

# 과제#12.1. 고급 프로그래머 되기

202114226 송재현

## 1. scanf("%s", str); 와 gets(str)의 차이를 조사하시오.

`scanf("%s", str);`

- 공백, 탭, 또는 새줄 문자를 만나면 입력을 중단함. 즉, 단어 하나만 읽어들이는.
- 버퍼 오버플로우를 방지하기 위한 길이 제한을 지정하지 않으면, 사용자가 입력한 길이에 따라 버퍼 오버플로우가 발생할 수 있다.
- 예: `scanf("%s", str);` 에서 `str`에 "Hello World"를 입력하면 "Hello"만 읽어들이.

`gets(str);`

- 한 줄 전체를 읽어들이. 공백도 포함된다.
- 버퍼 오버플로우에 취약하다. 사용자가 입력하는 길이에 대한 제한이 없기 때문에, 입력 길이가 `str` 배열의 크기를 초과하면 버퍼 오버플로우가 발생할 수 있다.
- C11 표준 이후로 `gets`는 위험한 함수로 간주되어 더 이상 사용되지 않는다.
- 예: `gets(str);` 에서 `str`에 "Hello World"를 입력하면 "Hello World" 전체를 읽어들이.

## 2. fflush()가 무슨 기능을 하는지 조사하시오.

- `fflush()`는 표준 라이브러리 함수로, 파일 스트림의 출력 버퍼를 비우는 데 사용된다.
- 출력 스트림(`stdout`, `stderr`)의 경우: `fflush(stdout);`를 호출하면 출력 버퍼에 남아 있는 데이터를 실제 출력 장치로 내보낸다.
- 입력 스트림(`stdin`)의 경우: 정의되지 않은 동작이 발생할 수 있다. 그러나 특정 시스템에서는 `fflush(stdin);`를 사용해 입력 버퍼를 비울 수 있다. 일반적으로 입력 버퍼를 비우는 다른 방법을 사용하는 것이 좋다.

### 3. scanf와 scanf\_s의 차이점 조사

#### scanf

- 형식 지정자에 따라 표준 입력으로부터 데이터를 읽어온다.
- 입력 데이터에 대해 길이 제한을 지정하지 않으면 버퍼 오버플로우가 발생할 수 있다.
- 예: scanf("%s", str);는 길이 제한 없이 입력을 받아 버퍼 오버플로우의 위험이 있다.

#### scanf\_s

- Microsoft에서 만든 보안 강화 버전의 scanf이다.
- 입력 버퍼의 크기를 명시적으로 지정해야 하며, 이는 버퍼 오버플로우를 방지 하는데 도움이 된다.
- 예: scanf\_s("%s", str, size);는 size를 사용하여 str의 최대 크기를 지정함으로써 안전하게 입력을 받을 수 있다.