공유 라이브러리 개요를 이해한다.
공유 라이브러리 호출 방법과 생성 방법을 익힌다.

CHAPTER 8. 공유 라이브러리

1. 공유 라이브러리 개요

공유 라이브러리란 무엇인가?

프로그래밍 방법

1. 함수를 만들고, 함수를 Copy and Paste한다.
   → SubVI
2. 공통적으로 사용하는 함수의 모음을 따로 관리해서 공유한다.
   → 공유 라이브러리

SubVI 사용의 단점

- 함수 사용 중에 수정될 수 있다. → 관리 및 디버깅이 어렵다.
- 한 개 프로그램에 동일한 함수가 여러 번 복사되어 사용된다.
- 함수에 문제가 있을 경우, 모든 응용 프로그램을 수정해서 다시 배포해야 한다.
- 다른 프로그래밍 언어들과 공유가 힘들다.

공유 라이브러리 개요

- 공유 라이브러리는 어플리케이션이나 다른 공유 라이브러리가 호출할 수 있는 실행 코드 또는 데이터를 포함하는 소프트웨어 모듈
- 공유 라이브러리의 함수와 데이터는 실행시 로드되고 연결됨
- 다양한 언어에서 공유 라이브러리를 작성할 수 있음

공유 라이브러리 개요(계속)

- 공유 라이브러리는 표준 인터페이스를 사용하여 함수와 데이터를 공개
- 많은 공유 라이브러리 타입 정의는 C 프로그래밍 언어에서의 함수 정의와 비슷함
- 공유 라이브러리를 사용하는 플랫폼에 따라 공유 라이브러리의 이름이 달라짐
  - Windows = DLLs
  - MacOS = Frameworks
  - Unix = Shared Library

공유 라이브러리의 장점

- 많은 경우, 공유 라이브러리를 변경해도 호출 VI를 변경할 필요가 없음
- 공유 라이브러리를 사용하면 다른 언어로 된 기존 코드를 호출할 수 있음
- 오늘날 대부분의 개발 환경은 DLL 생성을 지원

2. 공유 라이브러리 호출

Windows DLL

Mac OS Framework

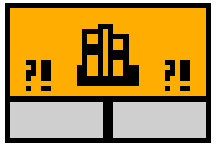
Linux Shared Library

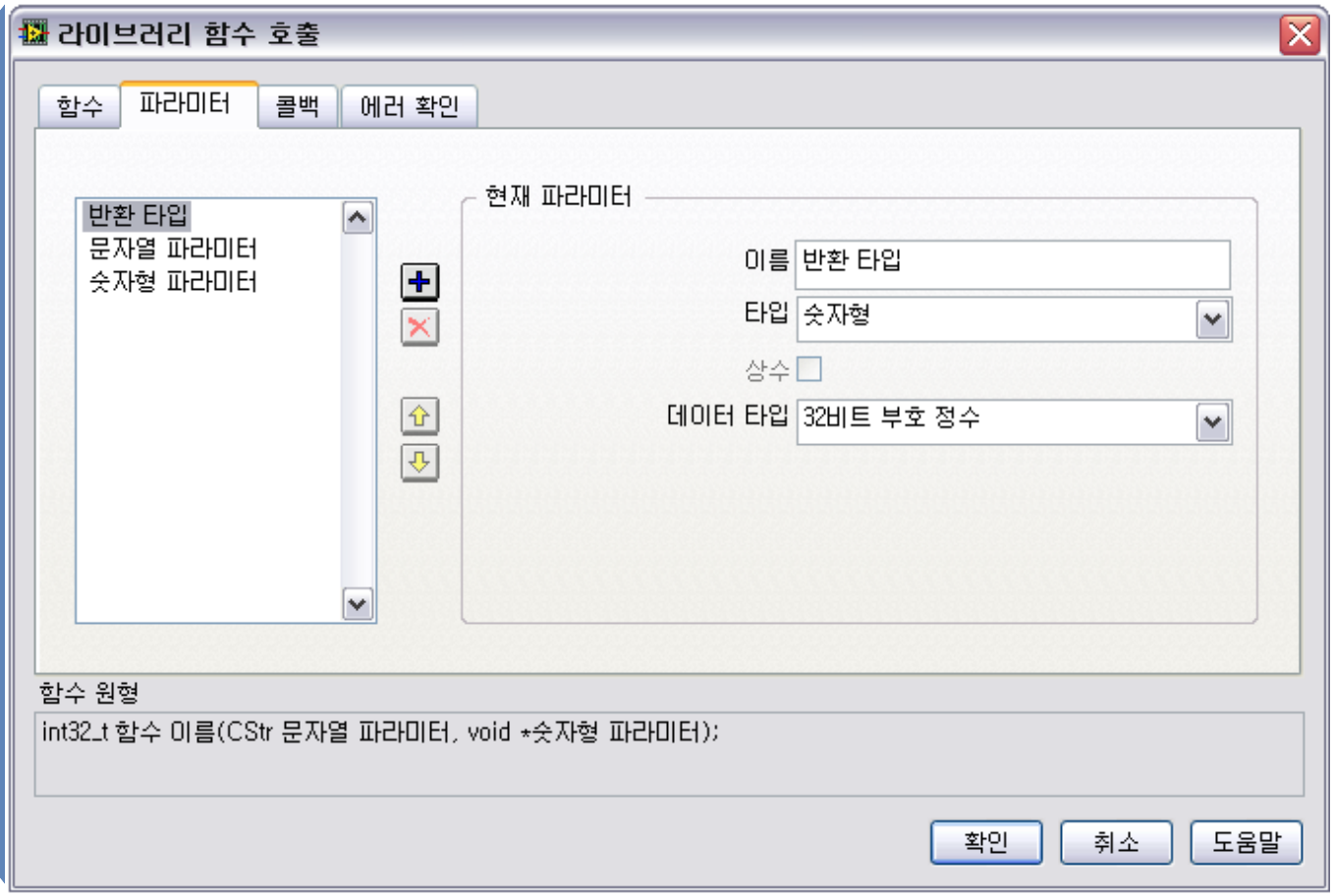
공유 라이브러리 호출하기

LabVIEW에서 공유 라이브러리를 호출하는 두 가지 방법:

- 라이브러리 함수 호출 노드를 사용하여 함수를 직접 설정
- 공유 라이브러리 반입 마법사를 사용하여 LabVIEW가 코드를 생성하도록 함

라이브러리 함수 호출 노드



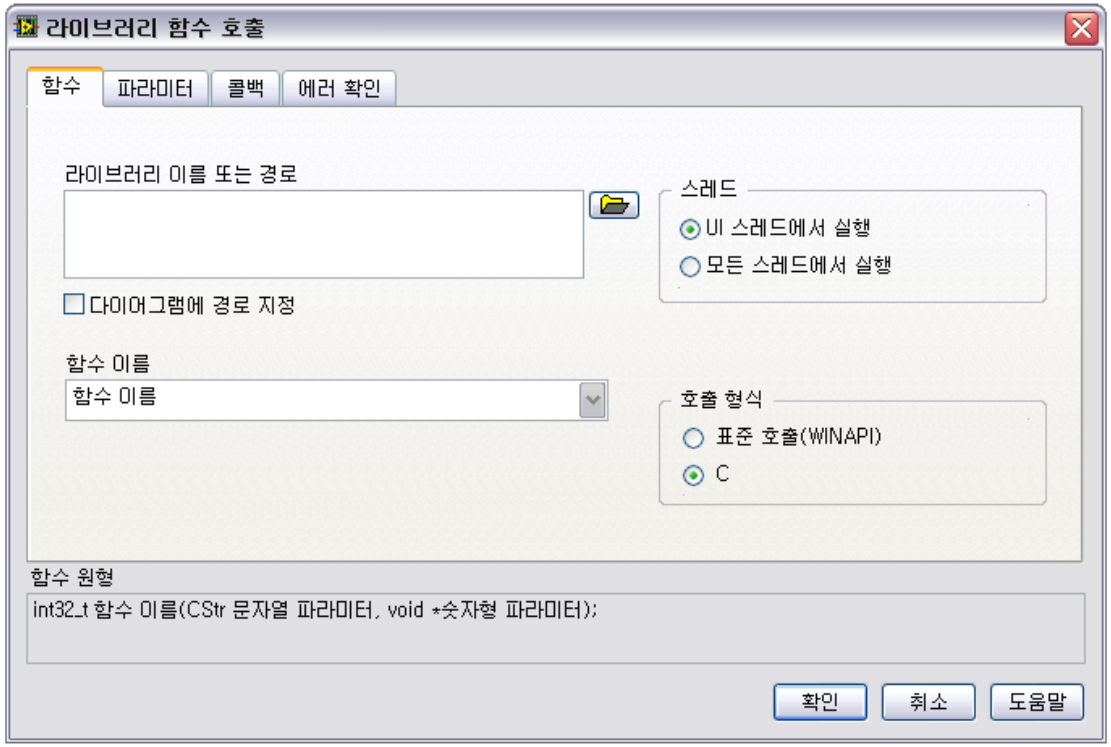


라이브러리 함수 호출 설정하기

- 호출 형식 설정
- 스레드 설정
- 파라미터 설정
- 반환 타입
- 파라미터 편집
- 콜백

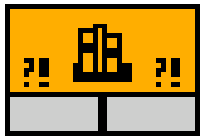
스레드 안전한(Thread-Safe) DLL 및 스레드 불안전한(Thread-Unsafe) DLL

- 기본으로 모든 호출 라이브러리 객체는 UI 스레드에서 실행
- **스레드** 옵션을 **모든 스레드에서 실행**으로 변경하면,
  - 라이브러리를 동시에 여러번 호출할 수 있음
  - 스레드 라이브러리가 스레드 안전한 DLL이어야 함

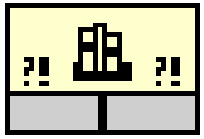


스레드 안전한 DLL 특성

- 보호되지 않는 글로벌 데이터는 포함하지 않음
- 어떠한 하드웨어에도 접근하지 않음
- 스레드 안전하지 않은 함수를 호출하지 않음



UI 스레드에서 실행



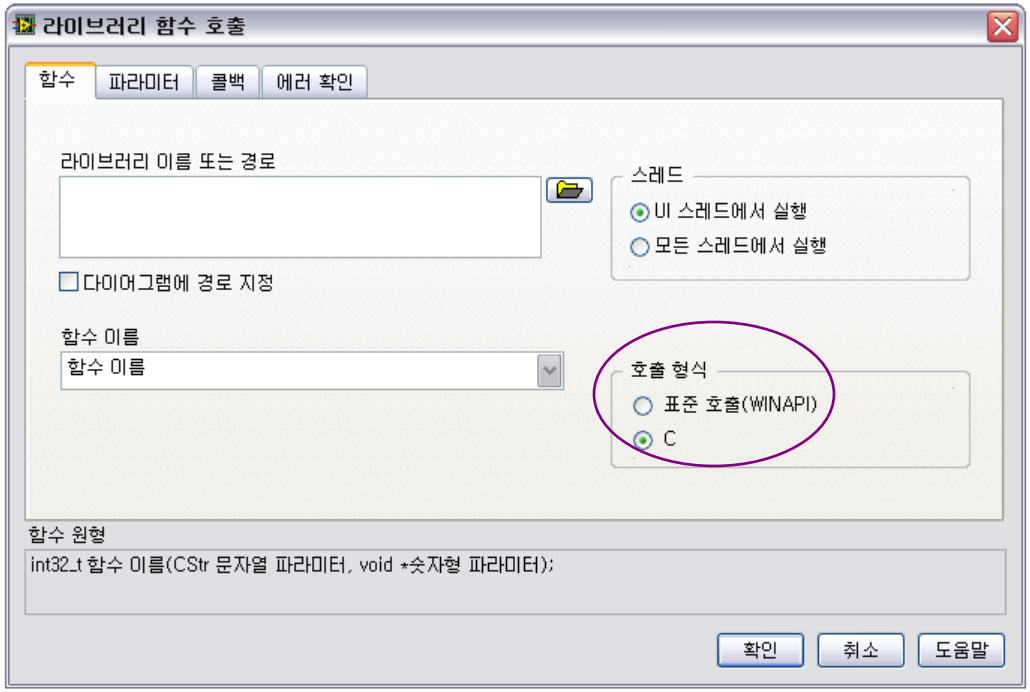
모든 스레드에서 실행

호출 형식

호출 규약은 파라미터가 공유 라이브러리로/에서 전달되는 방식을 정의

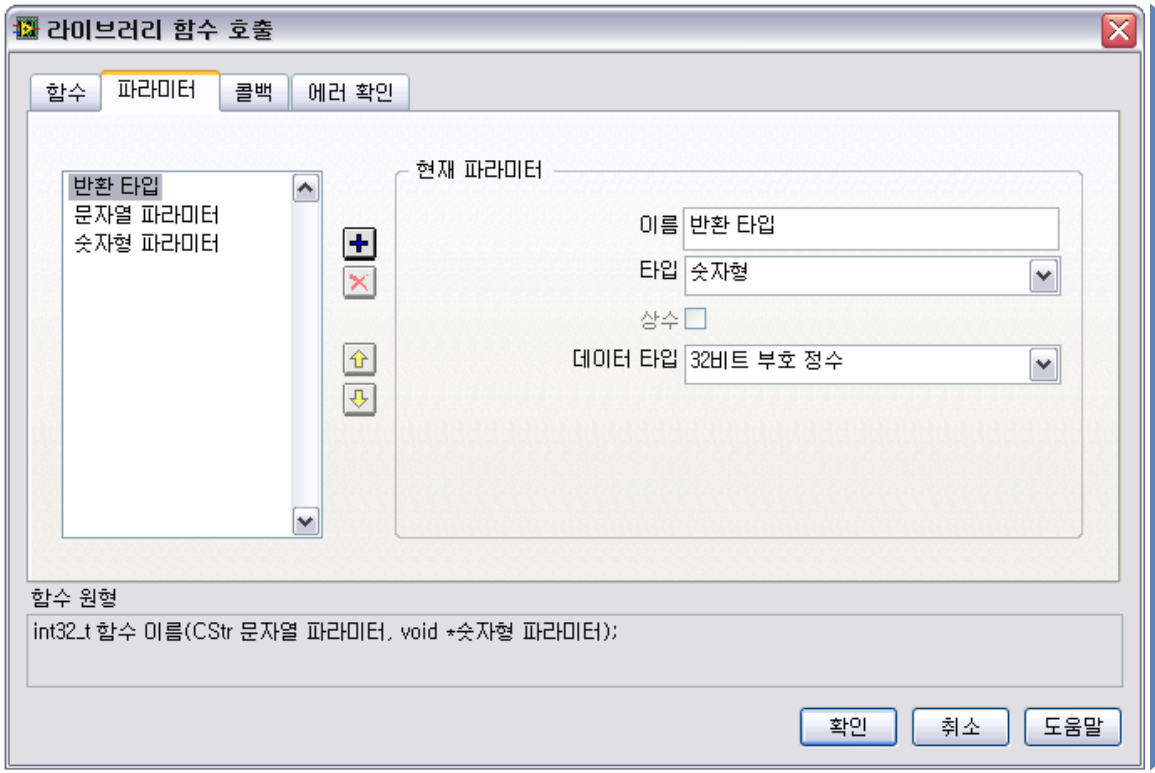
- Stdcall (WINAPI)
- C

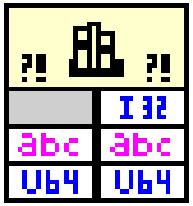
공유 라이브러리 문서는 일반적으로 호출 형식을 지정함



파라미터

각 파라미터는 라이브러리 함수 호출 노드의 터미널 쌍에 대응





파라미터 타입

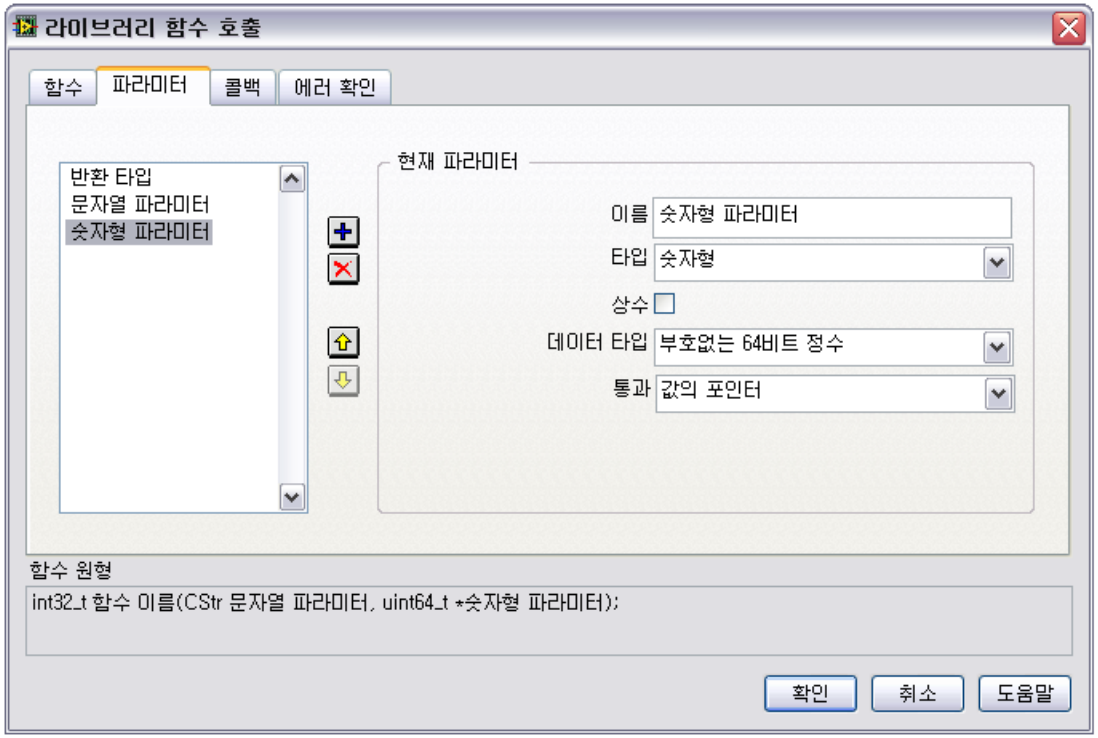
각 파라미터에 대한 데이터 타입 선택

일반적인 파라미터 타입

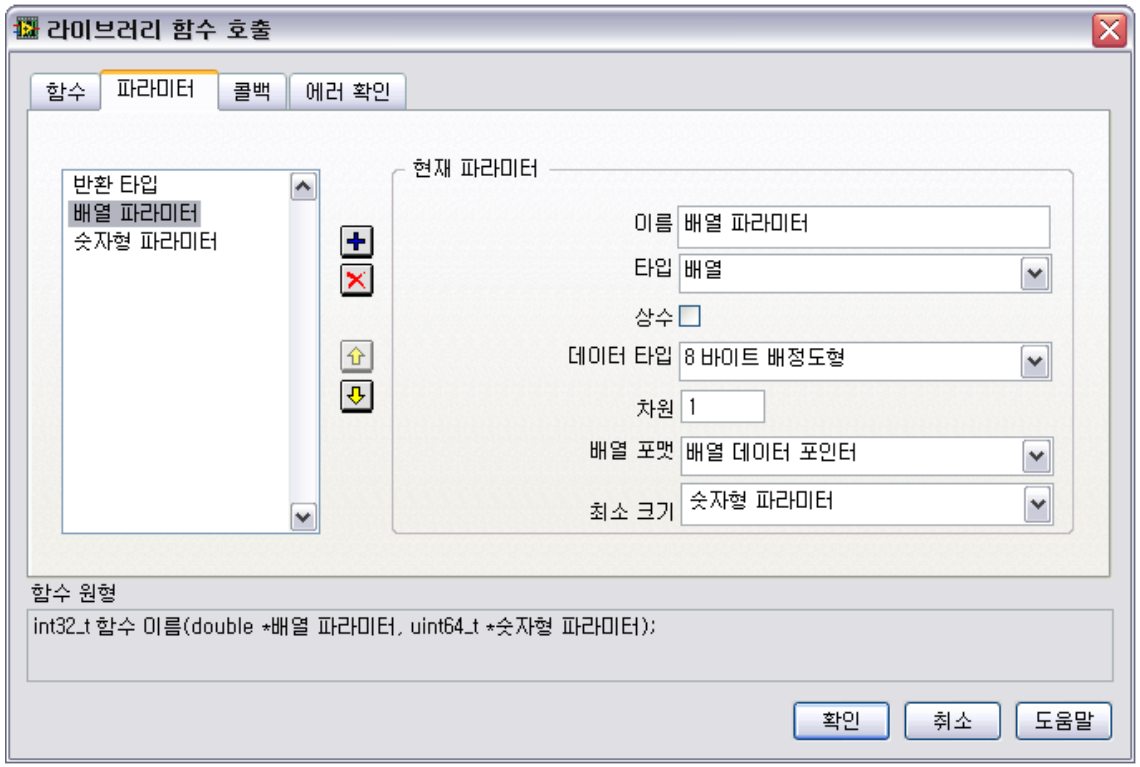
- 숫자형
- 배열
- 문자열
- 타입에 적용

숫자형 파라미터 타입

- 숫자형을 지정
- 값으로 전달할지 또는 참조로 전달할지를 결정

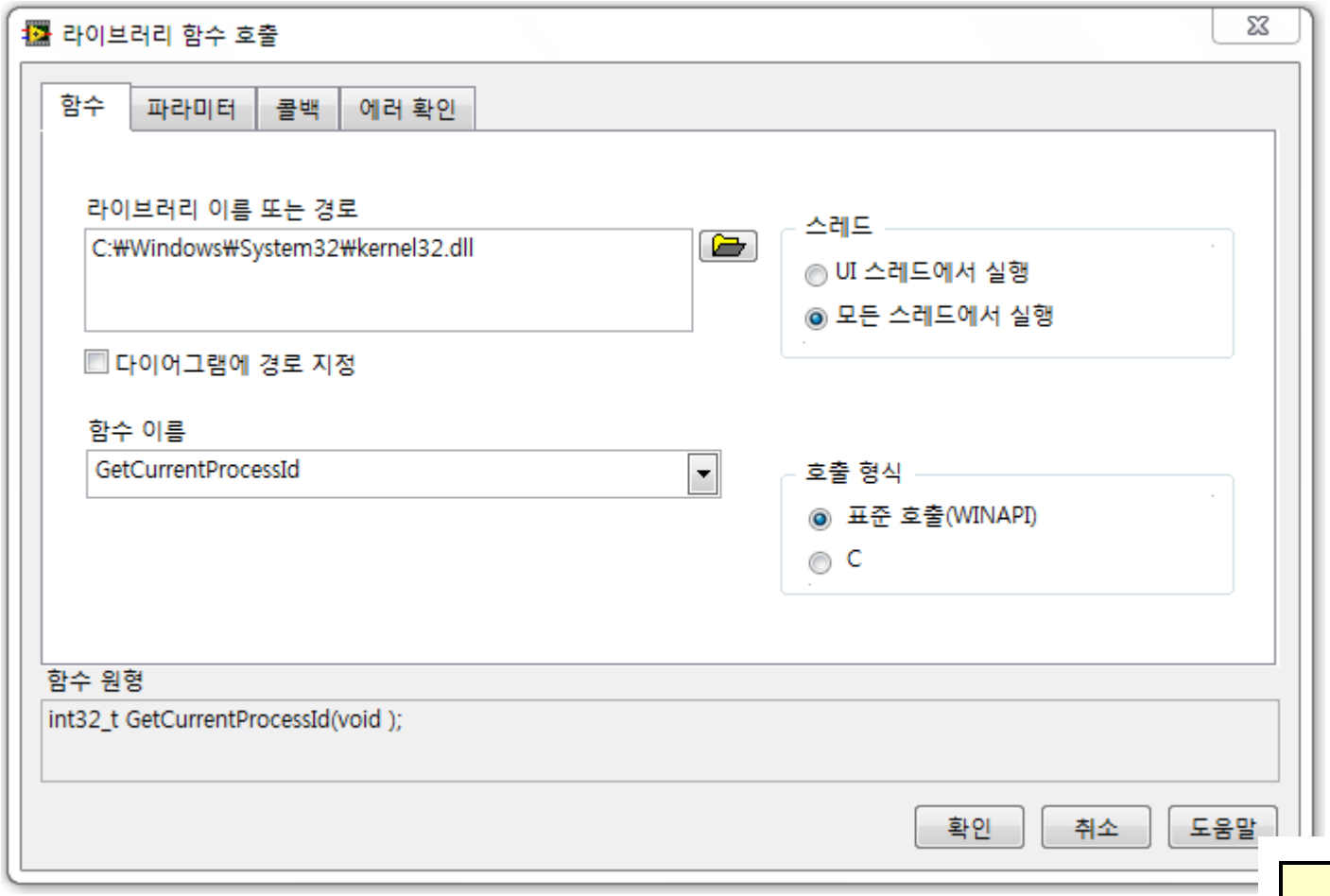


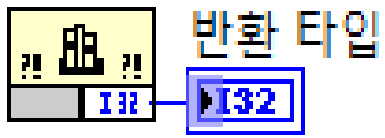
배열 파라미터 타입



- 배열의 데이터 타입 선택
- 배열의 차원 수 선택
- 배열의 전달 방식 (**배열 포맷**)을 결정
- **배열 포맷**이 배열 데이터 포인터인 경우, 배열 크기를 지정하는 다른 파라미터를 생성

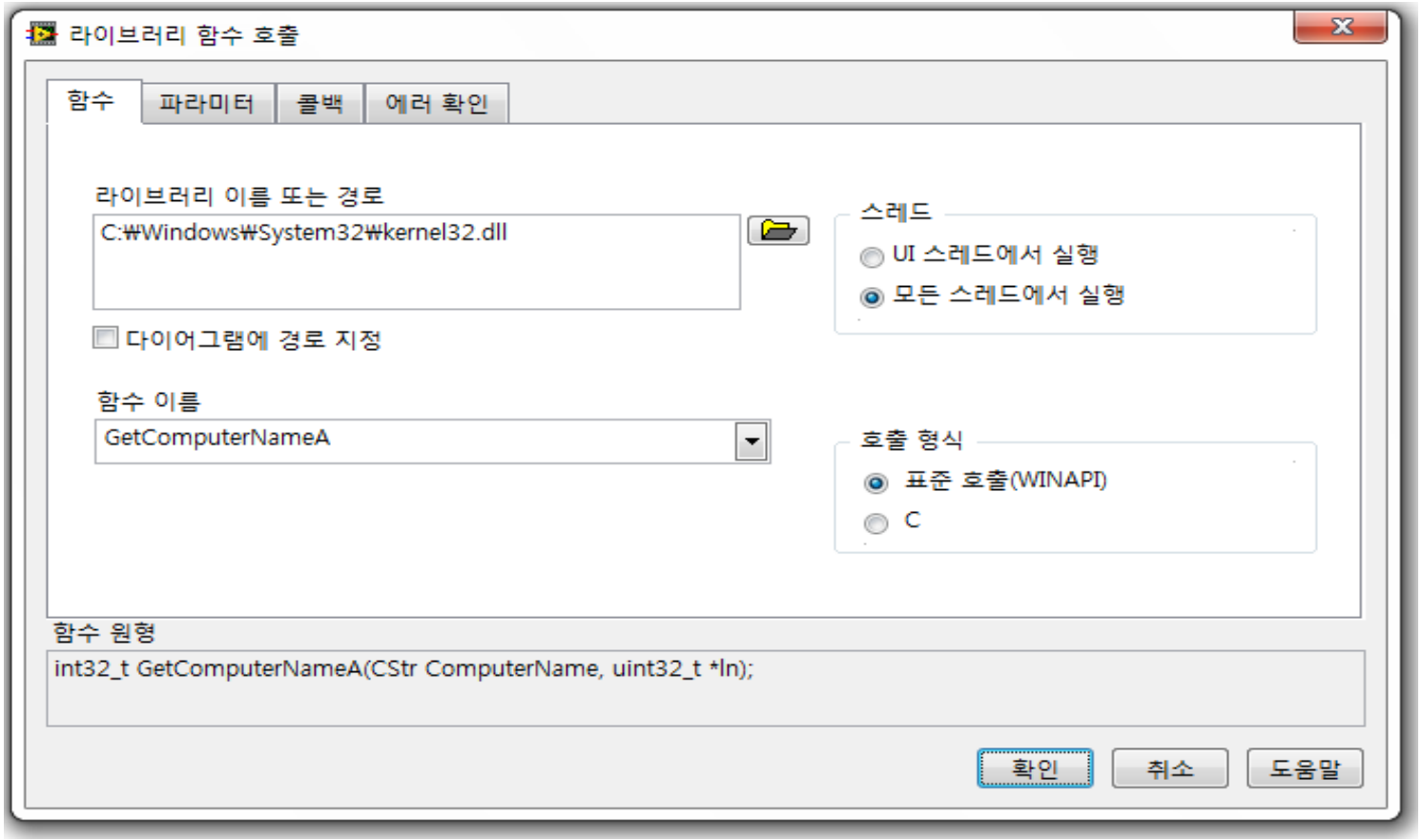
실습 8-1. 컴퓨터 프로세서 ID 읽어오기



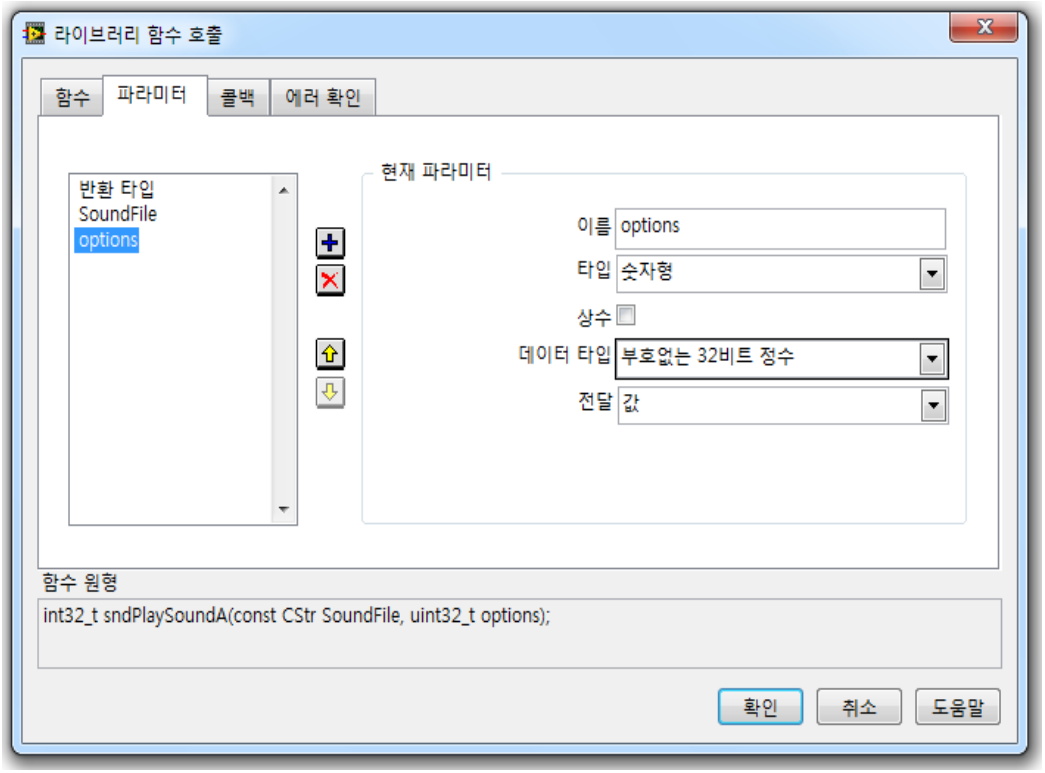


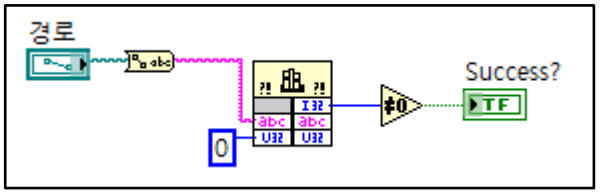
실습 8-2. 컴퓨터 이름 읽어오기

- Windows System32 >> GetComputerNameA



실습 8-3.
Windows Media Player 실행하기





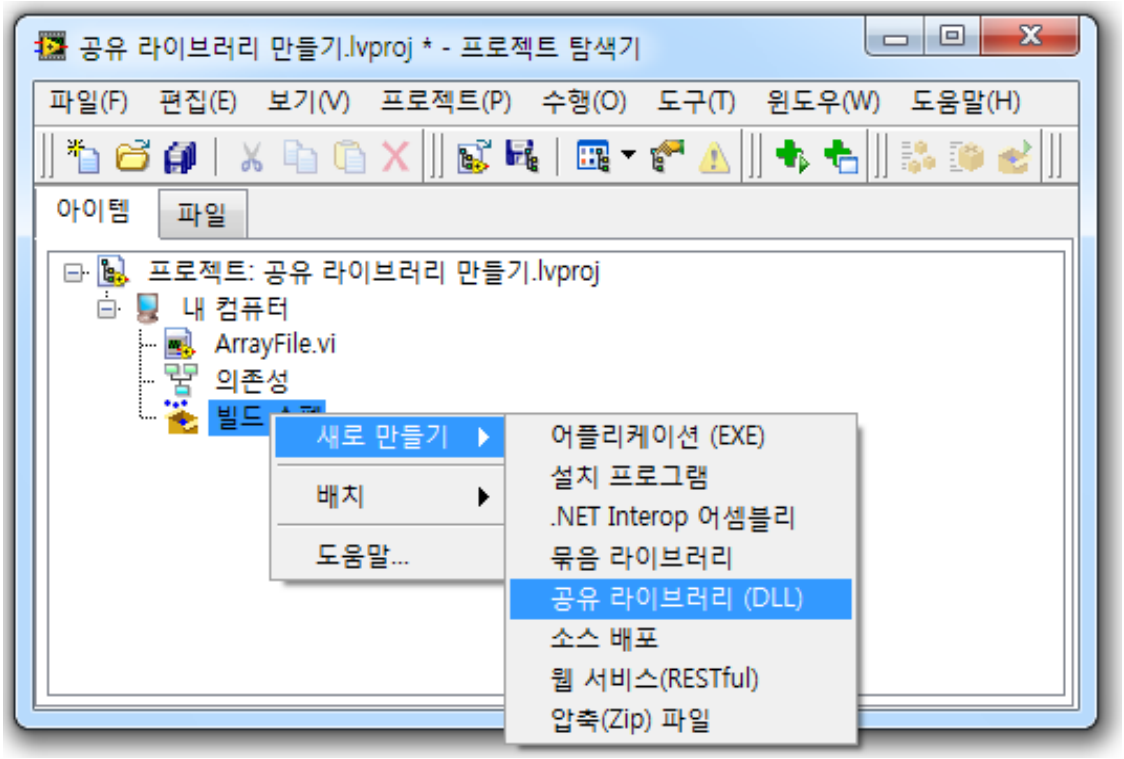
int32_t sndPlaySoundA(const CStr SoundFile, uint32_t options)

3. 공유 라이브러리 만들기

어플리케이션 빌드

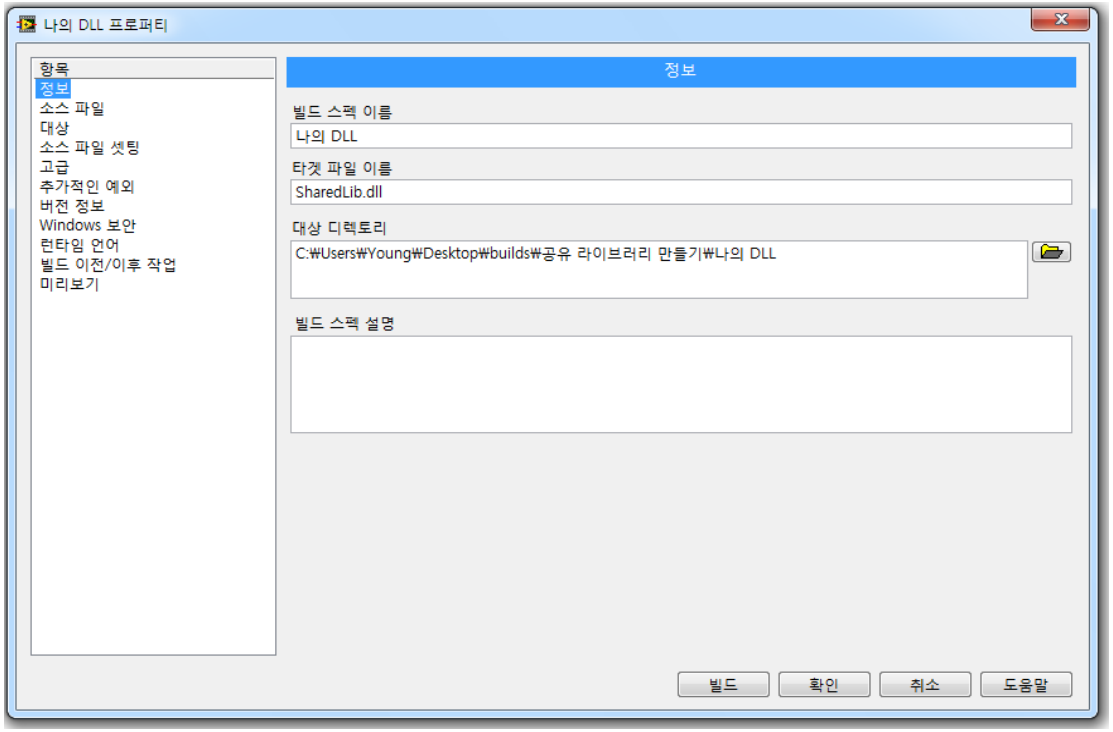
시작하기

- VI를 준비한다.
- 빌드 스펙 >> 공유 라이브러리 (DLL)



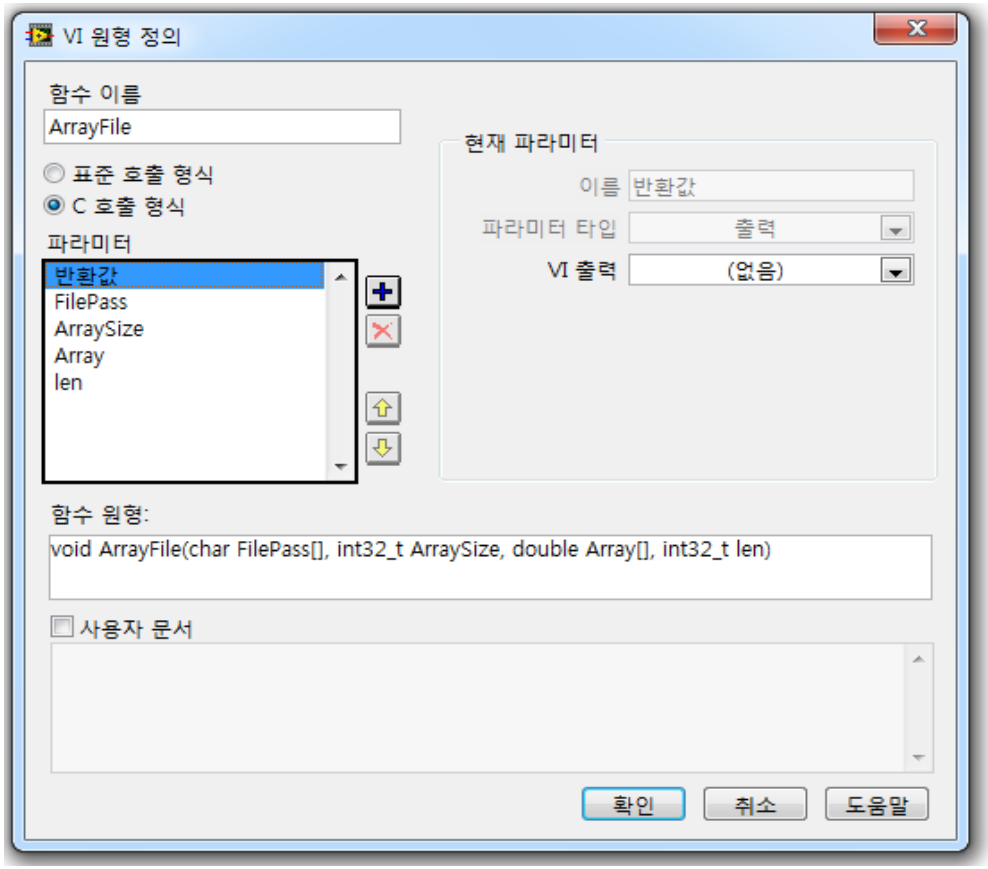
빌드 스펙 설정하기

- 정보
- 소스 파일

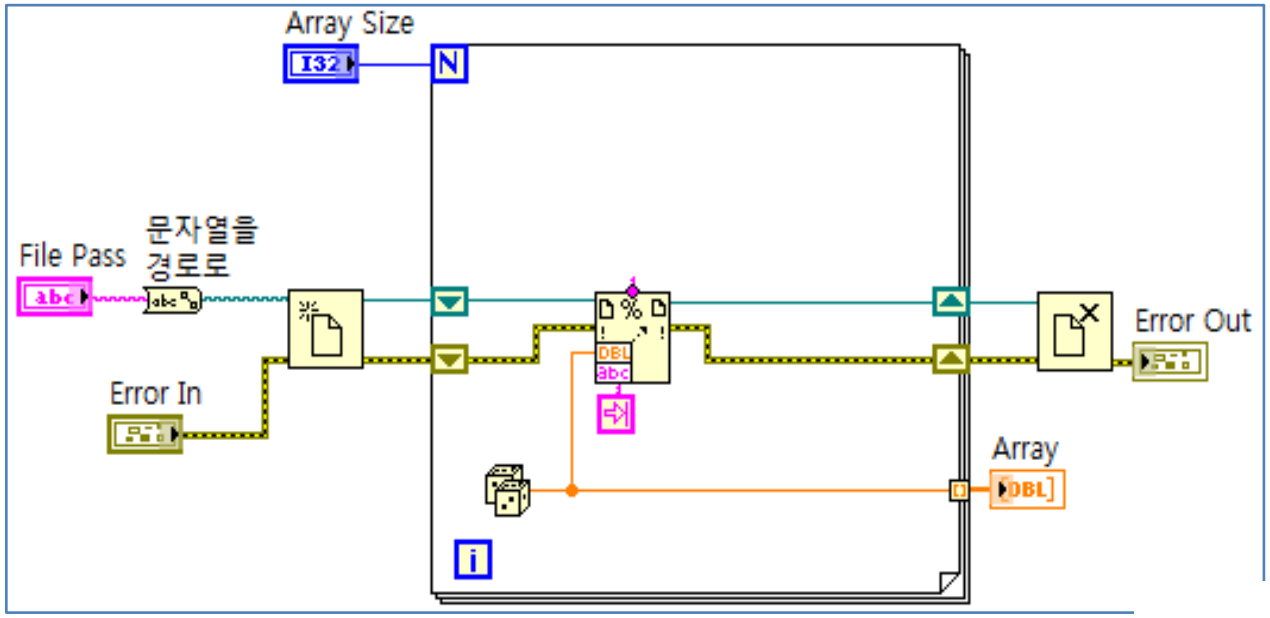


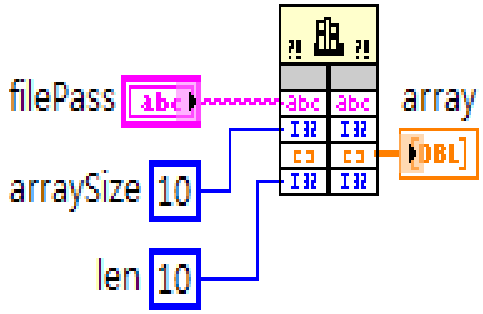
VI 원형 정의

- VI 반출의 파라미터를 정의한다.
  - 호출 형식
  - 파라미터 정의
  - 함수 원형 확인
  - 사용자 문서



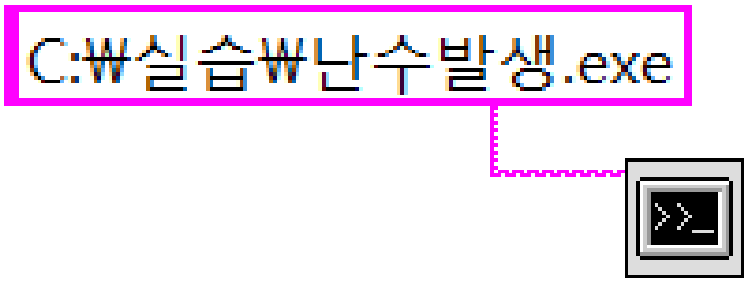
실습 8-4.
공유 라이브러리 만들기 및 사용하기

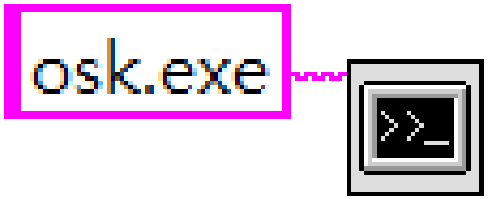




시스템 실행

- 시스템 실행 함수
  - 문자열 경로
  - C:\Windows\system32 폴더





실습 8-5. 화상키보드 실행