



Embedded Operating System

Che-Wei Chang

chewei@mail.cgu.edu.tw

Department of Computer Science and Information
Engineering, Chang Gung University



Lab 2: Exercise on uC/OSII

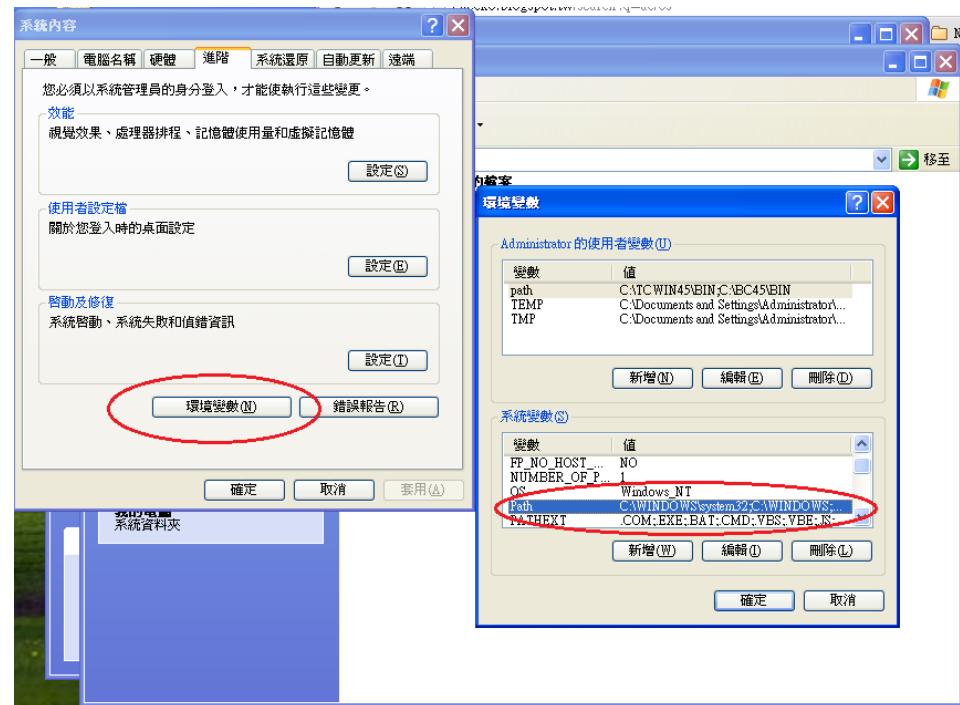
Requirements of μC/OS-II Emulator

- ▶ Operating System
 - Windows XP 32bits
 - Use virtual machine to install the OS
 - Install “Guest Additions” for Virtualbox
- ▶ Tools
 - Borland C++ compiler (V4.5)
 - BC45 is the compiler
 - Turbo Assembler
 - The assembler is in tasm
 - The source code and the emulation environment of μC/OS-II
 - SOFTWARE is the package
- ▶ Full Package
 - Download it from the course website with password: csie2018



Borland C++ Compiler

- ▶ Download Borland C++ and install it on your windows XP environment
 - Double click the “INSTALL.EXE”
- ▶ Add “;C:\BC45\BIN” to your system Path



Turbo Assembler

- ▶ Download Turbo assembler and unzip the file
- ▶ Copy “\tasm\BIN\TASM.EXE” to your “C:\BC45\BIN”
 - Include the missing assembler which is going to be used during we compile the source code of µC/OS-II

Compile µC/OS-II Example Code

- ▶ Download the source code and emulator µC/OS-II
 - It is recommended to put the source code package “SOFTWARE” directly in C:\
- ▶ Test the first example
 - Execute C:\SOFTWARE\uCOS-II\EX1_x86L\BC45\TEST\TEST.EXE
 - Press ECS to leave
- ▶ Rename or remove the executable file
 - Rename TEST.EXE
- ▶ Compile the µC/OS-II and the source code of the first example
 - Run C:\SOFTWARE\uCOS-II\EX1_x86L\BC45\TEST\MAKETEST.BAT
 - A new “TEST.EXE” will be created if we compile it successfully

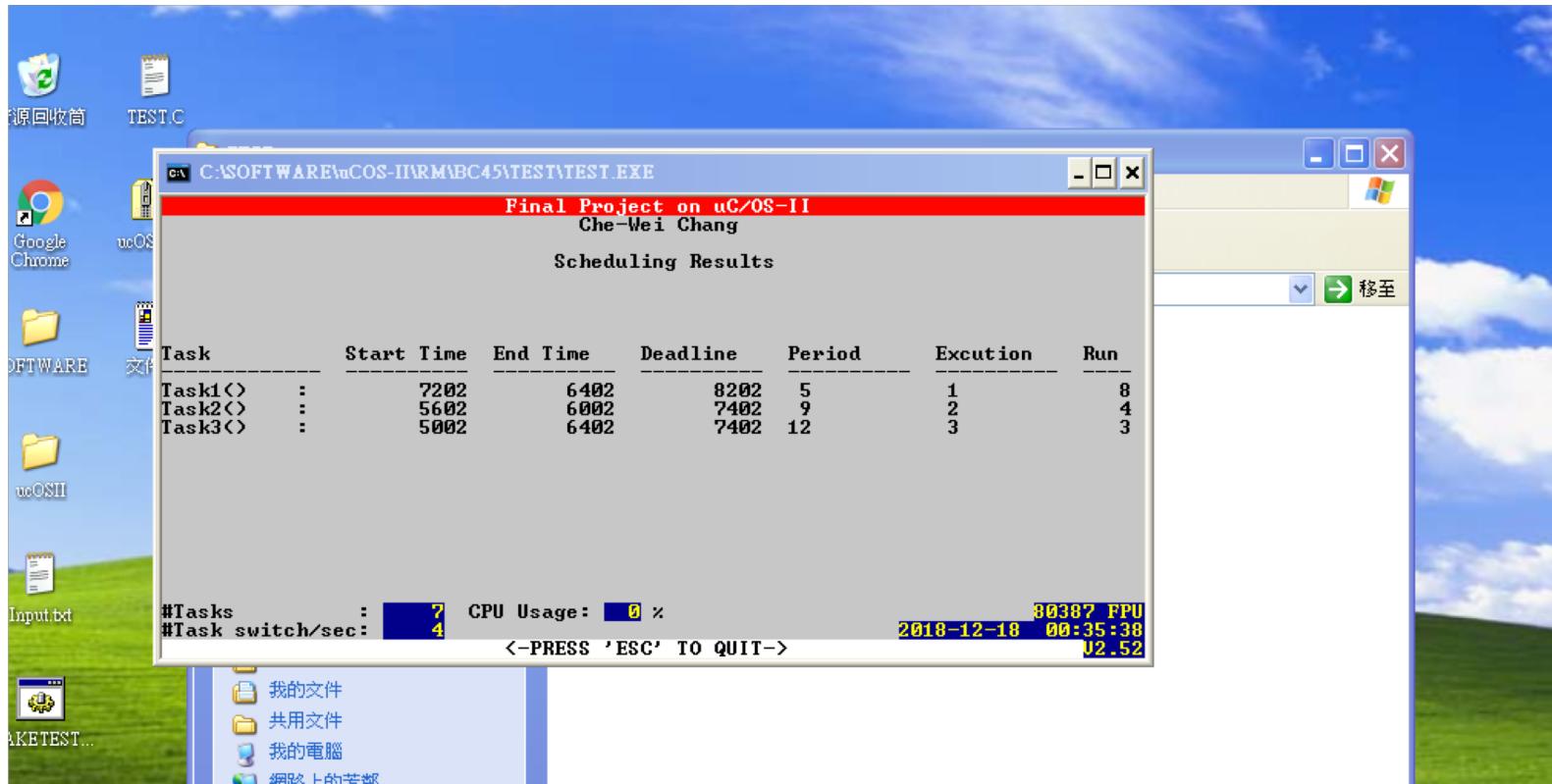
Common Mistakes

- ▶ Did you directly put the package “SOFTWARE” in C:\ ?
- ▶ Have you copied the correct file “TASM.EXE” to your “C:\BC45\BIN” directory?
- ▶ Did you set the Path correctly?
 - See the picture in Page 4
 - There is no space
- ▶ Fatal: Command arguments too long

Debuging Techniques

- ▶ Dump results to a file for further checking
- ▶ Use “cmd” environment to run the makefile
- ▶ Do the instructions in the batch file step by step by yourself
- ▶ Ask your friends

Check Point 1: Run the RM Sample Code



Check Point 2: Study and Use OSTaskChangePrio()

- ▶ Search OSTaskChangePrio in the textbook
 - Study the using and the implementation of the function OSTaskChangePrio()
- ▶ Requirements:
 - Based on the RM scheduler
 - Every tasks should complete their first execution at beginning
 - After every tasks complete their first execution, use RM for the priority management
 - Use OSTaskChangePrio() for the implementation

Check Point 3: Study and Use OSTaskResume() and OSTaskSuspend()

- ▶ Search OSTaskResume and OSTaskSuspend in the textbook
 - Study the using and the implementation of the functions OSTaskResume() and OSTaskSuspend()
- ▶ Requirements:
 - Based on the RM scheduler
 - Use OSTaskChangePrio() for the implementation
 - Details of the requirements of check point 3 is going to be announced before our lab exercise

Grading this Exercise

- ▶ Check point 1: 20%
- ▶ Check point 2: 20%
- ▶ Check point 3: 20%
- ▶ Report before the exercise: 20%
 - Two page A4, 12 pt font
 - Deadline is 10:00 2018/12/24
 - File name: EOS-Lab2-Study-Student_ID
 - File type: PDF or Word
 - Send it to my email: chewei@mail.cgu.edu.tw
 - Email title: EOS Lab2 Study Student_ID
- ▶ Report after the exercise: 20%
 - Two page A4, 12 pt font
 - Deadline is 10:00 2018/12/31
 - File name: EOS-Lab2-Report-Group_ID
 - File type: PDF or Word
 - Send it to my email: chewei@mail.cgu.edu.tw
 - Email title: EOS Lab2 Report Group_ID