enough == 0 || get_enough(enough) <= 100000) ? TRIES : 1;\
if (enough < LONGER) {loop_body;} /* warm the cache */ \
for (__i = 0; __i < __N; ++__i) { \
BENCH1(overhead_body, enough); \
if (gettime() > 0) \
insertsort(gettime(), get_n(), &__overhead); \
BENCH1(loop_body, enough); \
if (gettime() > 0) \
```

```
insertsort(gettime(), get_n(), &__r); \
} \
for (__i = 0; __i < __r.N; ++__i) { \
__oh = __overhead.u[__i] / (double)__overhead.n[__i]; \
__r.u[__i] -= (uint64)((double)__r.n[__i] * __oh); \
} \
save_results(&__r); \
}

#define BENCH(loop_body, enough) { \
long __i, __N; \
result_t __r; \
insertinit(&__r); \
__N = (enough == 0 || get_enough(enough) <= 100000) ? TRIES : 1;\
if (enough < LONGER) {loop_body;} /* warm the cache */ \
for (__i = 0; __i < __N; ++__i) { \
BENCH1(loop_body, enough); \
if (gettime() > 0) \
insertsort(gettime(), get_n(), &__r); \
} \
save_results(&__r); \
}

#define BENCH1(loop_body, enough) { \
double __usecs; \
BENCH_INNER(loop_body, enough); \
__usecs = gettime(); \
__usecs -= t_overhead() + get_n() * l_overhead(); \
settime(__usecs >= 0. ? (uint64)__usecs : 0.); \
}

#define BENCH_INNER(loop_body, enough) { \
static u_long __iterations = 1; \
int __enough = get_enough(enough); \
u_long __n; \
double __result = 0.; \
\
while(__result < 0.95 * __enough) { \
start(0); \
for (__n = __iterations; __n > 0; __n--) { \
loop_body; \
} \
__result = stop(0,0); \
if (__result < 0.99 * __enough \
|| __result > 1.2 * __enough) { \
if (__result > 150.) { \
double tmp = __iterations / __result; \
tmp *= 1.1 * __enough; \
__iterations = (u_long)(tmp + 1); \
} else { \
if (__iterations > (u_long)1<<27) { \
__result = 0.; \
break; \
} \
__iterations <<= 3; \
} \
} \
} /* while */ \
save_n((uint64)__iterations); settime((uint64)__result); \
}


/*
* Generated from msg.x which is included here:

program XACT_PROG {
version XACT_VERS {
```

```
char
RPC_XACT(char) = 1;
} = 1;
} = 3970;

* Please do not edit this file.
* It was generated using rpcgen.
*/

#include <rpc/types.h>

#define XACT_PROG ((u_long)404040)
#define XACT_VERS ((u_long)1)
#define RPC_XACT ((u_long)1)
#define RPC_EXIT ((u_long)2)
extern char *rpc_xact_1();
extern char *client_rpc_xact_1();

#endif /* _BENCH_H */
```

? ? : ? ? ? , ? ? ? ? ? , ? ? ? ? ? ?

mrunix 05-06-23, 11:27

? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ?
http://blog.csdn.net/mrunix
? ? ? ? , ? ? ? PDF? ? ? ? ? ? ? ? ? ? ? ?
http://www.teemall.com.cn/other/ELinuxDevApp.pdf

jet_ok 05-06-23, 11:42

? ? ? ? ? ? ? ? ? ? , ? ? ? ? ? ?
http://blog.csdn.net/mrunix
? ? ? ? , ? ? ? PDF? ? ? ? ? ? ? ? ? ? ? ?
http://www.teemall.com.cn/other/ELinuxDevApp.pdf

? ? mrunix, ? ? ? ? ? ? ? ? ? ? ? ?

? ? 06-01-17, 22:21

? ! ! ! !