

Project 보고서

2316 황윤서

1. Project 개요

1.1. 프로젝트 목표

본 프로젝트는 Text File에 기록된 S/W 블록들을 초기화 시키고, S/W 블록의 이상동작 (블럭 다운) 발생 시 재초기화를 수행하고 그 기록을 출력하는 Application을 개발하는 것을 목표로 한다.

1.2. 프로젝트 구현

1.2.1. 코드 파일에 대한 설명

본 프로젝트는 각각 C언어 파일 1개와 헤더 파일 1개로 구성된 모듈 4개와 S/W블록 정보/기동정보를 저장하기 위한 구조체가 선언되어 있는 sw_info.h, 메인 로직을 수행하는 main.c파일로 구성되며 각 파일이 수행하는 역할/기능은 아래와 같다.

sw_info.h

: S/W 블록 정보와 기동 정보를 저장하기 위한 구조체가 선언되어 있는 헤더 파일이다.

convert_to_string.c/convert_to_string.h

: 어떠한 값을 문자열로 바꿔주는 역할을 하며, 구체적으로 status값과 time값을 문자열로 바꿔주는 convertStatusToString() 함수와 convertTimeToString() 함수가 포함되어 있다.

sw_log.c/sw_log.h

: ./log/restart.txt에 S/W 블록 기동 로그를 남기는 역할을 하며 이를 위한 writeSwStartLog() 함수가 여기에 작성되어 있다.

sw_read_info.c/sw_read_info.h

: 프로그램 실행 경로에 있는 FileList.txt를 읽어 들여 S/W 블록 정보를 가져오는 역할을 하며, 이 역할을 수행하는 readFileList() 함수와 readFileList() 함수에서 필요로 하는 내부 함수들로 구성되어 있다.

sw_start.c/sw_start.h

: 본 프로젝트의 핵심적인 기능인 S/W 블록 초기화와 재초기화 기능과 S/W 블록 기동 정보를 출력하는 기능을 수행한다. 각각 S/W블록 초기화, S/W블록 재초기화, S/W블록 기동 정보 출력 기능을 담당하는 swInitStart() 함수와 swRestart() 함수, printSwStartInfos() 함수, 그리고 그 외 내부 함수들로 구성되어 있다.

main.c

: 본 프로젝트의 로직을 구현한 파일로 위에서 언급된 모듈들의 함수들을 호출하여 S/W 블록 초기화/재초기화/기동정보 출력/기동정보 로그 출력 등의 기능을 수행한다.

1.2.2. 코드 구조

각 모듈은 main.c를 최상위 루트로 하는 계층 구조를 형성한다. 이 계층 구조에서 상위 모듈은 하위 모듈을 참조할 수 있으나 하위 모듈은 상위 모듈을 참조할 수 없다. 계층 구조와 각 모듈들이 참조하고 있는 하위 모듈은 아래와 같다.

```
- main.c : sw_info, sw_read_info, sw_start
- - sw_start : sw_info, sw_log, convert_to_string
- - sw_read_info : sw_info
- - - sw_log : sw_info, convert_to_string
- - - - sw_info
- - - - convert_to_string
```

1.2.3. 개발 및 빌드 환경

본 프로젝트는 Ubuntu 64-bit 18.04 환경에서 vim 8.0.1453을 이용하여 개발하였으며 gcc 7.5.0 환경에서 빌드하였다. 또한 make를 이용하여 테스트를 위한 S/W 블록과 본 프로젝트의 실행파일들을 생성하였다. 루트 디렉토리에서 src 디렉토리를 생성하여 이 디렉토리에 소스파일들을 작성하였고 실행 파일들을 bin 디렉토리를 생성하여 위치시켰다. 원활한 테스트를 위해 빌드를 할 때마다 프로그램 로그 파일을 자동으로 삭제하고 bin 폴더 내부의 실행파일들을 삭제하여 다시 빌드되도록 쉘 스크립트를 이용하여 build.sh를 작성하여 자동화시켰다.

1.2.4. 실행 결과

Application을 실행하게 되면 아래와 같이 S/W블록이 계속 기동되며 그 정보를 출력한다.

S/W Block Name	Restart Count	Start Time	Reason
SwBlock1	1	2020.07.24. 10:39:42	Init
SwBlock2	1	2020.07.24. 10:39:42	Init
SwBlock3	1	2020.07.24. 10:39:42	Init
SwBlock4	1	2020.07.24. 10:39:42	Init
SwBlock5	1	2020.07.24. 10:39:42	Init
S/W Block Name	Restart Count	Start Time	Reason
SwBlock1	1	2020.07.24. 10:39:42	Init
SwBlock2	1	2020.07.24. 10:39:42	Init
SwBlock3	1	2020.07.24. 10:39:42	Init
SwBlock4	2	2020.07.24. 10:39:46	Signal(7, Bus error)
SwBlock5	1	2020.07.24. 10:39:42	Init
S/W Block Name	Restart Count	Start Time	Reason
SwBlock1	1	2020.07.24. 10:39:42	Init
SwBlock2	1	2020.07.24. 10:39:42	Init
SwBlock3	1	2020.07.24. 10:39:42	Init
SwBlock4	2	2020.07.24. 10:39:46	Signal(7, Bus error)
SwBlock5	2	2020.07.24. 10:39:47	Exit(4)
S/W Block Name	Restart Count	Start Time	Reason
SwBlock1	2	2020.07.24. 10:39:50	Signal(9, Killed)
SwBlock2	1	2020.07.24. 10:39:42	Init
SwBlock3	1	2020.07.24. 10:39:42	Init
SwBlock4	2	2020.07.24. 10:39:46	Signal(7, Bus error)
SwBlock5	2	2020.07.24. 10:39:47	Exit(4)
S/W Block Name	Restart Count	Start Time	Reason
SwBlock1	2	2020.07.24. 10:39:50	Signal(9, Killed)
SwBlock2	1	2020.07.24. 10:39:42	Init
SwBlock3	1	2020.07.24. 10:39:42	Init
SwBlock4	3	2020.07.24. 10:39:51	Exit(200)
SwBlock5	2	2020.07.24. 10:39:47	Exit(4)

2. 시험 결과

2.1. S/W 블록 초기화 기능 시험 결과

2.1.1. Test Case 1. Text File Parsing 시험 결과

S/W 블록 정보 파일과 S/W 블록을 준비한 상태로 Application을 실행하고 S/W 블록 기동 정보를 조회하였을 때 모든 S/W 블록의 이름이 출력된 것과 printf함수를 이용하여 임시 코드를 작성하여 테스트 한 결과를 보았을 때 정상적으로 Text File Parsing 동작이 수행되었음을 확인할 있었다.

2.1.2. Test Case 2. Log File Write 시험 결과

Application을 실행하였을 때 log 디렉토리가 생성되고 하위 경로에 restart.txt 파일이 정상적으로 생성되었고 cat 명령을 활용하여 그 내용을 확인하였을 때 “SwBlock1 2020.07.24. 09:48:56 Init 1”과 같이 S/W 블록 기동 정보가 출력되는 것으로 보아 Log File Write 동작이 정상적으로 수행됨을 알 수 있었다.

2.2. S/W 블록 재초기화 시험 결과

2.2.1. Test Case 1. 재초기화 시험 결과

Application을 실행하고 초기화된 S/W 블록을 kill 시키고 “ps -al | grep SwBlock” 명령을 수행하여 S/W 블록들의 프로세스 정보를 확인하였을 때, 죽은 S/W 블록 이름 옆의 <defunct>가 확인되었다. 이후 다시 “ps -al | grep SwBlock” 명령을 수행하였을 때 <defunct>가 사라지는 것으로 보아 S/W 블록이 정상적으로 재초기화 되었음을 알 수 있었다. 이는 ./log/restart.txt에서 restart count 값이 정상적으로 증가하였음을 통해서도 확인할 수 있었다.

3. Source Code

본 프로젝트의 소스코드는 아래와 같은 경로로 별도 첨부한다.

- <https://github.com/yunseo-h68/execute-swblocks>

4. Project 수행에 따른 경험 및 느낀 점

본 프로젝트를 수행함으로써 얻은 경험은 실패의 경험이다. 구조체와 포인터 등을 복합적으로 사용하고 여러 모듈로 분할하면서 코드는 깔끔해지지만 생각해야 할 것은 많아졌고 이로 인해 여러 실수가 발생했다. 그러나 오류를 해결하는 과정에서 고민한 것은 더 나은 결과물로 이어졌고 이는 더 많은 성취감을 스스로에게 주었다고 생각한다.

본 보고서의 1,2를 보면 알 수 있다시피 본인은 본 프로젝트를 수행하면서 리눅스 명령어, 셸 프로그래밍, Make, 리눅스 API 등을 복합적으로 사용하였다. 이는 더 많은 경험과 흐름을 깨달을 수 있게 하는 지혜를 주었다고 생각한다. 사실만을 말하자면 본 프로젝트의 수행 과정은 결코 쉽지 않았다. 그리고 이 과정은 내 실력의 부족으로부터 비롯되었음을 나는 알고 있다. 그러나 그만큼 어려운 과정을 통해 작업을 수행하였기에 1학기 동안 배운 리눅스 프로그래밍 지식들을 디테일하게 복습할 수 있었고 그 결과 이 지식을 오랫동안 기억할 것이라는 확신을 얻었다. 본 프로젝트는 매우 성공적이었음을 말하고 싶다.