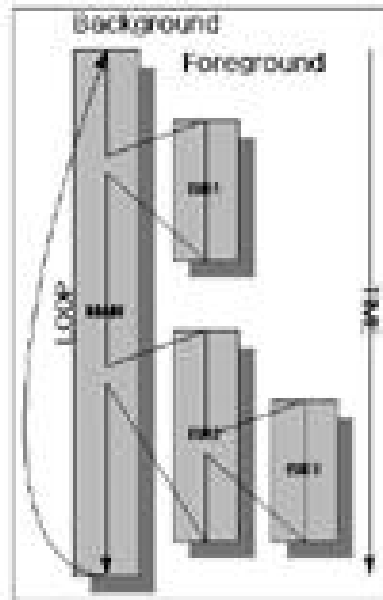


Embedded System (uC/OS-II)

RTOS

Digital System Design Lab.
(promise@secsm.org) (psh@cscsm.org)

가
가
가 ..
가
가
“Foreground/Background”
“Super Loops”



? 가
가
LED TASK CPU 가
(TASK 2 !
).
가
4 가 6 TASK CPU가
TASK 가 가?
가 TASK 가 가
가 , TASK가 2 가
2 가 , LED, 가



(,)

(가) (가

가 가 ..) 가 가

가

() RTOS

2 3 가

가 Interrupt 가

가 TASK 가

가

RTOS

Flow DSP ASIC

RTOS uC/OS-II (Hard-Real-Time) 가

가 CPU RTOS

RTOS

가

(RTOS : Real-Time Operating System)

(Real-Time OS) 가

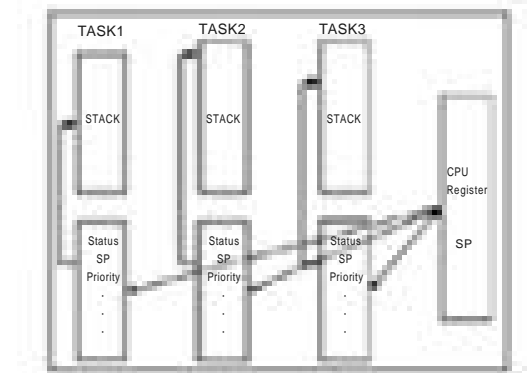
가

Mutual Exclusion

· Multitasking : ()
 가
 가
 가
 (1 CPU) CPU

· Context Switching : TASK가 가
 TASK
 TASK
 TASK CPU
 Context Switching
 Over-head 가

TASK
 TASK
 가
 가
 가
 CPU
 TASK management(TASK scheduling, Communication, Synchronization) Memory
 Memory management, Data I/O management, Time Management, Interrupt service, File system management

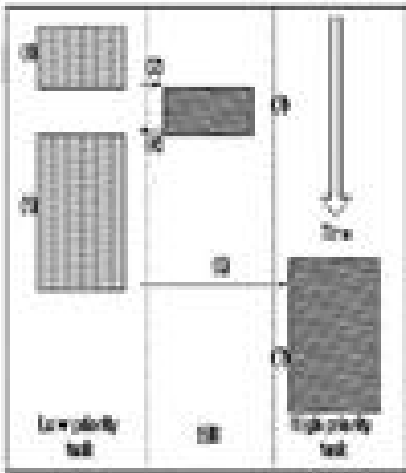
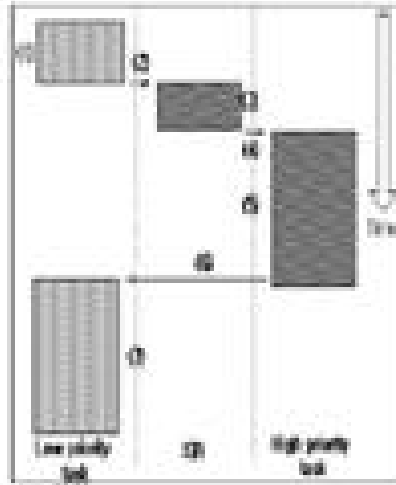


· Resource : TASK가 Entry(I/O,)
 · Shared Resource : TASK ()

· Kernel : Multitasking
 Context switching, TASK scheduling, Memory management (2 5%)
 가 ROM



RAM
 Scheduler : Multitasking
 TASK
 가 Scheduling
 scheduling
 Priority-based
 Priority
 FIFO RR(Round-Robin)



mutual exclusion

TASK

· TASK Communication : TASK

Global variable()

Message passing 2가 . Global variable
Exclusive access

ISR(Interrupt Service Routine) Interrupt
Interrupt Disable TASK Interrupt

Disable Semaphore

Message passing Mailbox, Queue,
Pipe

· Message Mailboxes : Message

TASK . Message Exchange
, TASK Message

Message

· Message Queues : TASK 1

가

TASK 가 FIFO(Queue)

* : TASK가
TASK TASK

* : TASK TASK
Interrupt latency가
Priority

TASK가 TASK가

· Critical Section : TASK

· Deadlock :

가

가

mutual exclusion

Semaphore

Semaphore

· Semaphore : Mu-

· Priority Inversion : Priority가

TASK Priority

TASK 가 Semaphore TASK Priority 가 pSOS VxWorks가 Embedded Linux 가 pSOS 가

· TASK Synchronization : TASK Semaphore Event flag, Signal

· Interrupt Service : Asynchronous event CPU Interrupt latency가 가

ISR(Interrupt Service Routine) TASK 가 Priority TASK 가

· Clock Tick : TASK La-

· Reentrancy : 가 tency 가

TASK 가 POSIX API TASK 가 , C , Object-Oriented Language 가 , License

Semaphore uC/OS(Micro-C Operating System) 가 Target uC/OS-II가

RTOS 1. uC/OS- uC/OS-II Portable, ROMable 가 (C , , 가) , Preemptive Kernel(, 255 Nest Interrupt가) , Real-Time, Multitasking(가 64 TASK 가) 가 가 ANSI C 가 가

POSIX(Portable Operating System Interface) , Mailbox, Queue, Semaphore, Memory partition , 가 가 가



. VxWorks pSOS, Windows CE

Porting

. LINUX

가
가
가
가

RTOS

2. uC/OS-II

uC/OS-II Critical Section,
Tasks, Task Scheduling, Interrupt Service
Routines(ISRs) uC/OS-II Initializing

(1) Critical Sections

uC/OS-II Critical
Section Disable ,
Enable
Disable time Real-time
events
uC/OS-II Disable time
uC/OS-II Disable Enable
ON_ENTER_CRITICAL() ON_EXIT_CRITICAL
() . CPU
Disable Enable

(2) Tasks

TASK
delete
Listing 1 TASK가 delete
, OSTaskDel() TASK de-
lete . TASK
void

. uC/OS-II 64 TASK
가 0, 1, 2, 3
OS_LOWEST_PRIO-3, OS_LOWEST_PRIO-2,
OS_LOWEST_PRIO-1, OS_LOWEST_PRIO
56 TASK가 가

OSTaskChangePrio(), OSTaskDel() TASK

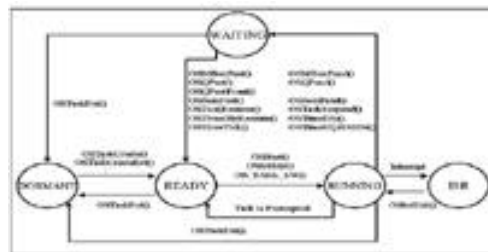
```
void MyTask (void *pArg)
{
    for(;;)
    {
        /* Your code here */
    }
}

void MyTask (void *pArg)
{
    /* Your code here */
    OSTaskDel(OSTaskID);
}
```

Listing 1. TASK

(3) TASK States

1. TASK
DORMANT TASK가 ROM RAM
TASK가 Delete
TASK OSTaskCreate() OSTaskCreateExt
() 1.
DORMANT TASK가 READY
가 . TASK TASK



1. TASK



```

*.OSTCBOpt option
OSTaskCreateExt(
OS_TASK_CREATE_EXT_EN "1"
OS_TASK_OPT_STK_CHK Task
, OS_TASK_OPT_STK_CLR TASK가
, Clear
OS_TASK_OPT_SAVE_FP TASK가
가 가
*.OSTCBId TASK
*.OSTCBNext .OSTCPrev OS_TCB
OS_TCB 가
*.OSTCBEventPtr Event Control Block
가
*.OSTCBMsg TASK
IPC
*.OSTCBDly TASK가 Clock
Delay Event가
Time Tick "1" "0"
TASK Delay
*.OSTCBStat TASK 가 "0"
TASK가
uC/OS-II.H
*.OSTCBPrio TASK
가 .( TASK
56 .)
*.OSTCBX, .OSTCBy, .OSTCBBitX,
OSTCBBitY Listing 3
가 TASK

```

```

OSTCBOPT = OS_OPT_STK_CHK | OS_OPT_STK_CLR | OS_OPT_SAVE_FP;
OSTCBPRT = OS_OPT_STK_CHK | OS_OPT_STK_CLR | OS_OPT_SAVE_FP;
OSTCBOPT = OS_OPT_STK_CHK | OS_OPT_STK_CLR | OS_OPT_SAVE_FP;
OSTCBBITX = OS_OPT_STK_CHK | OS_OPT_STK_CLR | OS_OPT_SAVE_FP;

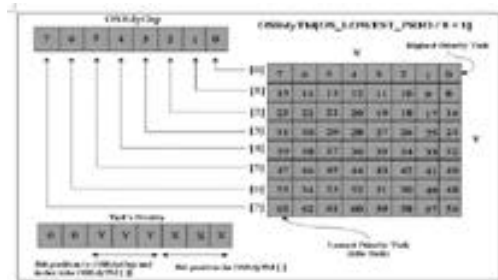
```

Listing 3. Calculating OS_TCB members

```

*.OSTCBDelReq Boolean 가
TASK가
(5) Ready List
TASK "0" OS_LOWEST_PRIO
가 . uC/OS-II가 가
OS_LOWEST_PRIO idle
TASK . MAX_TASKS
TASK
OS_LOWEST_PRIO . TASK
2. OSRdyGrp OSRdyTbl[ ]
8 TASK 가 , OSRdyTbl[ ]
64 TASK

```



2. The uC/OS-II ready list

```

OSRdyTbl[0] 가 "1"
OSRdyGrp 0 "1" 가
OSRdyTbl[3] 가 "1"
OSRdyGrp 3 "1" TASK
Ready List Listing 4
prio TASK
TASK Ready List Listing 5
Listing 6 Ready List 가

```

```

OSRdyGrp = OSRdyTbl[0];
OSRdyTbl[0] = OSRdyTbl[0];
OSRdyTbl[0] = OSRdyTbl[0];
OSRdyTbl[0] = OSRdyTbl[0];

```

Listing 4. Making a task ready to run


```

int OSMapTbl [prio > 0]
{
    OSMapTbl [prio & 15] = 0;
    OSMapTbl [prio & 31] = 0;
}
    
```

Listing 5. Removing a task from the ready list

```

p = OSMapTbl [OSMapTbl];
m = OSMapTbl [OSMapTbl | 0x1];
prio = (p << 8) + m;
    
```

Listing 6. Finding the highest Priority task ready to run

```

void OS_Sched(void)
{
    OS_SchedLock();
    OS_Sched();
    OS_SchedUnlock();
}

void OS_Sched(void)
{
    OS_SchedLock();
    OS_Sched();
    OS_SchedUnlock();
}

void OS_SchedLock(void)
{
    OS_SchedLock();
}

void OS_SchedUnlock(void)
{
    OS_SchedUnlock();
}
    
```

Listing 7. Task Scheduler

TASK

(6) Task Scheduling

가.
 가.
 가.
 TASK
 OSSched() TASK
 . OSSched() ISR (OSIntNesting
 > 0), OSSchedLock()
 Disable,
 . Listing 7 (2) READY TASK 가
 TASK (3) 가
 TASK가, TASK가
 TASK 가
 TASK가 TASK Context
 Switching . Context Switching
 TASK
 TASK
 READY TASK
 가 TASK

(7) Locking and Unlocking the Scheduler

uC/OS-II OSSchedLock()
 OSSchedUnlock() 가 TASK가
 OSSchedLock() , TASK
 CPU 가 TASK가 TASK

OSSchedLock() 255 가
 , OSSchedLock()
 OSSchedUnlock()
 . OSLockNesting "0", OSSchedUnlock
 () OSSchedlock()
 TASK가
 (8) Idle TASK
 ?
 가
 uC/OS-II TASK READY
 Idle TASK
 . Idle TASK
 TASK READY
 . OSTaskIdle() 32bit
 (OSIdleCtr) 가 가
 가.
 (9) Statistics TASK
 uC/OS-II TASK
 가 . OS_TASK_STAT_EX "1", OSTaskStat
 () 가 , (OS_CFG.H)
 OSTaskStat() 1 , CPU
 가
 OSStatInit()
 TASK



```

void main(void)
{
    OSInit() /* Initialize uC/OS-II */
    /* Create first startup task */
    /* Create at least one task using either
    OSTaskCreate() or OSTaskCreateExt() */
    OSStart() /* Start multitasking */
}

void OSTask(void)
{
    /* Main task execution - uC/OS-II takes over */
    OSIdle() /* Idle task (no application tasks) */
    /* ... */
    /* Code for Task() goes here. */
}

```

Listing 8. Initializing the statistic task

OSStart() Mutitasking
 OSStart() 가 Startup code가
 Listing 8 Statistic TASK

Main() OSStart() uC/OS-II
 TASK TaskStart
 () OSTaskIdle() OSTaskStat()
 가
 (OS_LOWEST_PRIO, OS_LOWEST_PRIO-1)

(10) Clock Tick
 uC/OS-II TASK Time Delay
 Tick 1/10
 1/200 Tick Rate
 Overhead CPU

. Mutitasking Ticker 가
 Enable (OSStart()),
 OSTimeTick() 가 uC/OS-II Clock
 tick . OSTimeTick() 가
 OSTimeTickHook() .
 OSStatInit()
 TASK
 (11) uC/OS-II Initialization Starting

```

OSInit() uC/OS-II

OSInit() OSInit()
OSTaskIdle() OS_CFG.H
OS_TASK_STAT_EN "1"
OS_LOWEST_PRIO "63"
OS_MAX_TASKS "2"
uC/OS-II TASK
OSStart()
Mutitasking
Listing 9 uC/OS-II Initializing Starting

```

```

void main(void)
{
    OSInit() /* Initialize uC/OS-II */

    /* Create at least 1 task using either
    OSTaskCreate() or OSTaskCreateExt() */

    OSStart() /* Start multitasking
    (OSStart() will not return) */
}

```

```

}

void OSTask(void)
{
    OSIdle()
    OSIdle()

    /* OSIdle() -- PALM */
    /* OSIdle() -- PALM */
    /* OSIdle() -- PALM */

    OSTimeTick() = OSTimeTick() - 1;
    OSTimeTick() = OSTimeTick()
    OSTimeTick()
    OSTimeTick() = OSTimeTick()
    OSTimeTick() = OSTimeTick()
    OSTimeTick() = OSTimeTick()
}


```

Listing 9. Initializing the statistic task

RTOS
 가
 uC/OS-II
 OS RTOS
 uC/OS-II
 Management
 TASK

E E

계재된 기사는 본지의 웹사이트를 통해서도 보실 수 있습니다
<http://www.chomdan.co.kr>



● ●

uC/OS-ii v2.0 uC/OS Jean J.Labrosse(Jeam.
 Labrosse@Micrium.com) os 가 v2.03 50\$
 v2.00

www.ucos-ii.com uC/OS-ii uC/OS, uC/OS-ii가 가
 Errata, Application Note, FAQ TCP/
 IP Protocol Stack

uC/OS-ii
 MicroC/OS-II The Real-Time Kernel, R&D Books, 560pp
 Embedded Systems Building Blocks 2nd Edition, R&D Books, 611pp

()

가 가 1920 E.
 가 . 1927 가
 G.E. 가 가 가 가
 G.가 가 가 가
 가 가
 . 1965 A.A. R.W.
 가 가