과목 명: 임베디드시스템소프트웨어

<<HW 1>>

**서강대학교 컴퓨터학과**

**20141582**

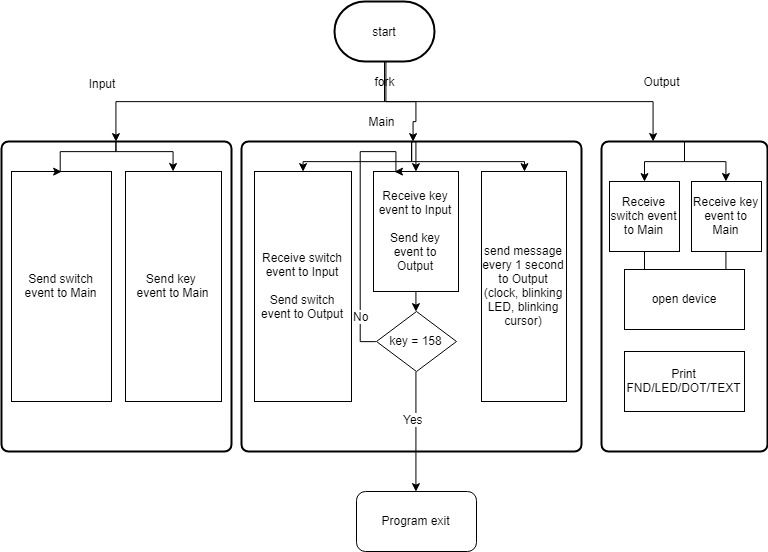
**정 필 립**

# 프로그램 개요

Main process는 input process와 output process를 fork하고 message-queue 방법을 이용한 IPC 통신을 한다. Input process는 계산을 위해 입력을 받는 장치를 관리하고, Main process는 연산의 실질적인 수행을 한다. Output porcess는 Main process로부터 전달 받은 값을 해당하는 장치에 출력하는 역할을 한다.

# 프로그램 설명

## 프로그램 흐름도



## 모듈 이름 : void \*switch\_function(void \*data)

### 기능

switch device를 open하고 눌려진 값을 message queue로 전송한다.

## 모듈 이름: void \*key\_function(void \*data)

### 기능

key device를 open하고 눌려진 값을 message queue로 전송한다.

## 모듈이름: int input\_main(void)

### 기능

input process에서 switch 버튼 event를 담당하는 thread와 key 버튼 event를 담당하는 thread를 생성한다.

### 사용변수

pthread\_t p\_thread[2] – 생성할 2개의 thread를 저장

## 모듈이름: void shift\_down()

### 기능

ddong\_matrix를 한 줄씩 아래로 내리고 맨 윗 줄에는 새로 생성한 line을 덮어씌운다.

## 모듈이름: void create\_random\_line()

### 기능

추가구현에 사용되는 함수. 매 초마다 새로운 장애물이 한 줄씩 random하게 생성된다. 미리 결정된 level에 따라서 난이도 조절을 위해 처음 첫 줄에는 장애물이 level 개 만큼이 생성되고, 다음 줄은 level-1 개 만큼의 장애물이 생성되고, 그 다음 줄은 다시 level 개의 장애물이 생성된다.

### 사용변수

int count – 생성된 장애물의 개수를 저장함. 목표 개수를 이미 생성했다면 나머지에는 장애물을 만들지 못하도록 함.

int func\_level – level에 따라 생성해야 할 장애물의 개수를 결정함.

## 모듈이름: void convert\_ddong()

### 기능

추가구현에서 장애물의 위치가 기록된 10\*7 행렬을 dot\_matrix에 출력할 수 있는 format으로 변환해줌. (한 줄씩 16진법으로 변환시킴.)

## 모듈이름: void convert\_tmp()

### 기능

모드 4에 사용되는 함수. 내가 선택한 위치와 커서 위치가 함께 저장되어 있는 0, 1로 이루어진 10\*7 행렬을 dot\_matrix에 출력할 수 있는 format으로 변환해줌.

## 모듈이름: void convert\_draw()

### 기능

내가 선택한 위치만 기록되어 있는 10\*7 행렬을 dot\_matrix에 출력할 수 있는 format으로 변환해줌.

## 모듈이름: int convert\_to\_10(int not, int num)

### 기능

해당 진법(not)의 num를 10진수로 변환해주고 그 값을 return 해준다.

### 사용변수

int not – 무슨 진법인지 저장함.( ex. 2, 4, 8, 16, 10 )

int num – 진법으로 변환된 값을 저장함. ( ex. 2진법일 때 num = 10010 )

int result – 10진법으로 변환한 값을 저장함.

## 모듈이름: int convert\_to\_notation(int not, int num)

### 기능

num(10진수)를 해당 진법(not)으로 변환해주고 그 값을 return 해준다.

### 사용변수

int not – 무슨 진법으로 변환할 지

int num – 변환시킬 10진수

int result – 해당 진법으로 변환된 값

## 모듈이름: void matrix\_empty()

### 기능

0, 1로 이루어진 10\*7 행렬의 값을 전부 0으로 초기화 해준다.

## 모듈이름: void matrix\_reversed()

### 기능

0, 1로 이루어진 10\*7 행렬의 값을 전부 반전시킨다.

## 모듈이름: void \*input\_msgtype1()

### 기능

무한 반복문을 통해 input process에서 보낸 switch event 메시지(type 1)를 끊임없이 읽는다. switch 문을 이용해 현재 mode에 맞춰서 적절한 연산을 수행하고 output process로 message(type 3)을 전송한다.

### 사용변수

int rcvtype – input process로부터 읽을 메시지 타입 값

int sndtype – output process로 보낼 메시지 타입 값

int switch\_num – 읽은 메시지에서 입력된 switch button값을 정수형으로 전환시킨 변수

int switch\_count – mode 3에서 같은 번호를 몇 번 눌렀는지 저장하는 변수

char next\_char – mode 3에서 해당 번호를 눌렀을 때 입력되어야 할 문자

msgbuf mybuf2 – output process로 보낼 구조체

## 모듈이름: void \*input\_msgtype2()

### 기능

무한 반복문을 통해 input process로부터 key event message(type 2)에 해당하는 메시지를 끊임없이 읽는다. mode를 증가시키고, 감소시키며, 프로그램 종료를 담당한다. 또한 모드가 수정되었을 때, default 설정값을 output process로 전송시키고 출력하게끔 한다.

### 사용변수

int rcvtype – input process로부터 읽을 메시지 타입

int sndtype – output process로 보낼 메시지 타입

msgbuf mybuf2 – output process로 보낼 데이터 구조체

## 모듈이름: void \*switch\_3\_4()

### 기능

1초마다 메시지를 전송하는 thread 함수. mode 1 에서 modify mode일 때 1초마다 3, 4번 LED가 깜박이도록 메시지를 output process로 전송해준다. 일반 clock 모드일 경우, sleep(1)을 60번 수행하고 그 때마다 메시지를 전송해줌으로 1분 마다 시간이 변화하는 것을 구현함. mode 4에서 현재 위치의 커서를 깜박이도록 1초마다 메시지를 전송해준다. 추가구현에서 1초마다 장애물이 위에서 점차 떨어지도록 한 줄 씩 아래로 shift하고 이것을 출력하도록 1초마다 메시지를 전송한다.

### 사용변수

msgbuf mybuf2 – 1초마다 output process로 전송할 데이터

## 모듈이름: void show\_fnd(int hourMin\_int)

### 기능

hourMin\_int에 해당하는 값을 FND에 출력시킴.

## 모듈이름: void show\_led(int num)

### 기능

num 값에 해당하는 LED를 출력시킴.

## 모듈이름: void show\_dot\_matrix(unsigned char matrix[10])

### 기능

matrix값을 dot matrix에 출력시킴.

## 모듈이름: void show\_text(unsigned char arr[33])

### 기능

문자열 arr를 TEXT\_LCD에 출력시킴.

## 모듈이름: void \*output\_msgtype3()

### 기능

main process에서 전송한 switch event에 해당하는 message(type 3)를 끊임없이 읽는다. 각 모드에 대해서 device를 open해주고 출력시킨다.

## 모듈이름: void \*output\_msgtype4()

### 기능

main process에서 전송한 key event에 해당하는 message(type 4)를 끊임없이 읽는다. 각 모드에 대해서 device를 open해주고 출력시킨다.

## 모듈이름: void output\_main()

### 기능

message type 3과 type 4를 읽을 thread 2개를 생성한다.

# 전역 변수 정의

## struct msgbuf

input process로부터 전송 받은 데이터와 process로부터 전송할 데이터를 저장할 구조체

## int mode

현재 모드

mode 1 : clock

mode 2 : counter

mode 3 : text editor

mode 4 : draw board

mode 5 : 똥 피하기 (추가구현)

## int switch\_num

switch button을 눌렀을 때의 정수형 값. ( ex. 동시에 5, 6번 switch를 눌렀다면 56이 저장됨 )

.

## pid\_t intput\_pid, output\_pid

input, output process의 id값

## int switch\_flag

1인 경우, 시계를 수정하지 못하고

-1인 경우, 수정할 수 있도록 함.

## int hourMin

시간을 HHMM형식의 정수값으로 저장함.

## int kkk

60초를 셀 때 사용됨.

## int counter\_count

mode 2에서 버튼이 몇 번 눌렸는지 저장함.

## int notation[4]

몇 진법으로 할 지 저장한 배열.

## int notation\_led[4]

해당 진법을 가리키는 LED 값을 저장한 배열

## int notation\_idx

switch 1을 누를 때마다, 1씩 증가하며 몇 진법인지, 몇 번 LED를 켜야할지에 사용.

## int text\_count

mode 3에서 버튼이 몇 번 눌렸는지 저장함.

## int dot\_idx

mode 3에서 dot matrix에 “A”를 출력할지 “1”을 출력할지 결정함. 0 인 경우 A, 1인 경우 1을 가리킴.

## int before\_switch

이전에 누른 switch가 몇 번인지 기록함. 같은 번호를 눌렀을 때 문자판처럼 새로운 문자가 입력되는 것이 아닌 다른 문자로 바뀌도록 할 때 사용함.

## unsigned char string[33]

TEXT\_LCD에 출력시킬 문자열

## unsigned char fpga\_number[3][MAX\_ROW]

dot matrix에 출력시킬 값을 미리 계산해서 저장함. “A”, “1”, NULL이 저장되어 있음.

## unsigned char text\_lcd\_1[10]

mode 3에서 버튼이 숫자를 가리킬 때, 누른 switch number에 따른 문자형 숫자를 저장함.

## unsigned char text\_lcd\_A[10][3]

mode 3에서 버튼이 문자를 가리킬 때, 문자판을 배열에 저장해 놓음.

## int draw\_count

mode 4에서 버튼이 몇 번 눌렸는지 저장.

## int cursor\_flag

cursor가 표시되어야 하는지 판단함. 1인 경우 표시, -1인 경우 표시하지 않음.

## unsigned char draw\_matrix[MAX\_ROW][MAX\_COL]

내가 선택한 위치들이 저장되어 있는 배열. dot matrix 한 점을 2차원 배열의 한 칸을 가리키며 0인 경우 선택하지 않음, 1인 경우 선택함을 뜻함.

## unsigned char draw[MAX\_ROW]

draw\_matrix를 dot matrix에 출력하기 쉬운 형태로 변환시킨 값

## unsigned int matrix[MAX\_ROW][MAX\_COL]

내가 선택한 위치와 내 커서가 위치한 곳을 포함한 배열. 커서위치에 불이 들어올 때, 내 위치에 해당하는 값은 1, 불이 꺼질 경우 내 위치에 해당하는 값은 0으로 변환됨.

## unsigned char tmp[MAX\_ROW][MAX\_COL]

matrix를 dot matrix에 출력하기 쉬운 형태로 변환시킨 값.

## unsigned int ddong\_matrix[MAX\_ROW][MAX\_COL]

장애물(똥)의 위치가 저장되어 있는 배열. 1초마다 해당 장애물이 한 줄 씩 아래로 내려오며 맨 윗줄에는 새로운 장애물 위치가 배치됨. 1인 경우 장애물이 있다고 판단.

## unsigned char ddong[MAX\_ROW]

ddong\_matrix를 dot matrix에 출력하기 쉬운 형태로 변환시킨 값.

## unsigned int new\_line[MAX\_COL]

새로운 장애물의 위치를 저장한 변수

## char manual[33]

mode 5에 진입했을 때, 게임 조작 방법에 대한 문자열을 저장함. TEXT\_LCD에 출력시킬 때 사용함.

## int score

게임 점수를 저장함. 1줄 씩 내려올 때 1점씩 증가함.

## int level

게임의 레벨. level=3 인 경우 장애물이 한 줄에 최대 3개 생성됨.

## int level\_flag

게임의 난이도를 생각하여 처음에 최대 3개 생성되면, 다음 줄에는 최대 2개 생성함. 처음에 최대 2개가 생성되었으면, 다음 줄에는 최대 3개 생성하도록 함.

## int ddong\_x

현재 내 위치 x좌표

## int ddong\_y

현재 내 위치 y 좌표

## int start\_flag

switch 5를 누르면 게임이 시작되고 내 위치와 장애물이 닿으면 게임이 종료됨. 1인 경우 게임을 시작하고, 0인 경우 게임을 종료함.

# 추가구현

## 설명

처음 dot matrix의 맨 아랫줄 가운데에서 시작함. 좌/우 방향으로만 움직이며 장애물(똥)에 닿을 경우 게임이 종료됨. 장애물의 위치는 랜덤하게 생성되며 기본 설정된 level에 따라 난이도가 결정됨. 장애물은 1초마다 1칸 씩 내려오며 장애물을 피하며 오래 버티는 게임임. 1줄 내려올 때마다 점수가 1점씩 증가함.

## 작동법

switch 5 : 게임 시작 / 한 번 시작하면 죽을 때까지 멈출 수 없음.

switch 4 : 좌로 이동

switch 6 : 우로 이동

죽고 나서 다시 게임을 하고 싶으면 다른 모드로 이동했다가 다시 돌아와야함.