
문제정의서(연구개발계획서)


Project Name	Fuzz Testing을 통한 위성 SW 분석
-----------------	---------------------------

05 조

202002473 김승혁

201902733 이정윤

202002699 조민기

지도교수: 이성호 교수님  (서명)

Document Revision History

REV#	DATE	AFFECTED SECTION	AUTHOR
1	2025/04/02	초안 작성	김승혁
2	2025/04/03	인터뷰 목록 추가, 퇴고	김승혁

Table of Contents

1.	연구 개발의 필요성	5
2.	연구 개발의 목표 및 내용	6
3.	이해당사자 인터뷰/ 설문 인사이트	7
4.	기대 효과 및 향후 확장 가능성	9
5.	연구 개발의 추진전략 및 방법	9
6.	AI 도구 활용 정보	10
7.	참고문헌(Reference)	10

List of Figure

그림 1 브레인스토밍 결과.....	6
---------------------	---

1. 연구 개발의 필요성

최근 저비용 위성 발사와 소형 위성 기술의 발전으로 우주 산업의 문턱이 낮아짐에 따라 NASA의 fprime과 같은 오픈소스 위성 비행 소프트웨어 활용이 증가하고 있습니다. fprime은 컴포넌트 기반 아키텍처와 직접 통신(fpp) 방식을 특징으로 소형 위성 개발에 적합하지만, 역사가 짧아 검증 사례나 특화된 테스트 도구가 부족합니다. 위성 비행 소프트웨어는 임무 성공과 직결되는 미션 크리티컬 시스템으로, 사소한 오류도 치명적인 결과를 초래할 수 있습니다. 예를 들어, 온보드 컴퓨터 오작동으로 인한 자세 제어 실패는 태양 전지판의 태양 추적 실패로 이어져 배터리 방전과 위성 기능 상실을 야기할 수 있으며, 잘못된 명령 처리는 연료 낭비로 임무 수행 능력을 저하시킬 수 있습니다.

퍼징(Fuzzing)은 예기치 않은 입력값으로 소프트웨어의 잠재적 결함을 발견하는 효과적인 테스트 기법이지만, Stateful 특성을 가진 위성 비행 소프트웨어의 고유 특성과 fprime의 컴포넌트 기반 설계 및 내부 통신 방식(fpp)을 고려한 퍼징 연구는 초기 단계입니다. 기존 범용 퍼저(AFL, LibFuzzer 등)는 상태를 고려하지 못해 특정 조건에서만 발생하는 오류를 놓칠 수 있고, fpp 같은 SW 내부 프로토콜 구조를 제대로 반영하지 못하면 컴포넌트 간 인터페이스 취약점을 충분히 검증하기 어렵습니다. 그럼에도 불구하고, 이러한 한계를 극복할 수 있는 퍼징 기법이나 가이드라인은 아직 명확히 제시되지 않은 상태이며, 따라서 위성 비행 소프트웨어에 특화된 효과적인 퍼징 전략을 연구하는 것이 중요한 과제가 됩니다.

2. 연구 개발의 목표 및 내용

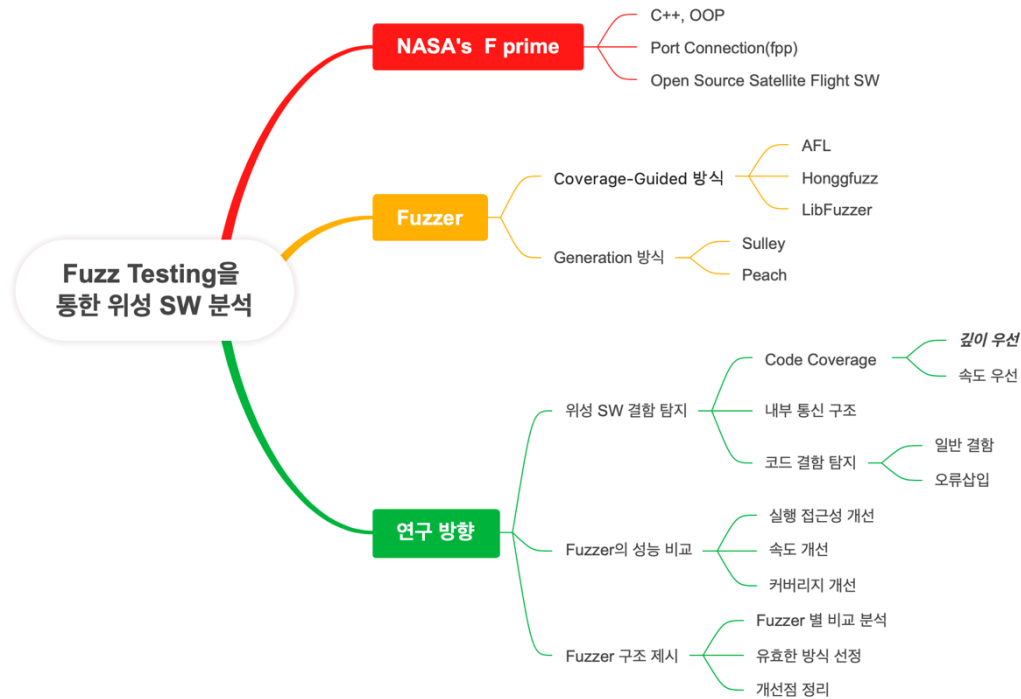


그림 1 브레인스토밍 결과

본 연구는 NASA의 fprime 위성 비행 소프트웨어 환경에서 다양한 퍼징 도구들의 성능을 비교 분석하고, fprime의 특성을 고려한 효과적인 퍼징 적용 전략을 제시하는 것입니다.

이를 위해,

1. Coverage-Guided 기반 퍼저(예: AFL)와 Generation 기반 퍼저(예: Boofuzz, Kitty 등)를 포함한 여러 유형의 퍼저를 fprime 환경에 적용합니다.
2. fprime의 주요 특징인 Stateful 환경과 컴포넌트 기반 아키텍처, 컴포넌트 간 통신 방식(fpp)을 고려하여 각 퍼저의 성능(코드 커버리지, 결함 탐지율, 상태 탐색 능력 등)을 정량적으로 평가하고 비교 분석합니다.
3. 분석 결과를 바탕으로 fprime 환경에서 각 퍼저 유형의 장단점을 명확히 하고, 어떤 유형의 퍼저가 fprime의 특정 구조나 통신 방식 테스트에 더 적합한지 밝힙니다.

최종적으로, fprime 기반 위성 소프트웨어의 신뢰성 검증을 위한 최적화된 퍼징 적용 방안과 향후 fprime 특화 퍼저 개발에 필요한 요구사항을 도출하는 것을 목표로 합니다.

3. 이해당사자 인터뷰/ 설문 인사이드

저희는 총 3명의 이해 관계자 분들과 인터뷰를 나눌 수 있었으며, 각각 다른 경험을 쌓아 온 분들에게 다양한 관점에서 조언 및 인사이트를 얻을 수 있었습니다.

● 이성호 교수님 (Software Analysis & Testing Research Group, 충남대학교)

인터뷰 내용	<ul style="list-style-type: none"> ● 위성 SW 테스트: 기능 테스트, 유닛 테스트, 정적 분석 등이 활용되나 미션 크리티컬 SW 맞춤형 기법은 부족함. ● 퍼징 연구 방향: 퍼저 성능 비교(정량 지표 활용: 커버리지, 결함 탐지율 등) 후 실제 결함 탐색 시도 권장. 간단한 대상(fprime)부터 시작하여 경험 축적 후 복잡한 대상(cFS) 고려. ● 기존 퍼저 한계: 위성 SW의 특성(예: 모듈 간 통신)을 고려하지 못해 커버리지 확보 및 모듈 간 상호작용 결함 탐지에 한계가 있을 수 있음.
인사이드	대상 SW(fprime)를 실제 구동해보고 단계적으로 퍼저를 적용해보는 것이 가장 중요하며, 실제 문제를 직접 부딪혀야 문제 정의 및 해결 방향이 명확해지기 때문에, 정량적 평가 지표 설정을 통해 분석해 봐야한다.

● 김형신 교수님 (Embedded Systems Laboratory, 충남대학교)

인터뷰 내용	<ul style="list-style-type: none"> ● 위성 SW 오류의 치명성: 온보드 컴퓨터 오작동은 치명적. 특히 자세 제어 실패로 인한 배터리 방전, 잘못된 명령 처리로 인한 연료 낭비 등은 위성 기능 상실 또는 임무 실패로 이어질 수 있음. ● 오류 발생 원인: 방사선, 온도 등 환경적 요인도 있지만, 기록상 가장 많은 원인은 운영자의 잘못된 명령 전송(Operator Error)임. 소프트웨어 자체 결함도 원인이 될 수 있음. ● 위성 시스템 특징: 고신뢰성 설계(하드웨어/소프트웨어 리던던시), 자동 복구 기능(와치독 등) 존재. 하지만 비정상적인 상태(예: 잘못된 컨트롤 로직 수행)는 탐지 및 복구가 어려울 수 있음. 배터리 방전 시 원격 복구 불가능. ● 사전 테스트 중요성: 발사 전 지상에서의 철저한 테스트(정적/동적 테스트, 고장 주입 시험 등)가 매우 중요함. 대학 개발 위성은 검증 부족으로 실패 확률 높음. 퍼징은 다양한 테스트 방법 중 하나로, 특히 예상치 못한 입력에 대한 강건성 검증에 기여할 수 있음.
인사이드	위성 SW는 일반 SW와 다른 특수성을 가지고 있으며, 테스트에서도 이를 반영하는 효과적인 테스트가 매우 중요하다.

● Jakob Holst Svenningsen (퍼징 연구 경험자(MMS-Fuzzer), DMC)

인터뷰 내용	<ul style="list-style-type: none"> ● 연구 경험: 상태 기반 시스템(MMS) 대상 네트워크 프로토콜 퍼저 개발 경험. ● 퍼저 유형 비교: Mutation 기반 퍼저는 사용이 간편하고 상태 없는 시스템(Stateless)에 효과적이거나, 상태 기반 시스템(Stateful)의 복잡한 구조나 취약점 식별에는 한계가 있음. Generation 기반/Genetic 퍼저가 상태 기반 시스템에 더 적합할 수 있음. ● 퍼징 연구 어려움: 기존 라이브러리(Kitty, Boofuzz 등) 문서 부족, 좋은 테스트 케이스 설계의 어려움 (단순 랜덤 입력 이상의 설계 필요). ● 평가 지표: 실행 시간 대비 발견된 취약점 수(효율성), 서버 응답 시간 (DoS 취약점 관련) 등. 시스템의 중요도에 따라 결과 해석 필요. ● 개발 조언: 대상 프로토콜(fprime의 fpp 등)에 대한 깊은 이해가 중요함. 기존 라이브러리 활용 권장.
인사이트	대상 SW에 특성(프로토콜 등)을 잘 분석한 뒤, 이에 맞는 Fuzzing 전략을 찾아 적용하는 것이 가장 중요하다.

4. 기대 효과 및 향후 확장 가능성

1. fprime 신뢰성 검증 효율성 증대: fprime 환경에 적합한 퍼징 도구 및 적용 전략 가이드라인을 제공하여, 향후 fprime 기반 위성 시스템 개발 시 소프트웨어 신뢰성 검증의 효율성을 높이고 초기 단계에서 잠재적 결함을 발견할 가능성을 높입니다.
2. 퍼저 성능 비교 데이터 제공: Mutation 기반 및 Generation 기반 퍼저를 fprime 환경에서 비교한 정량적 데이터를 제공하여, 유사 시스템에 퍼징을 적용하려는 연구자나 개발자에게 참고 자료를 제공합니다.
3. fprime 특화 퍼저 개발: 본 연구의 분석 결과(fprime 환경에서의 퍼저 유형별 장단점, fpp 처리 방식 등)를 바탕으로 fprime의 아키텍처와 통신 방식에 최적화된 새로운 퍼저 또는 퍼징 프레임워크 개발 연구로 확장될 수 있습니다.
4. 다른 위성 SW 적용 가능성 연구: 본 연구에서 사용된 퍼저 비교 방법론 및 분석 결과를 cFS(Core Flight System) 등 다른 오픈소스 위성 비행 소프트웨어 시스템에 확장 적용하여 일반화 가능성을 검증할 수 있습니다.

5. 연구 개발의 추진전략 및 방법

- 4월 초: fprime 실행 환경 설정 완료
- 4월 중: fprime 구조 분석
- 4월 말: 1차 Fuzzer 적용 및 결과 분석
- 5월 초: 2차 Fuzzer 적용 및 분석
- 5월 중: 1차, 2차 비교 분석 및 3차 Fuzzer 적용
- 5월 말: 연구 결과 분석 및 논문 초안 작성

협업은 Github을 통해 진행하며, 각자 특정 Fuzzer를 사용해 분석해본 결과를 공유하고 연구 방향을 정하는 주간 회의를 매주 가지며 진행합니다. 주간 회의는 1회 대면, 1회 비대면으로 진행하며, 소통은 디스코드 앱을 통해 채팅 및 음성 통화로 진행합니다.

6. AI 도구 활용 정보

사용 도구	네이버 클로바노트, Claude 3.7, o3-mini
사용 목적	인터뷰 전사문 생성, 인터뷰 내용 요약
프롬프트	<ul style="list-style-type: none"> ● 인터뷰 내용을 정리해줘 ● 작성한 문단에서 문법에 틀린 표현이 있는지 검사해줘
반영 위치	<ol style="list-style-type: none"> 1. 인터뷰 질문 목록 (p.7) 2. 연구 개발의 필요성 (p.5)
수작업	있음(논리 보강, 인터뷰 내용 정정)
수정	

7. 참고문헌(Reference)

1. Incorporating security practices into the development of the Maritime Messaging Service - Jakob Holst Svenningsen (2024)
2. 인공 위성 탑재 소프트웨어의 소프트 에러 고장 주입시험 - 배지훈, 김형신(2022)