



프로그래밍

프로그래밍과 준비

- 운영체제
 - MS-DOS, Windows 98, OS/2, UNIX, LINUX
- 컴파일러
 - Borland C++
 - cc (UNIX)
 - gcc
- 목적 파일
 - a.out
 - *.exe

프로그래밍 절차

- 프로그램 작성
 - 문서 편집기 사용
 - 파일 이름 : *filename.c*
- 컴파일
 - `cc filename.c` 또는 통합환경에서 컴파일 명령 사용
 - 컴파일 결과(목적 파일)
 - UNIX : *a.out*
 - DOS : *filename.exe*
- 실행
 - UNIX : *a.out*
 - DOS : *filename*
- * C 프로그래밍에 있어서는 UNIX나 DOS가 같기 때문에, 앞으로는 UNIX만을 기준으로 하여 설명함.

프로그래밍 예

- “from sea to shining C”를 출력하는 프로그램 작성
 1. 문자 편집기를 사용하여 다음과 같은 내용을 가지는 파일을 작성하고 파일 확장자가 .c인 파일이름을 줌 (예, sea.c)

```
#include <stdio.h>
int main(void) {
    printf("from sea to shining C\n");
    return 0;
}
```

(참고) 파일 이름은 프로그램 성격에 맞는 것으로 선택해야 함

프로그래밍 예

- “from sea to shining C”를 출력하는 프로그램 작성 (계속)

2. 앞에서 작성한 프로그램을 컴파일

```
$ cc sea.c
```

(참고) 코드에 오류가 없다면 이것의 결과로 실행 파일(목적 파일)인 a.out 이 생성됨

3. 프로그램 실행

```
$ a.out
```

```
from sea to shining C
```

← 프로그램 수행 결과

sea 프로그램 분석(1)

- `/* In file sea.c */`
 - `/* */`은 주석을 나타내며 컴파일러는 이것을 무시함
- `#include <stdio.h>`
 - `#`으로 시작하는 행은 전처리기에 의해 처리됨
 - 프로그램을 컴파일하면 컴파일 전에 전처리기가 먼저 동작 됨.
- `int main(void)`
 - 식별자 다음에 괄호 ()가 오면 그 식별자는 함수라는 것을 나타냄
 - 모든 C 프로그램은 반드시 `main()` 함수를 포함함
 - 프로그램의 수행은 항상 `main()` 함수로부터 시작함

sea 프로그램 분석(2)

- { }
- 중괄호는 여러 문장들을 그룹화하기 위해 사용됨
- 즉, 중괄호를 둘러싸인 것은 하나의 단위로 취급됨
- **"from sea to shining C\n"**
- 큰따옴표로 둘러싸인 일련의 문자들을 문자열 상수라고 함
- 문자열 상수를 이루는 단어들은 그 본래의 의미를 잃어버림
- \n은 개행 문자를 나타냄
- 프로그램 상에서 일반 문자로 표현할 수 없는 것을 표현하고 싶을 때 역슬래시 \와 결합된 문자를 사용함

sea 프로그램 분석(3)

- `printf("from sea to shining C\n");`
 - `printf` 뒤에 괄호가 있기 때문에 `printf()`는 함수임
 - 제일 뒤에 세미콜론 `;`이 있기 때문에 이것은 문장임
 - 즉, C에서 모든 문장은 세미콜론으로 끝남

printf()

- 화면에 출력하는 함수
- 연속적으로 printf()가 있을 경우, 뒤에 나오는 printf()의 출력은 바로 앞 printf()의 마지막 출력 위치에서부터 시작하여 출력한다.
- 즉, 다음 printf()는

```
printf("from sea to shi ni ng C\n");
```

다음과 같이 사용해도 같은 출력을 낸다.

```
printf("from sea to ");  
printf("shi ni ng C");  
printf("\n");
```

printf() 예제

```
#include <stdio.h>
int main(void) {
    printf("\n\n\n\n\n\n\n\n\n\n");
    printf("*****\n");
    printf(" *   from sea * \n");
    printf(" *   to shining C * \n");
    printf("*****\n");
    printf("\n\n\n\n\n\n\n\n\n\n");
    return 0;
}
```