

1. 다음 중 폰 노이만 구조의 핵심 구성 요소가 아닌 것은 무엇인가요?

- ① 중앙처리장치(CPU)
- ② 메모리
- ③ 프로그램
- ④ 하드디스크

4

2. 다음은 컴퓨터 부팅 과정에 대한 설명입니다. 올바른 순서로 나열된 것을 고르세요.

- ① 운영체제 커널 로드
- ② POST (Power-On Self-Test) 실행
- ③ 부트 로더 (Boot Loader) 실행
- ④ 전원 인가
- ⑤ BIOS 설정 로드
- ⑥ 초기 프로세스 실행
- ⑦ 사용자 인터페이스 실행

① ④ → ② → ③ → ⑤ → ① → ⑥ → ⑦

② ④ → ② → ⑤ → ③ → ① → ⑥ → ⑦

③ ④ → ⑤ → ② → ③ → ① → ⑥ → ⑦

④ ② → ④ → ⑤ → ③ → ① → ⑥ → ⑦

⑤ ④ → ② → ⑤ → ① → ③ → ⑥ → ⑦

④ → ② → ⑤ → ③ → ① → ⑥ → ⑦

3. 다음은 컴퓨터 구조와 운영체제에 대한 설명입니다. 이 중 옳지 않은 것을 고르세요.

- ① CPU는 메모리에서 명령어를 읽어 해석하고 실행하는 컴퓨터의 핵심 부품으로, '컴퓨터의 뇌'라고 불린다.
- ② 프로그램은 보조 기억장치에 저장된 정적 파일이며, 실행 시 프로세스가 되어 동적으로 작동한다.
- ③ 스레드는 프로세스 내의 실행 단위로, 다른 프로세스의 메모리 공간을 자유롭게 접근할 수 있다.
- ④ 멀티스레딩은 하나의 프로세스에서 여러 스레드가 동시 작업을 수행하는 방식으로, 웹서버 등 고성능 애플리케이션에 활용된다.
- ⑤ 리눅스는 오픈 소스 운영체제로, 다중 사용자 및 다중 작업을 지원하며 서버 환경에 최적화되어 있다.

정답: ③

스레드는 자신이 속한 프로세스의 메모리만 공유

4. 다음은 리눅스 명령어와 그 용도를 설명한 표입니다. 이 중 잘못 연결된 것을 고르세요.

- ① pwd - 현재 작업 중인 디렉토리의 절대 경로를 출력
- ② ls -a - 숨김 파일을 포함한 디렉토리 내용을 상세 정보와 함께 출력
- ③ cd / - 시스템의 최상위 디렉토리(루트)로 이동
- ④ rmdir [이름] - [이름] 디렉토리와 파일 삭제
- ⑤ /bin - 시스템 관리자용 명령어가 저장된 디렉토리

4

rmdir은 비어 있는 디렉토리만 삭제 가능

5. 다음은 네트워크와 통신에 관한 설명입니다. 옳지 않은 것을 고르세요.

- ① API는 클라이언트와 프로그램 사이를 연결하는 인터페이스로, 정해진 방식으로 요청하면 기능이나 데이터를 제공한다.
- ② 라이브러리는 독립성을 가지며, 다른 라이브러리의 기능에 의존하지 않는다.
- ③ 프레임워크는 개발자가 능동적으로 제어하며, 프레임워크는 수동적으로 따라간다.
- ④ HTTPS는 HTTP의 보안 강화 버전으로, SSL/TLS를 통해 데이터를 암호화하여 통신한다.
- ⑤ IPv6는 128비트 주소 체계를 사용하며, IPv4에 비해 거의 무한한 주소 공간을 제공한다.

3

6. 다음은 로봇 공학과 센서 기술에 관한 설명입니다. 옳지 않은 것을 고르세요.

- ① PWM 신호는 아날로그 신호처럼 동작하는 효과를 디지털 방식으로 구현한 것이다.
- ② 임베디드 시스템은 고수준 언어(예: Python)로만 개발해야 하며, 메모리 조작이 불가능하다.
- ③ 로봇의 구면 관절은 3축 회전이 가능하여 구형 공간에서 자유로운 운동이 가능하다.
- ④ IoT 기술은 디바이스, 클라우드, 디바이스 간 통신을 용이하게 하는 연결 네트워크를 의미한다.
- ⑤ 직교 좌표계(Cartesian)는 x, y, z축을 기반으로 한 가장 기본적인 좌표계이다.

2

임베디드 시스템은 저수준 언어(C/C++)로 개발되며, 메모리 조작이 핵심

7. 다음 중 ROS2의 핵심 통신 방식에 대한 설명으로 옳지 않은 것을 고르세요.

- ① 토픽(Topic)은 비동기식 단방향 통신으로, 퍼블리셔(Publisher)와 서브스크라이버(Subscriber)로 구성된다.
- ② 서비스(Service)는 동기식 양방향 통신으로, 클라이언트가 요청을 보내면 서버가 응답한다.
- ③ 액션(Action)은 비동기식 단방향 통신만 지원하며, 장기 실행 작업에 적합하다.
- ④ 파라미터(Parameter)는 노드의 설정 값을 동적으로 변경할 수 있도록 서비스 기반으로 동작한다.
- ⑤ DDS는 ROS2의 기본 미들웨어로, UDP 기반 멀티캐스트를 통해 효율적인 통신을 제공한다.

3

액션(Action)은 비동기식 + 동기식 양방향 통신을 지원

8. 다음 중 ROS2 명령어와 그 기능에 대한 설명으로 옳지 않은 것을 고르세요.

- ① `ros2 node list` - 현재 실행 중인 모든 노드의 이름을 표시합니다.
- ② `ros2 run turtlesim turtlesim_node` - turtlesim 패키지의 시뮬레이터 노드를 실행합니다.
- ③ `ros2 topic info /cmd_vel` - 지정된 토픽의 발행자(Publishers)와 구독자(Subscribers) 정보를 표시합니다.
- ④ `ros2 service call /spawn turtlesim/srv/Spawn` - turtlesim에서 새로운 거북이를 생성하는 서비스를 호출합니다.
- ⑤ `rqt_graph` - 패키지의 빌드 환경을 설정하는 CMake 파일을 생성합니다.

5

`rqt_graph`는 노드와 토픽의 연결 관계를 시각화하는 GUI 도구

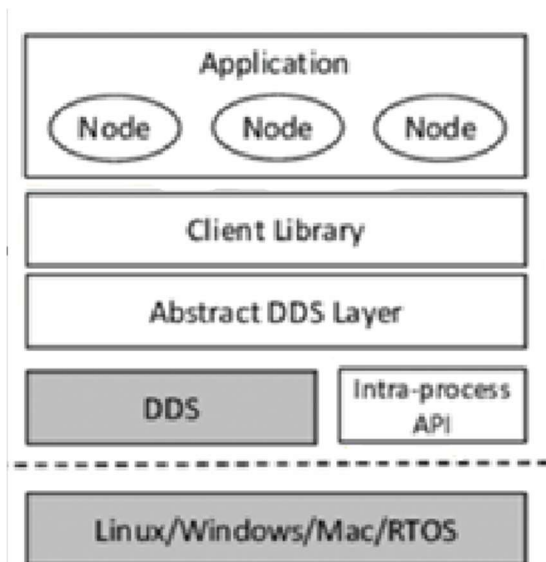
9. 다음 중 ROS1과 ROS2의 차이점에 대한 설명으로 옳지 않은 것을 고르세요.

- ① 운영체제 지원: ROS1은 Linux만 지원하지만, ROS2는 Linux, Windows, macOS, RTOS 등을 지원한다.
- ② 미들웨어 계층: ROS1은 자체 프로토콜(TCPROS/UDPROS)을 사용하며, ROS2는 DDS 기반으로 통신한다.
- ③ 마스터 노드: ROS1은 중앙 집중형 마스터 노드가 필수적이지만, ROS2는 분산 discovery로 마스터가 불필요하다.
- ④ 실시간성: ROS1은 실시간성을 보장하지만, ROS2는 DDS 사용으로 인해 실시간성이 저하되었다.
- ⑤ 클라이언트 라이브러리: ROS1은 roscpp/rospy만 제공하지만, ROS2는 rclcpp/rclpy 등 멀티 언어를 지원한다.

4

ROS2는 DDS의 QoS 설정을 통해 실시간성과 신뢰성을 향상시켰으며, ROS1보다 우수한 실시간 성능을 제공합니다.

10. 어떤 시스템 구조를 나타낸 다이어그램인가?



Ros2

11. 다음은 ROS2 인터페이스 패키지에 대한 설명입니다. 이 중 옳지 않은 것을 고르세요.

- ① .msg 파일은 토픽 통신에서 사용되는 단방향 데이터 형식을 정의하며, 기본값 설정이 가능하다.
- ② .srv 파일은 요청(Request)과 응답(Response) 섹션으로 구분되며, 상수값을 정의할 수 있다.
- ③ .action 파일은 Goal, Result, Feedback 3개 섹션으로 구성되며, 장시간 작업에 사용된다.
- ④ 인터페이스 패키지 생성 시 Python 패키지는 ament_python 빌드 타입을, C++ 패키지는 ament_cmake 빌드 타입을 사용한다.
- ⑤ 패키지 빌드 전 package.xml과 CMakeLists.txt 파일에 msg/srv/action 파일을 명시적으로 등록해야 한다.

4

인터페이스 패키지는 언어와 무관하게 ament_cmake 빌드 타입을 사용

12. 다음은 ROS2 Executor의 동작 과정에 대한 설명입니다. 이 과정을 순서대로 나열한 것으로 올바른 것은 무엇인가요?

- ① 이벤트 감지 → 메시지 가져오기(take) → 콜백 실행(execute) → 대기(wait)
- ② 대기(wait) → 이벤트 감지 → 메시지 가져오기(take) → 콜백 실행(execute)
- ③ 이벤트 감지 → 대기(wait) → 메시지 가져오기(take) → 콜백 실행(execute)
- ④ 대기(wait) → 메시지 가져오기(take) → 콜백 실행(execute) → 이벤트 감지

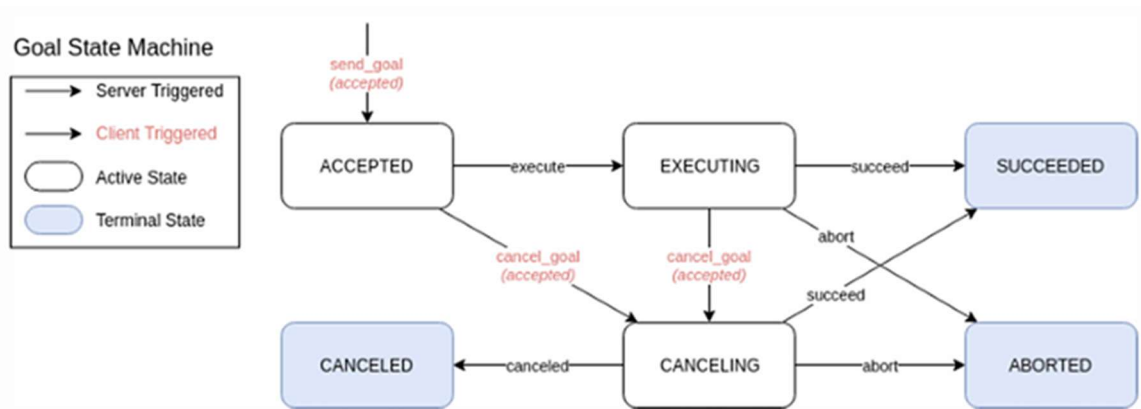
2

13. 다음은 ROS Action의 동작 단계를 설명한 것입니다. 올바른 실행 순서로 나열된 것은 무엇인가요?

- Ⓐ Goal 수락: 서버가 요청 수락/거절
- Ⓑ 취소 처리: 클라이언트 요청 시 서버가 Goal 취소
- Ⓒ Goal 전송: 클라이언트가 서버에 작업 요청
- Ⓓ 결과 전송: 작업 완료 시 결과(성공/실패/취소) 전송
- Ⓔ 실행 및 피드백: 서버가 작업 실행하며 피드백 전송

- ① Ⓒ → Ⓐ → Ⓔ → Ⓑ → Ⓓ
- ② Ⓐ → Ⓒ → Ⓔ → Ⓓ → Ⓑ
- ③ Ⓒ → Ⓔ → Ⓐ → Ⓑ → Ⓓ
- ④ Ⓑ → Ⓒ → Ⓐ → Ⓔ → Ⓓ

1



14. 다음은 ROS2 Action 서버의 상태(state)와 해당 상태의 설명을 짝지은 것입니다. 이 중 옳게 짝지어지지 않은 것을 고르세요.

- ① ACCEPTED - 목표가 수락되었으며 실행을 대기 중인 상태
- ② EXECUTING - 목표가 현재 액션 서버에서 실행 중인 상태
- ③ CANCELING - 액션 서버가 내부 오류로 인해 목표 처리를 중단한 상태
- ④ SUCCEEDED - 액션 서버가 목표를 성공적으로 달성한 상태
- ⑤ ABORTED - 외부 요청 없이 작업 서버에 의해 목표가 비정상 종료된 상태

3

CANCELING – 클라이언트가 목표 취소를 요청했고 액션 서버가 취소 요청을 수락

ABORT

15. 다음 ROS2 Action 명령어 코드와 설명이다. 잘못된 것을 모두 고르세요.

- ① ros2 action list - 액션 서버 / 클라이언트 목록
- ② ros2 action info - 액션 타입 조회
- ③ ros2 action show – 동작 중인 액션의 상태 조회
- ④ ros2 action execute_goal - 액션 실행 및 피드백 / 결과 확인
- ⑤ ros2 action status - Goal 상태 모니터링
- ⑥ ros2 action feedback - 실시간 피드백 출력

3- 액션 메시지 구조 확인

4 – send_goal

16. 실습 중 계산기 프로그램 ROS2 노드 통신 구조 설명 중 내용이 틀린 것을 모두 고르세요.

- ① Node (operator)는 Service Client로 동작하며, Service Request를 보내고 Service Response를 받습니다.
- ② Node (calculator)의 Action Server는 Action Goal을 Topic으로 수신합니다.
- ③ Action (arithmetic_checker)에서 Feedback은 Topic으로, Result는 Service로 전송됩니다.
- ④ Node (argument)는 Parameter Server와 Topic Publisher 기능을 동시에 수행합니다.

2

17. 문제: 다음 ROS2 명령어와 설명 중 틀린 것을 모두 고르세요. (복수 정답 가능)

- ① `ros2 run <패키지명> <노드명>` - 특정 패키지에 포함된 노드를 실행하는 명령어
- ② `ros2 launch <패키지명> <런치파일명>` - 여러 노드와 설정을 한번에 실행하기 위한 런치 파일 실행
- ③ `ros2 topic list` - 현재 시스템에서 활성화된 모든 토픽의 목록을 표시
- ④ `ros2 topic echo <서비스명>` - 특정 서비스의 요청과 응답을 실시간으로 모니터링
- ⑤ `ros2 service call <토픽명> <메시지타입> "{파라미터:값}"` - 특정 토픽에 메시지 발행

4,5

4 - 토픽 데이터 출력

5 - 서비스 요청 전달

18. 다음 2D 회전 변환 행렬에서 빈칸을 올바르게 채운 것은?

[A B]

[C D]

- ① $A=\cos\theta, B=-\sin\theta, C=\sin\theta, D=\cos\theta$
- ② $A=\sin\theta, B=-\cos\theta, C=\cos\theta, D=\sin\theta$
- ③ $A=\cos\theta, B=\sin\theta, C=-\sin\theta, D=\cos\theta$
- ④ $A=-\cos\theta, B=\sin\theta, C=\sin\theta, D=-\cos\theta$
- ⑤ $A=\tan\theta, B=-\cot\theta, C=\cot\theta, D=\tan\theta$

1

19. 다음은 SLAM에 대한 설명입니다. 이 중 옳지 않은 것을 고르세요.

- ① SLAM은 "동시적 위치 추적 및 지도 생성"을 의미하며, 로봇이 미지의 환경에서 자신의 위치를 추정하면서 동시에 주변 지도를 작성하는 기술이다.
- ② Localization은 주변 환경에 대한 지도를 생성하는 과정이며, Mapping은 로봇의 현재 위치를 추정하는 과정이다.
- ③ Sensor Fusion은 라이다, 카메라, IMU 등 다양한 센서 데이터를 통합해 정확도를 높이는 핵심 기술이다.
- ④ SLAM 구현을 위해 확률적 추정 방법(예: 칼만 필터, 파티클 필터)이 자주 사용된다.

2

20. 다음은 URDF(Universal Robot Description Format)와 로봇 모델링에 대한 설명입니다. 이 중 옳지 않은 것을 고르세요.

- ① URDF는 XML 기반 포맷으로, 로봇의 링크(link)와 조인트(joint)를 정의하여 물리적 구조를 기술한다.
- ② fixed 타입 조인트는 완전히 고정된 연결을 의미하며, revolute 타입은 제한된 각도 범위에서 회전이 가능하다.
- ③ robot_state_publisher 노드는 /joint_states 토픽을 구독하여 TF(Transform Frame)를 계산하고 발행한다.
- ④ URDF 파일에서 <visual> 태그는 충돌 검출을 위한 기하학적 정보를 <origin> 태그는 부모 링크의 위치 정보를 나타낸다.
- ⑤ Xacro는 URDF의 확장으로, 매크로와 변수 기능을 제공하여 반복 구조를 효율적으로 정의할 수 있다.

4

<collision>

<visual>: 기하학적 정보

<origin> 부모 링크 기준으로 자식 링크가 어느 위치에 붙는지 설정