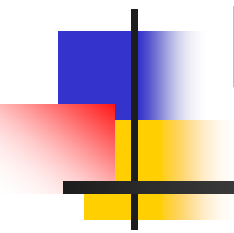
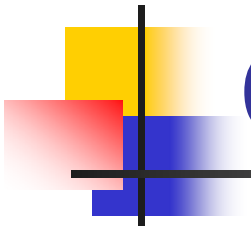


MicroC/OS-II Chapter 1

中興資科所 盧慶達

學號:79256022

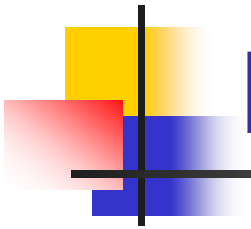
指導教授 張軒彬



Chapter 1

Example#1

1. Ten task display a number between 0 and 9 at random locations on the screen.
2. μC/OSⅡ is a multitasking kernel and allow you to have up to 63 application tasks.



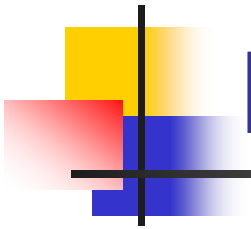
Example #1

(1)PC_DispClrScr(DISP_FGND_WHITE+DISP_BGND_BLACK);

1.the function from PC.C to privode services in a DOS Environment.

2. PC_DispClrScr() allows you to clear the entire display with space character.

3.setup the background(black) and foreground color(white).



Example #1

(2) OSInit()

1.initial the μC/OSII and create two tasks:

a.an idle task

Execute when no other task is ready to run.

b.a statistic task

computes CPU usage.

Example #1

(3) PC_DOSSaveReturn();

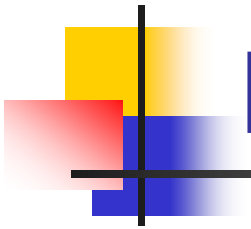
Allows your application to save the processor's important registers in order to properly return to DOS before you actually start multitasking with μC/OS II.

(4) PC_VectSet(μCOS,OSCtxSw);

used to set the contents of an interrupt-vector-table location.

(5)Random = OSSemCreate(1);

A semaphore created to guard access the random number generator function.



Example #1

(6) OSTaskCreate(TaskStart, (void*)0,
TaskStratStk[Task_Stk_SIZE-1], 0);

1.Before starting multitasking you have to create at least one task.

2.stack size is 1024 bytes.

3.the low the priority number the higher the priority

Example #1

(7) OSStart();

1.Called to start multitasking and give control to the μC/OS II kernels.

2.choice the highest priority rask to execute

(8)TaskStarst()

1.call the TaskStartDispInit() to initialize the display.

2. TaskStartDispInit() makes 25 consecutive calls to PC_DispStr() to fill the 25 consecutive lines of text of a typical DOS windows.

Example #1

3.TaskStart() to invokes the OS_ENTER_CRITICAL()
to disable interrupt.

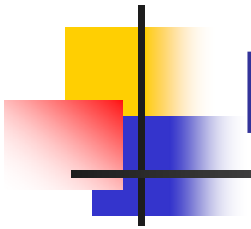
4.to use the DOS tick privode by PC(every 54.925ms).

5.set the tick rate to 200hz.

6.invokes OS_EXIT_CRITICAL() to reenable interrupt.

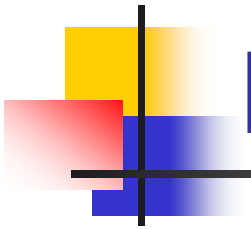
7.OSStatInit() is called to determine the speed of the CPU.

8.TaskStartCreateTasks() to lets μC/OSII manage more tasks



Example #1

9. TaskStartDisp() is called to display the information at the bottom of the DOS Window.
10. TaskStart() determines whether you pressed ESC key on your keyboard.



Example #1

(9) TaskStartCreateTasks()

1.initial 10 identical tasks call Task().

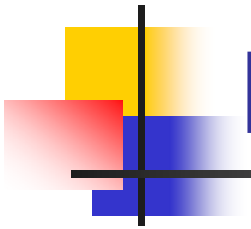
2.each task has its own stack space.

(10)Task()

1. x is the x-coordinate we want to display.

2. y is the y-coordinate we want to display.

3.OSTimeDly(1) to delay the calling function one clock.



Example #2

(1) PC_ElapsedInit() to initialize elapsed-time-measurement function that is used to measure execution time of OSTaskStkChk().

(2)OSTaskStartExt() instead of the OSTaskStart()

,because we modified the stack and want to check the Stack size at run time.

Example #2

(3) TaskStart()

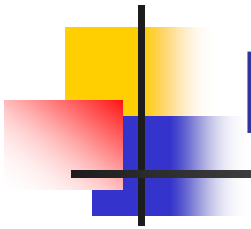
1.declared OSMboxCreate((void*)0) to initialize mailbox.

2.TaskStartCreateTasks() create six tasks using

OSTaskCreateExt().

(4)Task1()

1.OSTaskStkChk() is a service provided by μC/OSⅡ to allow your code to determine the actual stack usage of a task.



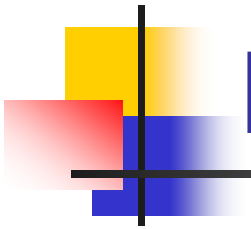
Example #2

(5) Task2() and Task3() display a spinning wheel

1.the two tasks are almost identical.

(6)Task4() send message to Task5() by posting the message in the mailbox.

1.Task4() waiting from Task5()’s acknowledgment by waiting on the ACKMox.



Example #3

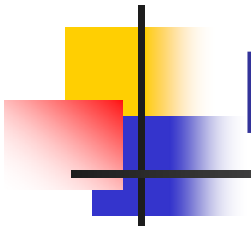
(1).create a data structure to hold task's additional information about a task.

(2).TASK_USER_DATA structure is allocated to hold information about each created.

(3)A message queue is used to send message.

a.message allow your task or ISR to send messages to others tasks.

b.create OS_EVENT and an array of pointers.



Example #3

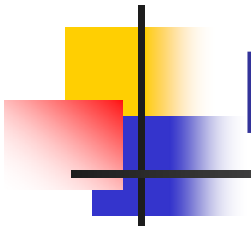
(3)use strcpy() function to store task name in the data structure.

(4)TaskStart() is created using OSTaskStartExt() and passed a pointer to its user data structure.

(5)each task is assigned an entry in the TaskUserData[] array,it assigned name .

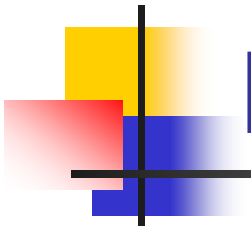
(6)Task1() waits forever for a message to arrive through amessage queue.

(7)Task2() send a message “Task2” to Task(),then Task2() delay 0.5sec.



Example #3

(8)Task3() and Task4() send message to each other and also wait 0.5 sec.



Example #3