

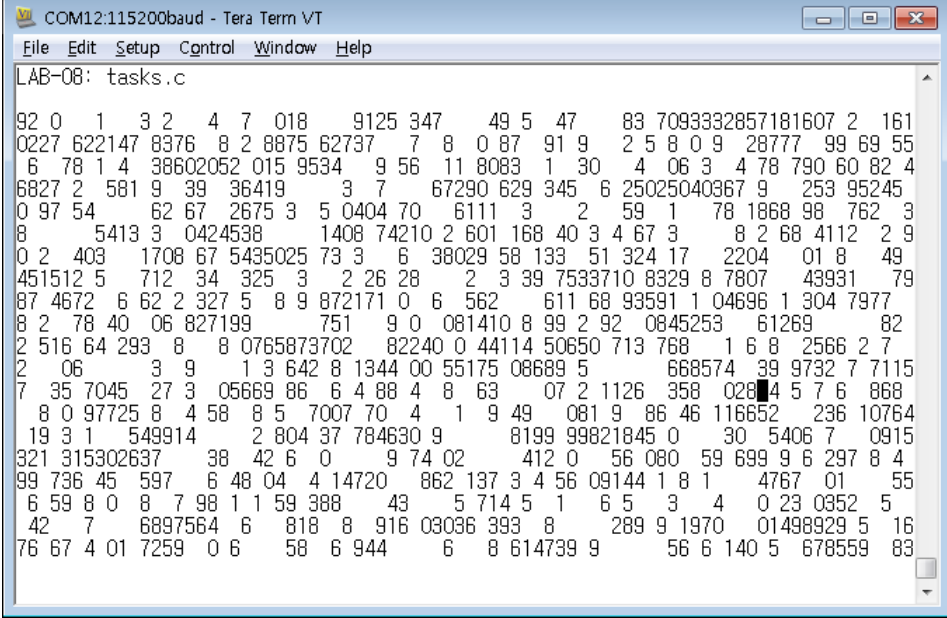
EA0028: 임베디드 소프트웨어 I 실습

상명대학교 컴퓨터과학과

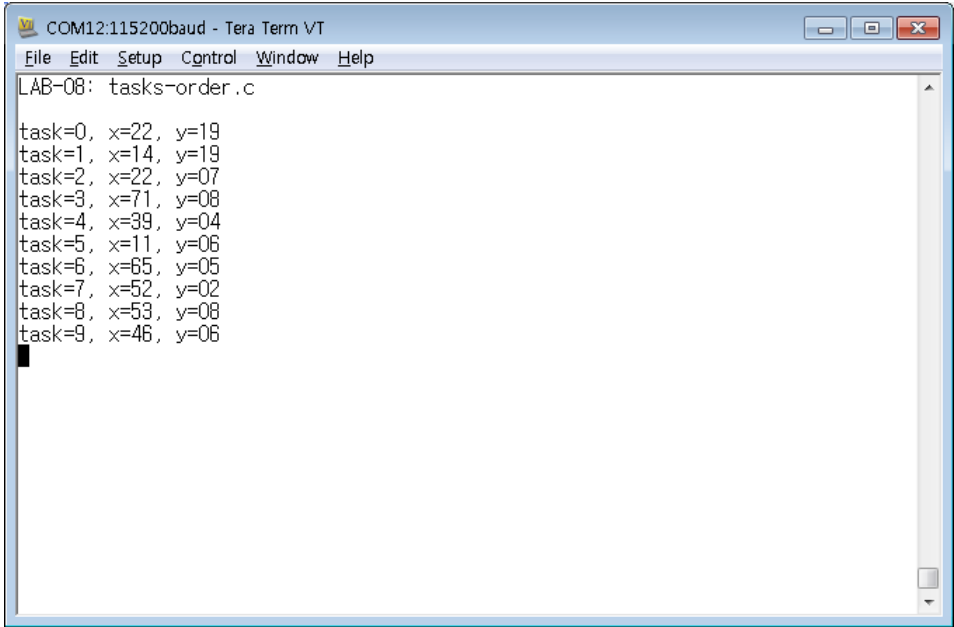
2019년 1학기

실습 번호	08	실습 점수	/9
실습 날짜	2019년 월 일	실습 폴더	~/es1/lab08
학생 이름		학번	
실습 제목	uC/OS-II Task Programming		
참고 자료	<ol style="list-style-type: none"> 1. 신동하, 1 Real Time System Concepts, 임베디드 소프트웨어 I 강의 자료, 2019. 2. 신동하, 2 Kernel Structure, 임베디드 소프트웨어 I 강의 자료, 2019. 3. 신동하, 3 Task Management, 임베디드 소프트웨어 I 강의 자료, 2019. 4. 신동하, 4 Time Management, 임베디드 소프트웨어 I 강의 자료, 2019. 5. 신동하, uC/OS-II Task Programming 기초 지식, 임베디드 소프트웨어 I 강의 자료, 2019. 6. Jean J. Labrosse, MicroC/OS-II The Real-Time Kernel Second Edition, CMP Books, 2002. 7. 성원호, MicroC/OS-II 실시간 커널, 에이콘 출판사, 2005. 		

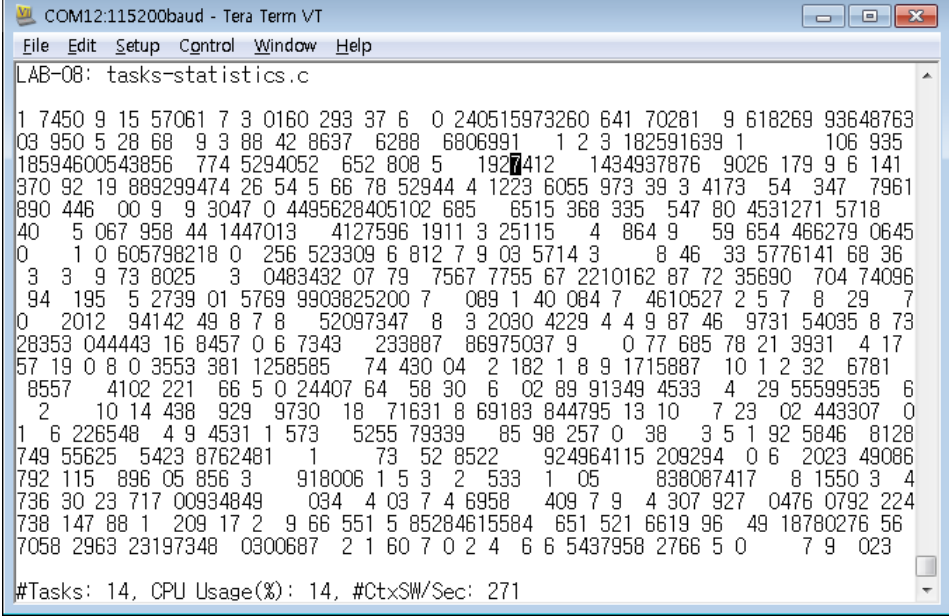
실습 번호	1	점수	3
실습 내용	<p>BeagleBone에서 다음 조건을 만족하는 실시간 커널 uC/OS-II 프로그램 파일 tasks.c를 작성하라.</p> <ul style="list-style-type: none"> • 이 프로그램의 함수 main()에서는 TaskStart()라는 task를 생성한다. 이 task의 priority는 10이다. • TaskStart()라는 task는 10개의 다른 task를 생성한다. 이때 생성되는 task의 priority는 11 ~ 20이다. • TaskStart()라는 task는 10개의 task를 생성한 후 UART에 연결된 화면을 clear하고 (0, 0) 위치에 "LAB-08: tasks.c"라는 string을 1초에 한번씩 계속 출력한다. • UART에 연결된 화면을 clear하기 위해서는 UART_clear()를 부른다. • TaskStart()라는 task에 의하여 생성된 task들은 5 Tick 마다 한번씩 $0 \leq x \leq 79$ 및 $0 \leq y \leq 19$ 를 만족하는 random 번호 x 및 y를 생성하여 UART에 연결된 화면의 위치 (x, y+2)에 task 번호(1자리 숫자임)를 출력한다. 이때 task 번호는 (task의 priority-11) 이다. • Random 번호를 생성할 때 안전을 위하여 semaphore를 사용하여 random 번호를 생성하는 함수가 여러 task에 의하여 동시에 불려지지 않게 한다. • 이 프로그램의 수행 화면은 다음 그림과 같다. 		

	
<p>실습 결과</p>	<p>1.1 작성한 파일 tasks.c를 프린터로 출력하고 출력물 우측 빈 여백에 연필로 직접 프로그램의 설명을 적어서 제출하기.</p> <p>1.2 위 그림과 같이 uC/OS-II 프로그램 tasks.bin을 BeagleBone에서 수행시킨 후 화면에 나타나는 동작 화면을 캡처하여 첨부하기.</p>

실습 번호	2	점수	3
<p>실습 내용</p>	<p>BeagleBone에서 다음 조건을 만족하는 실시간 커널 uC/OS-II 프로그램 파일 tasks-order.c를 작성하라.</p> <ul style="list-style-type: none"> 이 프로그램의 함수 main()에서는 TaskStart()라는 task를 생성한다. 이 task의 priority는 10이다. TaskStart()라는 task는 10개의 다른 task를 생성한다. 이때 생성되는 task의 priority는 11 ~ 21이다. TaskStart()라는 task는 10개의 task를 생성한 후 UART에 연결된 화면을 clear하고 (0, 0) 위치에 "LAB-08: tasks-order.c"라는 string을 1초에 한번씩 계속 출력한다. UART에 연결된 화면을 clear하기 위해서는 함수 UART_clear()를 부른다. TaskStart()라는 task에 의하여 생성된 task들은 1초 마다 한번씩 $0 \leq x \leq 79$ 및 $0 \leq y \leq 19$ 를 만족하는 random 번호 x 및 y를 생성하여 UART에 연결된 화면에 string "task=n, x=??, y=?"를 출력한다. 여기서 n은 task 번호로서 task의 priority-11이고 ??는 x 및 y이다. 위에서 출력되는 string의 순서는 task의 priority 순서로 출력되어야 한다. Random 번호를 생성할 때 안전을 위하여 semaphore를 사용하여 random 번호를 생성하는 함수가 여러 task에 의하여 동시에 불려지지 않게 한다. 이 프로그램의 수행 화면은 다음 그림과 같다. 		

	
실습 결과	<p>2.1 작성한 파일 tasks-order.c를 프린터로 출력하고 출력물 우측 빈 여백에 연필로 직접 프로그램의 설명을 적어서 제출하기.</p> <p>2.2 위 그림과 같이 uC/OS-II 프로그램 tasks-order.bin을 BeagleBone에서 수행시킨 후 화면에 나타나는 동작 화면을 캡처하여 첨부하기.</p>

실습 번호	3	점수	3
실습 내용	<p>Jupiter 교육용 보드에서 다음 조건을 만족하는 실시간 커널 uC/OS-II 프로그램 파일 tasks-statistic.c를 작성하라.</p> <ul style="list-style-type: none"> 이 프로그램은 실습 1의 프로그램과 유사하다. 단지 아래 기능이 더 추가된다. UART에 연결된 화면 (0, 23)에 그림 3과 같이 수행 중인 task의 개수, CPU의 usage 및 초당 context switch 회수를 출력한다. 이 프로그램의 수행 화면은 다음 그림과 같다. 		

	
<p>실습 결과</p>	<p>3.1 작성한 파일 tasks-statistic.c를 프린터로 출력하고 출력물 우측 빈 여백에 연필로 직접 프로그램의 설명을 적어서 제출하기.</p> <p>3.2 앞 그림과 같이 uC/OS-II 프로그램 tasks-statistic.bin을 BeagleBone에서 수행시킨 후 화면에 나타나는 동작 화면을 캡처하여 첨부하기.</p>