

Embedded Software

(MicroC/OS-II: 1)

MicroC/OS-II: 프로그램 시작하기

Outline

- Overview
- First program
- What to do?



Overview - 개요

- 개요
 - Jean J. Labrosse가 만든 실시간 운영체제 (1992년)
 - 학교나 개인의 교육과 같은 비상업적 목적에 한해 자유로이 사용 가능한 공개소스
 - 소스코드의 수정 및 커널의 내부 구조를 이해하기 용이
 - 상업적 용도 : 각종 장비 개발이 가능
 - Avionics, medical devices, mobile handsets, consumer electronics \(\equiv \)
 - 상업적인 목적에 사용될 경우 라이센스를 따로 얻어야 함
 - 공식 사이트:

http://www.micrium.com/page/products/rtos/os-ii

Overview - 특징

- Source code
 - 깔끔한 코드
- Portable
 - 커널 코드의 대부분이 이식 가능한 ANSI C를 기반
 - 일부 마이크로프로세서에 국한된 부분은 어셈블러로 코딩
 - 8Bit, 16Bit, 32Bit 및 64Bit, DSP로도 Porting 가능
- Reliable
 - Safety-critical system 에도 사용할 수 있는 안전한 운영체제
 - ▶ FAA(Federal Aviation Administration) 승인 (2000년 7월)
- ROMable
 - 해당 C 컴파일러, 어셈블러, 링커, 로더가 있으면 내장이 용이
- Preemptive real-time kernel
 - 높은 우선 순위 작업이 먼저 수행
- Multitasking
 - 64개의 태스크 지원

Overview - 소프트웨어 구성요소

Application Software

μC/OS-II

(Processor-Independent Code)

```
OS_CORE.C uCOS_II.C
OS_MBOX.C uCOS_II.H
OS_MEM.C OS_MUTEX.C
OS_Q.C OS_FLAG.C
OS_TASK.C
OS_TIME.C
```

μC/OS-II Configuration (Application-Specific Code)

OS_CFG.H INCLUDES.H

μC/OS-II Port

(Processor-Specific Code)
OS_CPU.H
OS_CPU_A.ASM
OS_CPU_C.C

Software

Hardware

CPU

Timer



First program - 소스 구조

uC/OS-II source code

• \SOFTWARE 루트디렉토리

• \SOFTWARE\BLOCKS PC관련기능처리

• \SOFTWARE\TO 유틸리티

• \SOFTWARE\uCOS-II 메인디렉토리

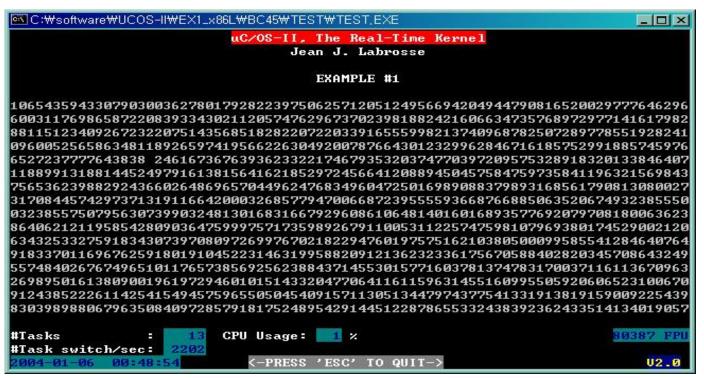
• \SOFTWARE\uCOS-II\Ix86L x86 리얼모드 포팅

\SOFTWARE\uCOS-II\lx86L-FP x86 FP 리얼모드 포팅

• \SOFTWARE\uCOS-II\SOURCE uC/OS-II 코드

Example 1

- 기본적인 멀티캐스팅 수행 과정을 보여줌
- 10개의 Task가 화면상 임의 위치에 각각 0에서 9사 이의 숫자를 표시





■ 소스 코드 내용

- C:\Software\uCOS-II\EX1_x86L\BC45\MAKE
 - MAKEAPP.BAT : Create EX1.EXE file
 - MAKECLEAN.BAT : Remove temporary files
 - EX1.MAK : Makefile
 - EX1.LIN : Linking files
- C:\Software\uCOS-II\EX1_x86L\BC45\SOURCE
 - INCLUDE.H : Header file configuration
 - OS_CFG.H : OS configuration
 - EX1.C : Example # 1 source code



```
#include "includes.h"
                                                                       (1)
                                /* 각 Task의 스택 크기(WORD 항목 단위)
#define TASK STK SIZE
                        512
                                                                        */
                                /* 생성할 Task 수
#define
                        10
                                                                        */
        N TASKS
                                           /* Task 스택
        TaskStk[N TASKS][TASK STK SIZE];
                                                                        */
OS STK
        TaskStartStk[TASK STK SIZE];
OS STK
                                            /* 각 Task에 넘겨줄 전달인자
char
        TaskData[N TASKS];
                                                                        */
OS EVENT
             *RandomSem;
```

필요한 모든 헤더 파일을 마스터 헤더 파일인 'INCLUDES.H'에 포함되어 있음



- (2) 멀티캐스팅을 시작하기 전에 최소한 하나 이상의 Task를 생성해야만 함 이 예제에서는 이 Task의 함수를 TaskStart()라고 명명
- (3) 멀티캐스팅을 시작하기 위해 OSStart()함수를 호출해서 uC/OS-II로 제어를 넘김 OSStart()를 호출하기 전에 반드시 1개 이상의 Task를 생성해야만 함



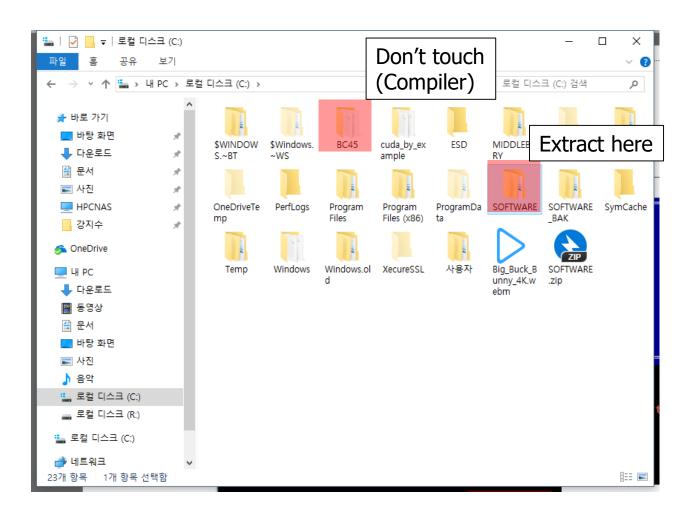
```
void TaskStart (void *data)
   UBYTE i;
   char
          s[100];
   WORD key;
                                                        /* 컴파일러 경고 방지 */
   data = data;
   OS ENTER CRITICAL();
   PC VectSet(0x08, OSTickISR);
                                               /* uC/OS-II 클럭 틱 ISR 설치 */
                                               /* 틱 주기 재설정
   PC SetTickRate(OS TICKS PER SEC);
                                                                         */
   OS EXIT CRITICAL();
                                              /* uC/OS-II 통계모듈 초기화 (1) */
   OSStatInit();
   TaskStartCreateTasks();
                                                                        (2)
   for (;;) {
                                             /* 키가 눌렸는지 확인
      if (PC GetKey(&key) == TRUE) {
           if (key == 0x1B) {
                                              /* 예, ESCAPE 키인지 확인
                                                                          */
                                              /* DOS로 복귀
               PC DOSReturn();
       OSCtxSwCtr = 0;
       OSTimeDlyHMSM(0, 0, 1, 0);
                                             /* 1초 지연
                                                                          */
```

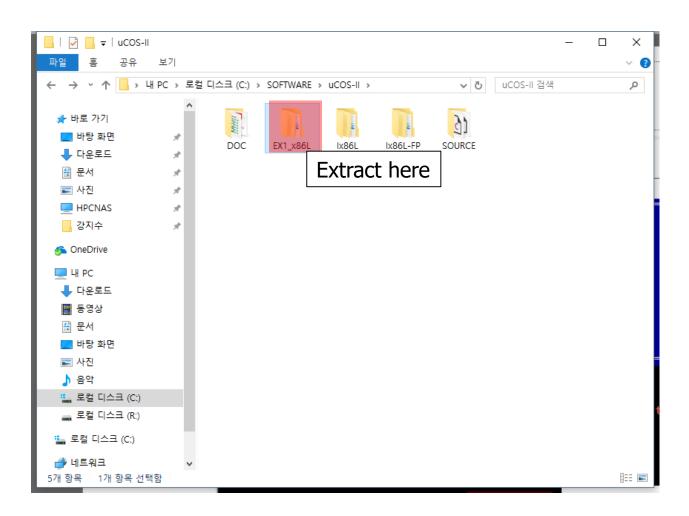


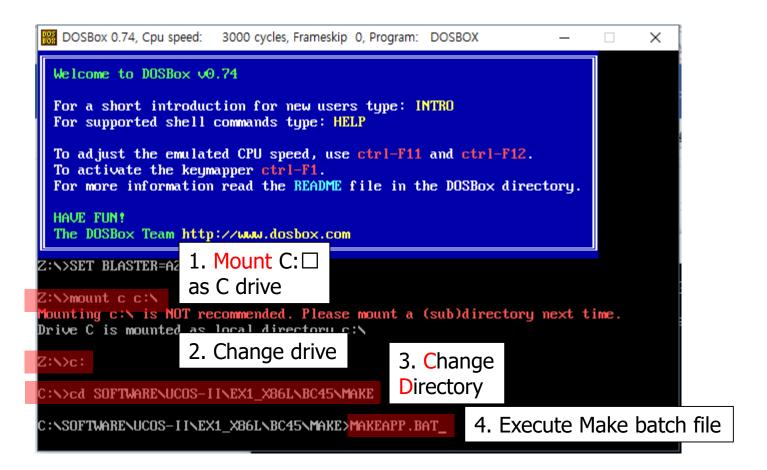
- for 루프는 Task()라는 N_TASKS개의 동일 Task를 초기화
 <각 Task는 1부터 10까지의 고유한 우선순위 숫자를 가짐>
- 우선순위 숫자가 낮을수록 우선순위가 높음



```
void Task (void *data)
{
 INT8U x;
 INT8U y;
 INT8U err;
 for (;;) {
   OSSemPend(RandomSem, 0, &err); /* 랜덤 넘버 함수를 사용하기 위해 세마포어 획득 */
                               /* 태스크 번호를 표시할 x 좌표 결정
                                                                    */
   x = random(80);
                             /* 태스크 번호를 표시할 Y 좌표 결정
   y = random(16);
                                                                    */
   OSSemPost (RandomSem); /* 세마포어 반납
                                                                    */
   PC DispChar(x, y + 5, *(char *)data, DISP FGND LIGHT GRAY);
                               /* 1 클럭 틱 지연
                                                                     */
   OSTimeDly(1);
                                                               (1)
```







```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
                                                                              ×
   1 File(s) copied.
        C:\BC45\BIN\BCC -c -ml -1 -G -0 -Ogemvlbpi -Z -d -n..\obj -k- -v -vi- -w
pro -IC:\BC45\INCLUDE -LC:\BC45\LIB uCOS II.C
Borland C++ 4.52 Copyright (c) 1987, 1994 Borland International
ucos_ii.c:
        COPY ...\SOURCE\EX1.C EX1.C
EX1.C
   1 File(s) copied.
        C:\BC45\BIN\BCC -c -ml -1 -G -O -Ogemvlbpi -Z -d -n..\obj -k- -v -vi- -w
pro -IC:\BC45\INCLUDE -LC:\BC45\LIB EX1.C
Borland C++ 4.52 Copyright (c) 1987, 1994 Borland International
ex1.c:
        C:\BC45\BIN\TLINK @EX1.LIN
Turbo Link Version 7.00 Copuright (c) 1987, 1994 Borland International
        COPY ... OBJNEX1.EXE ... MAKENEX1.EXE
EX1.EXE
  1 File(s) copied.
C:\SOFTWARE\UCOS-II\EX1 X86L\BC45\WORK>DEL
                                             EX1.MAK
C:\SOFTWARE\UCOS-II\EX1 X86L\BC45\WORK>DEL
                                             EX1.LIN
C:\SOFTWARE\UCOS-II\EX1_X86L\BC45\WORK>CD
                                              ...\MAKE
C:\SOFTWARE\UCOS-II\EX1 X86L\BC45\MAKE>
```



- If the compilation is failed,
 - Check the source code
 - Modify and save EX1.C to let Makefile know it (If you copy and paste EX1.C, Makefile doesn't notice it)
 - Restart DOSBox

```
C:\SOFTWARE\UCOS-II\EX1_X86L\BC45\WORK>DEL EX1.LIN
C:\SOFTWARE\UCOS-II\EX1_X86L\BC45\WORK>CD ..\MAKE
C:\SOFTWARE\UCOS-II\EX1_X86L\BC45\MAKE>
```



Assignments

- I-Class에 다음 슬라이드의 과제 제출
- 제출 파일
 - 소스 코드(EX1.C)
 - 보고서(.pdf)
 - 실행 화면 동영상 제출
 - You have to record and submit execution screen
- 파일명
 - 3주차_학번_이름
- 제출 기한
 - 다음주 화요일 자정까지

4

What to do?

- 1.4개의 태스크 생성
- 2. 각 태스크는 우선순위를 1,2,3,4 할당
- 3. 각 태스크에서 하는 일은 화면 전체를 색칠함
 - Red, Blue, Brown, Green
- 4. 다음과 같은 상황이 일어나게끔 구현
 - 1> Red 만 칠하기
 - 2> Red 와 Blue 만 칠하기
 - 3> Red, Blue, Brown 만 칠하기
 - 4> 4가지 색 모두 칠하기
 - (Optional) 위 상황이 Starvation에 의해 발생되게끔 구현
- 조건
 - 색깔이 칠해지는 순서는 중요하지 않음
 - 색깔의 변화를 육안으로 인지할 수 있어야 함
 - 색깔이 변하는 횟수에는 제한 조건이 없음

What to do?

- 1. Make and run 4 tasks
- 2. Assign each task a priority of 1 to 4
- 3. Each task prints color the entire screen
 - Red, Blue, Brown, Green
- 4. Implement below
 - 1> Colorize red only
 - 2> Colorize red and blue
 - 3> Colorize red, blue and brown
 - 4> Color in all 4 colors.
 - (Optional) Implement above with "starvation"

Conditions

- The order in which colors are painted is not important
- Be able to visually recognize changes in color
- There are no restrictions on the number of times the color changes



```
#define DISP FGND BLACK
                             0x00
#define DISP FGND BLUE
                             0x01
                              0x02
#define DISP FGND GREEN
#define DISP FGND CYAN
                             0x03
#define DISP FGND RED
                             0x04
#define DISP FGND PURPLE
                              0x05
#define DISP FGND BROWN
                               0x06
#define DISP FGND LIGHT GRAY
                                0x07
#define DISP FGND DARK GRAY
                                0x08
#define DISP FGND LIGHT BLUE
                                0x09
#define DISP FGND LIGHT GREEN
                                0x0A
#define DISP FGND LIGHT CYAN
                                0x0B
                               0x0C
#define DISP FGND LIGHT RED
#define DISP FGND LIGHT PURPLE
                                0x0D
#define DISP_FGND_YELLOW
                              0x0E
#define DISP_FGND_WHITE
                              0x0F
#define DISP BGND BLACK
                              0x00
#define DISP BGND BLUE
                             0x10
#define DISP BGND GREEN
                              0x20
#define DISP BGND CYAN
                             0x30
#define DISP BGND RED
                             0x40
#define DISP BGND PURPLE
                              0x50
#define DISP BGND BROWN
                               0x60
#define DISP BGND LIGHT GRAY
                                0x70
```

Upper 4bits for foreground color, Lower 4bits for background color

0x80

#define DISP BLINK



- 실습 후 C:\SOFTWARE 폴더 삭제해주세요.
- Before you go out, please remove C:\SOFTWARE folder.