

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

# C Project Design & Implementation

국 중 진 (tipsiness@gmail.com)

# Make

## ❖ *Multiple Source Files*

```
diary.h :  
#include <stdio.h>  
int memo();  
int calendar();
```

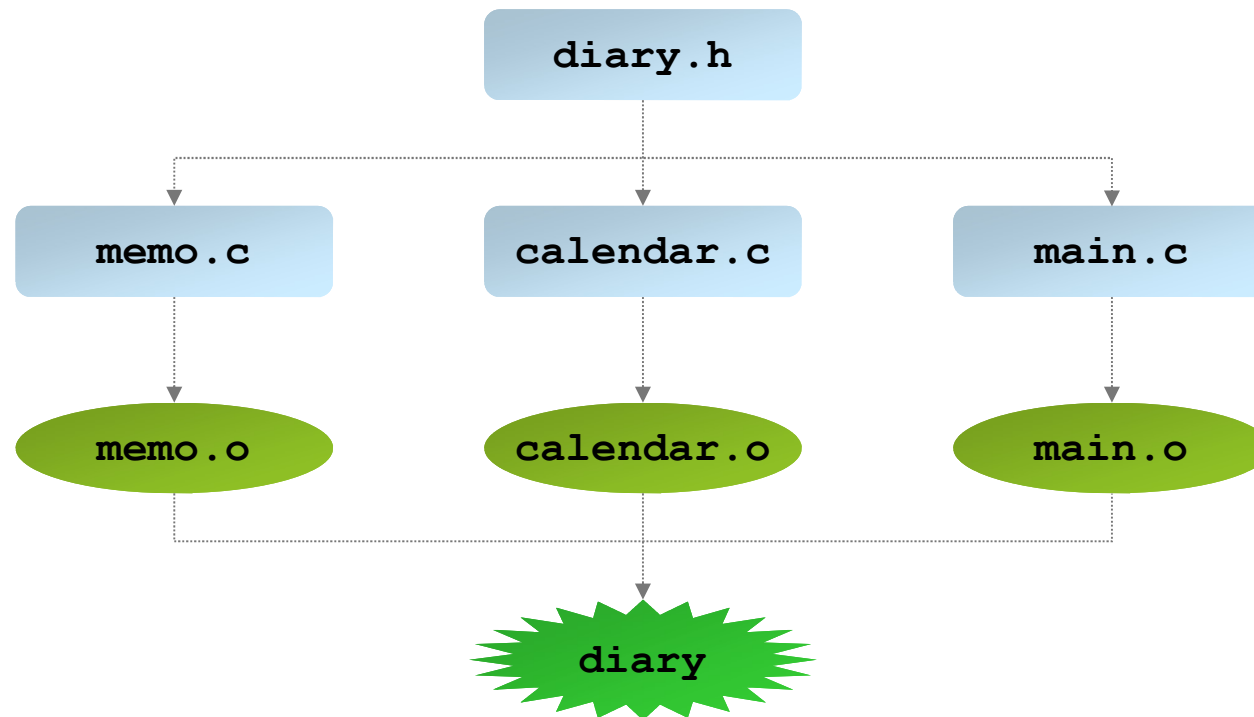
```
memo.c :  
#include "diary.h"  
int memo() {  
    printf("function  
memo.\n");  
    return 0;  
}
```

```
main.c :  
#include "diary.h"  
int main() {  
    memo();  
    calendar();  
    return 0;  
}
```

```
calendar.c :  
#include "diary.h"  
int calendar() {  
    printf("function  
calendar.\n");  
    return 0;  
}
```

# Make

## ❖ *Multiple Source Files*



# Make

## ❖ *Multiple Source Files*

### ▶ *How to build:*

```
$ gcc -c memo.c
```

```
$ gcc -c calendar.c
```

```
$ gcc -c main.c
```

```
$ gcc memo.o calendar.o main.o -o diary
```

or

```
$ gcc memo.c calendar.c main.c -o diary
```

# Make

## ❖ *Multiple Source Files*

### ▶ Makefile v1

```
#Makefile
diary: main.o memo.o calendar.o
(TAB) gcc -o diary main.o memo.o calendar.o
main.o: main.c
(TAB) gcc -c main.c
memo.o: memo.c
(TAB) gcc -c memo.c
calendar.o: calendar.c
(TAB) gcc -c calendar.c
```

# Make

## ❖ *Multiple Source Files*

### ► Makefile Format

# Makefile or makefile

comments

all : diary

diary : memo.o calendar.o main.o

<tab> gcc -W -Wall -o diary memo.o calendar.o  
main.o

memo.o : memo.c

rule

<tab> gcc -W -Wall -c memo.c

command

calendar : calendar.c

<tab> gcc -W -Wall -c -o calendar.o calendar.c

main.o : main.c

<tab> gcc -W -Wall -c main.c

# Make

## ❖ *Multiple Source Files*

### ► Makefile Format

```
# Makefile or makefile
```

```
CC=gcc
```

```
CFLAGS=-W -Wall
```

macro  
definition

```
all : diary
```

```
diary : memo.o calendar.o main.o
```

```
<tab> $(CC) $(CFLAGS) -o diary memo.o calendar.o main.o
```

```
memo.o : memo.c
```

```
<tab> $(CC) $(CFLAGS) -c memo.c
```

```
calendar : calendar.c
```

```
<tab> $(CC) $(CFLAGS) -c calendar.c -o calendar.o
```

```
main.o : main.c
```

```
<tab> $(CC) $(CFLAGS) -c main.c
```

# Make

## ❖ *Multiple Source Files*

### ▶ Makefile v2

```
CC = gcc
CFLAGS = -c

diary: main.o memo.o calendar.o
    $(CC) -o diary main.o memo.o calendar.o

main.o: main.c
    $(CC) $(CFLAGS) main.c

memo.o: memo.c
    $(CC) $(CFLAGS) memo.c

calendar.o: calendar.c
    $(CC) $(CFLAGS) calendar.c

clean:
    rm -f *.o diary
```



# Make

## ❖ *Multiple Source Files*

### ▶ Makefile v3

```
CC = gcc
CFLAGS = -c
EXENAME = diary

$(EXENAME): main.o memo.o calendar.o
    $(CC) -o $@ $^

main.o: main.c
    $(CC) $(CFLAGS) $^

memo.o: memo.c
    $(CC) $(CFLAGS) $^

calendar.o: calendar.c
    $(CC) $(CFLAGS) $^

clean:
    rm -f *.o $(EXENAME)
```

# Make

## ❖ *Multiple Source Files*

### ▶ Makefile v4

```
CC = gcc
EXENAME = diary
SOURCES := main.c memo.c calendar.c
OBJS = ${SOURCES:.c=.o}

all: ${OBJS}
    $(CC) -o ${EXENAME} $^

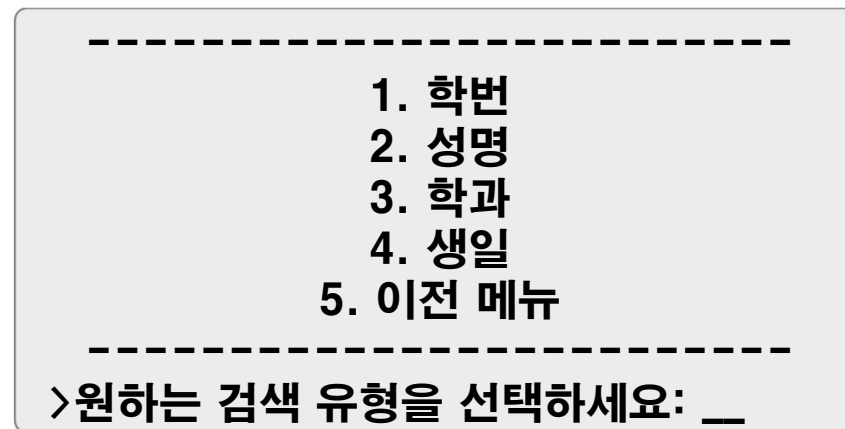
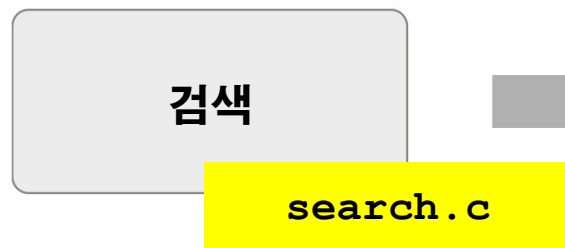
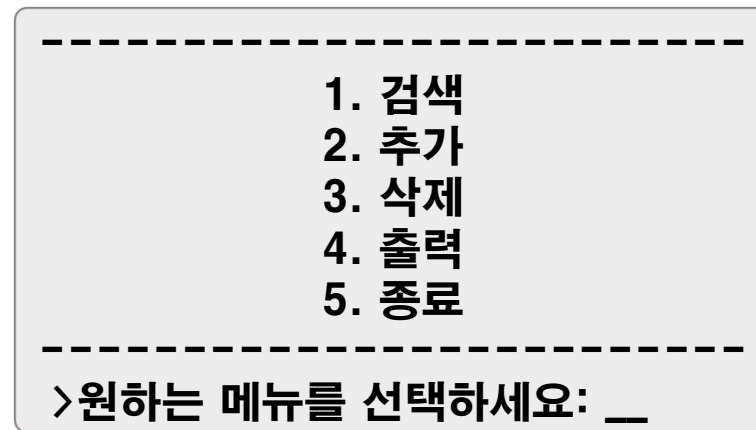
.c.o:
    $(CC) -c $^

clean:
    rm -f ${OBJS} $(EXENAME)
```

# Program Design

❖ 자료를 검색/추가/삭제/출력하기 위한 프로그램 개발

▶ ex.



# Program Design

❖ 자료 검색/추가/삭제/출력하기 위한 프로그램 개발

▶ ex. disp\_memu.c

```
int display_menu(void) {
    int sel = -1;
    printf("1. Search\n");
    printf("2. Append\n");
    printf("3. Delete\n");
    printf("4. Display\n");
    printf("5. Quit\n");

    do {
        puts("> Select menu: ");
        scanf("%d", &sel);
    } while (sel < 1 || sel > 5);

    return sel;
}
```

# Workshop #11

## ❖ 자료를 검색/추가/삭제/출력하기 위한 프로그램 개발

파일 (students.txt)에서 학생 데이터를 읽어 **리스트를 구성<sup>1)</sup>**하고, 구성된 리스트에 대해 **검색<sup>2)</sup>**, **추가<sup>3)</sup>**, **삭제<sup>4)</sup>**, **출력<sup>5)</sup>** 등을 수행하기 위한 프로그램을 개발한다.

- 전체 프로그램의 구조를 설계한다.
  - 프로그램은 필수 기능 1~5에 대해 최소 1개 이상의 함수로 구성
  - 각각의 함수는 별도의 파일로 구현 (make\_list.c, search.c, etc.)
  - main 함수는 main.c 파일로 구현
  - 컴파일을 위한 Makefile 구성
  - 프로그램 구조도 작성
- 파일 명, 파일 개수, 함수 명, 함수 형식에 대한 제약 없음
  - 각 기능적 요구사항을 만족시킬 수 있는 형태로 함수 형식 정의

ex. 출력

```
int display_students(struct student *st, int category, int order);  
/* st: students list */  
/* category: a field of student structure */  
/* order: ascend or descend */
```

# Workshop #11

## ❖ 자료를 검색/추가/삭제/출력하기 위한 프로그램 개발

### 제출 목록

- 프로그램 구조도
- Student 디렉터리 [소스 파일 (.h, .c), Makefile] 압축 파일
- 주요 기능에 대한 스크린 캡처 이미지 [images.zip]