Fuzz Testing을 통한 위성 SW 분석

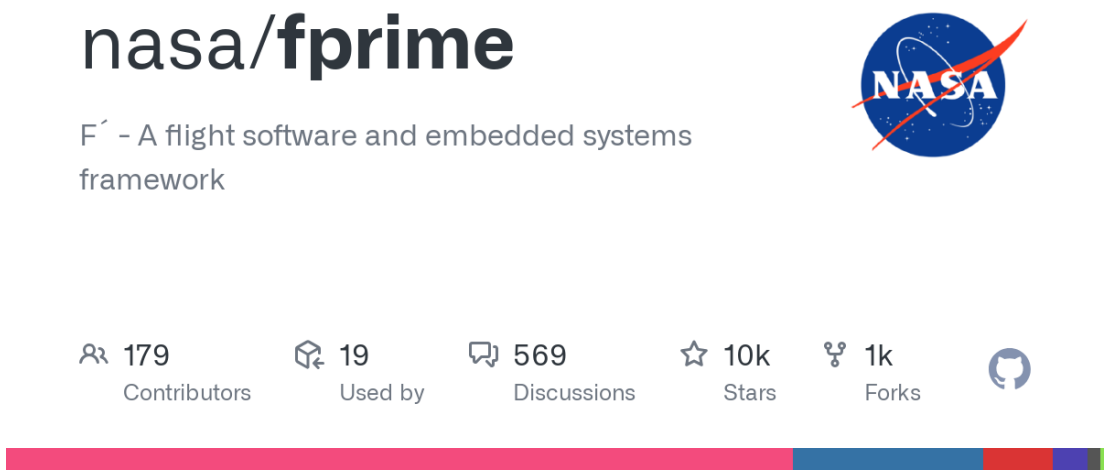
종합설계1 Week8

202002473 김승혁
201902733 이정윤
202002699 조민기

연구 배경

왜 필요한가 ?

- 최근 저비용, 소형 위성 개발 기술이 급속이 발전하면서 NASA의 오픈소스 위성 비행 소프트웨어인 fprime이 널리 사용됨.
- 하지만, 여전히 테스트 사례와 전용 도구가 부족함.
- Fuzzing은 결함을 찾아내는데 매우 효과적인 방법임.
- 그럼에도 fprime의 구조적 특성을 고려한 fuzzing 연구는 아직 초기 단계임.

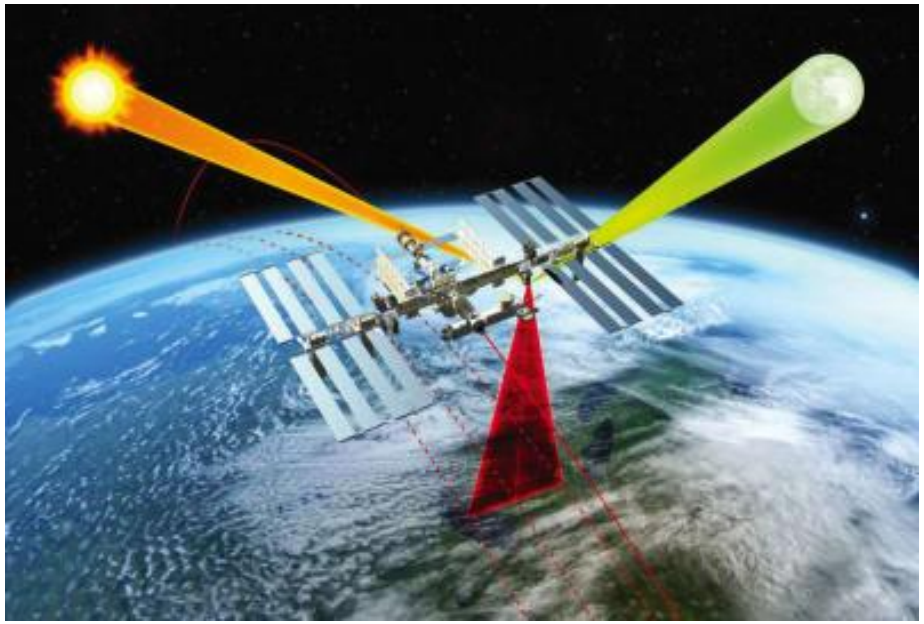


연구 목적

- Fprime 환경에서 다양한 fuzzing 도구를 비교 분석

- 컴포넌트 기반 구조에 최적화된 fuzzing 전략을 제시하는 것

-〉 발사 전 지상 테스트 단계에서 결함을 미리 잡아내고, 실제 위성 운영 중 발생할 수 있는 문제를 사전에 차단하는 것이 목표



연구 질문 및 가설

연구 질문

RQ1

- Fprime 환경에서 fuzzing을 사용하는 것이 기존 테스트 방식과 비교했을 때, 결함 탐지율에 어떤 영향을 미치는가?

RQ2

- Coverage-Guided 퍼저, Generation-Based 퍼저 등 다양한 퍼저 유형이 fprime의 컴포넌트 기반 아키텍처와 FPP 통신 방식에 따라, 결함 탐지 성능에 어떤 차이를 보이는가?

연구 가설

H1

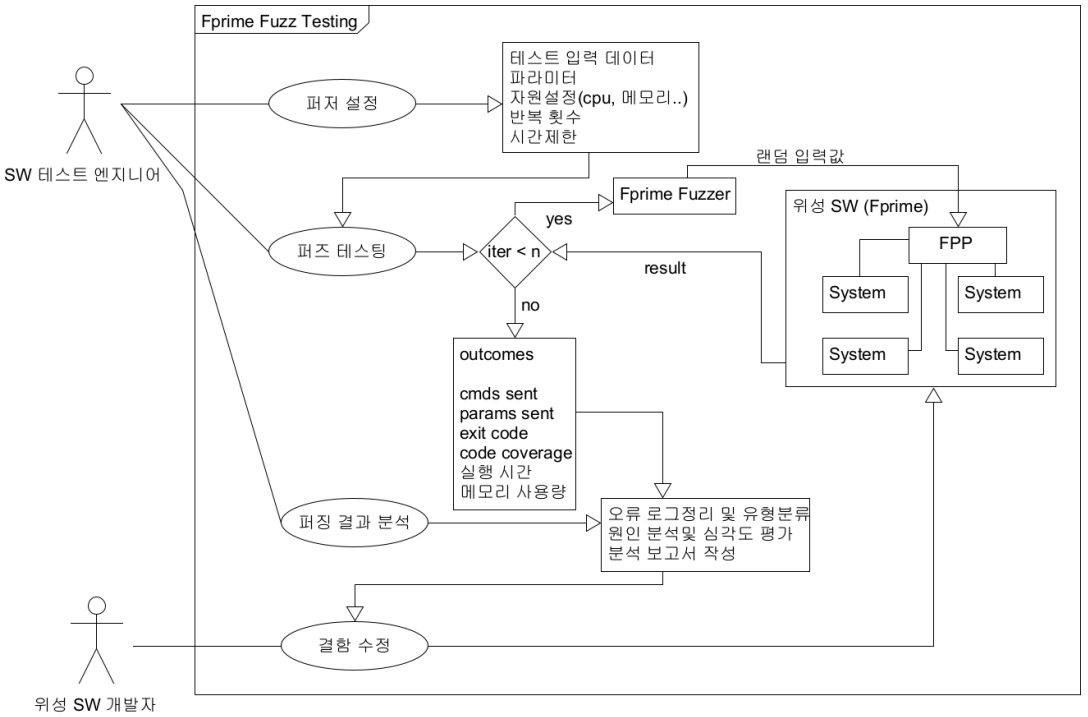
- 퍼징을 활용한 결함 탐지 활동은 기존 방식보다 결함 탐지율을 유의미하게 향상시킬 것이다.

H2

- Generation-Based 퍼저는 컴포넌트 간 복잡한 통신을 다루는데 더 효과적이어서, 더 많은 시스템 상태를 탐색하고 더 높은 결함 탐지 성능을 보일 것이다.

소프트웨어 사용 사례

1. 테스트 엔지니어가 퍼저 설정
2. 퍼저는 설정된 조건에 맞춰 fprime에 무작위 입력을 자동으로 주입하며 테스트 수행.
3. 테스트 끝나면 퍼저는 테스트 결과를 반환
4. 엔지니어는 반환된 데이터를 분석. 분석 결과를 개발자에게 전달
5. 개발자는 분석 결과를 토대로 소프트웨어 수정. 오류가 해결될 때까지 반복



문제 세부 요인 분석

직접적 요인

1. 구조적 복잡성
2. 통신 방식이 FPP라는 특수한 프로토콜로 이루어짐
3. 위성 소프트웨어가 상태 기반(Stateful)이라는 특성

간접적 요인

1. 기존 퍼저는 위성 소프트웨어의 특수성 고려 X
2. 테스트 환경 구축, 통합 어려움
3. 소형 인공위성의 예산이 제한적

   -〉 고비용의 테스트 장비 사용 불가능

1.5. 문제 해결에 대한 사용 사례 Diagram

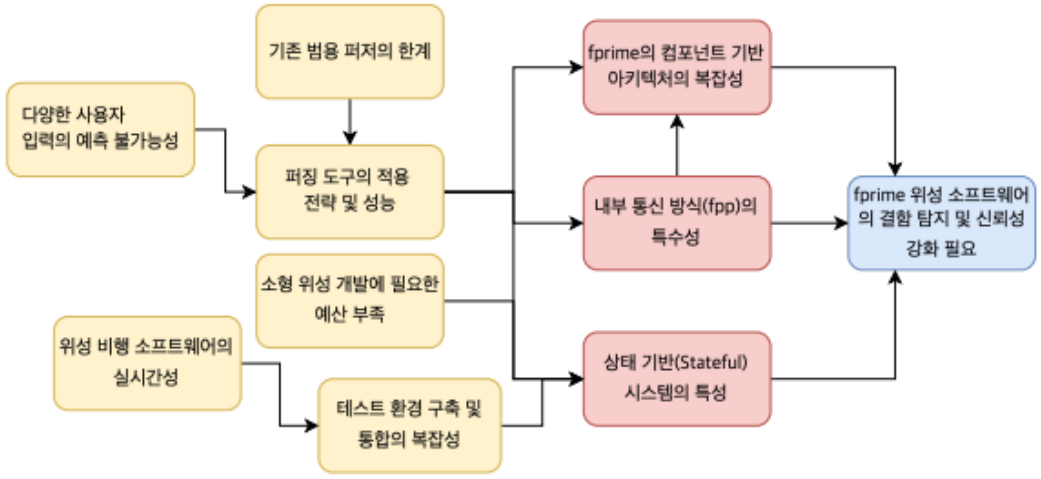


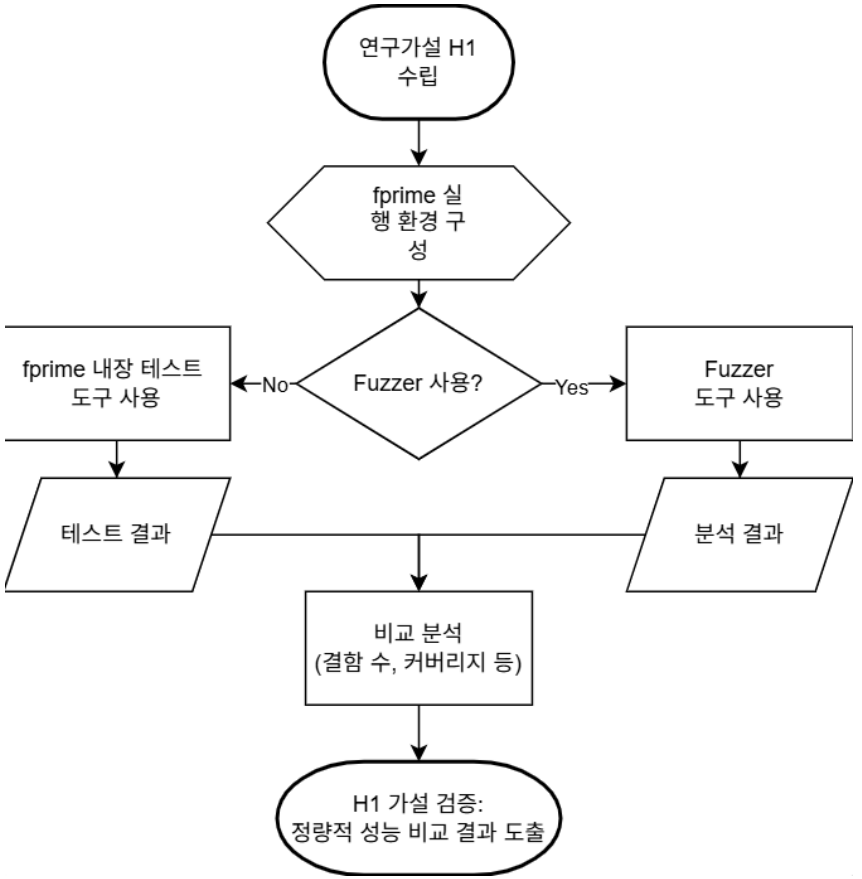
그림 2

알고리즘 순서도

퍼징이 결함 탐지율을 높일 것인가

H1. 퍼징 전략이 기존 테스트보다 결함 탐지율을 높인다는 가설 검증 순서도

1. 동일한 fprime 환경 구성.퍼저만 다르고 나머지 조건은 동일하게
2. 기본 테스트 도구를 사용해 테스트 수행
3. 퍼징 도구를 사용해 테스트 수행
4. 각 테스트 결과에서 결함 탐지율, 코드 커버리지. 자원 사용량을 비교 분석



알고리즘 순서도

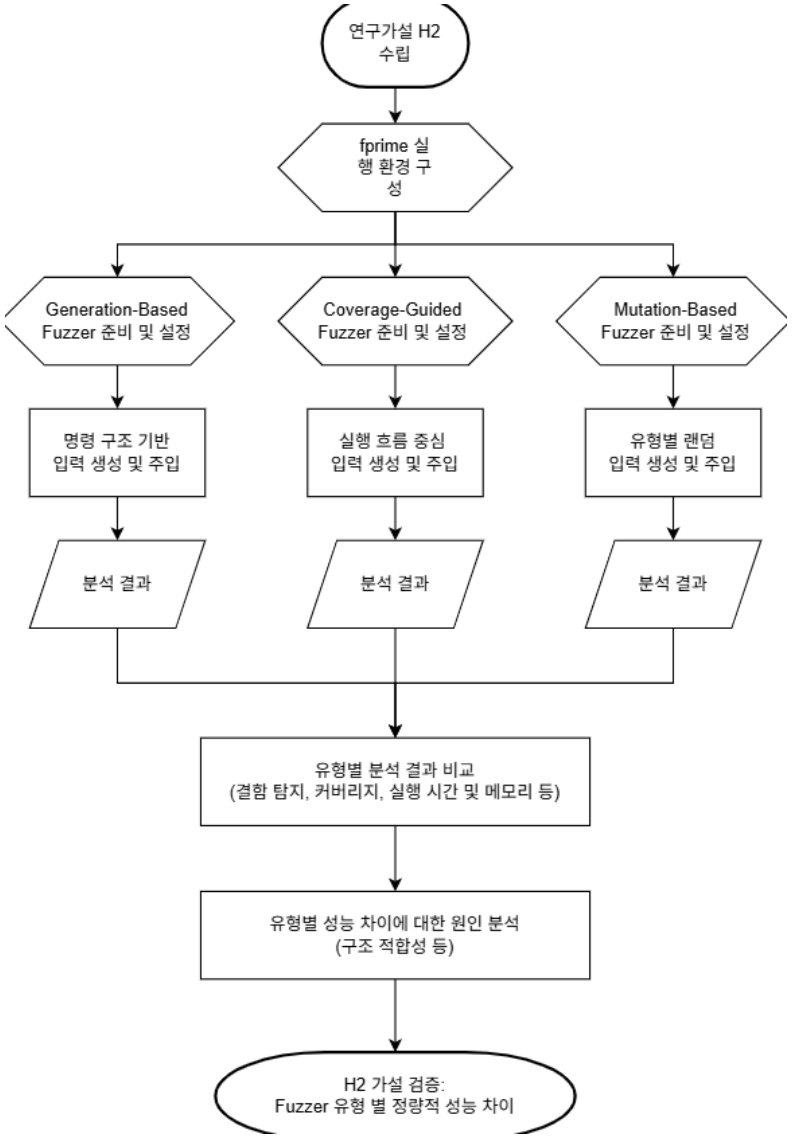
어떤 유형의 퍼저가 fprime 구조에 효과적인가

H2. Generation-Based 퍼저가 다른 퍼저보다 더 효과적이라는 가설 검증 순서도

1. 동일한 fprime 환경 구성. 퍼저 유형 별로 초기 입력 설정만 다르게 적용
2. Generation-Based 퍼저는 명령 구조 기반 입력

   Coverage-Guided 퍼저는 실행 흐름 중심 입력

   Mutation-Based 퍼저는 무작위 입력을 생성
3. 각 퍼저가 생성한 입력을 fprime에 주입해 테스트 결과를 수집
4. 수집된 결과에서 비교 분석하여 fprime에 가장 적합한 퍼저 유형 식별



Thank You