



# Google Summer Of Code

## 참여 프로젝트 소개

이태훈

경기대학교



# 차례

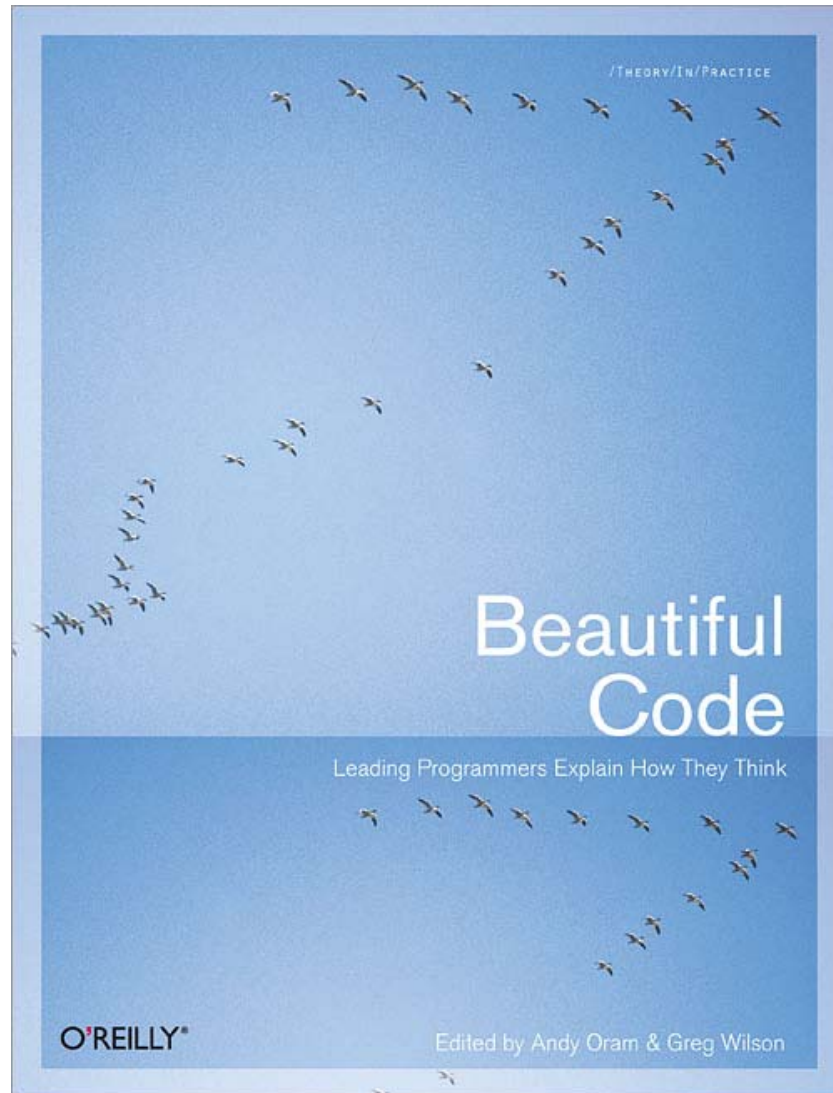
- Google Summer Of Code 소개
- 대학교 오픈 소스
- JavaPathfinder 소개



# Google Summer Of Code

- Google 의 오픈 소스 지원 행사
  - 2005년 부터 시작
  - 총 4500\$ 를 지원
    - 시작 : 500\$
    - 중간 평가 : 2000\$
    - 최종 평가 : 2000\$
  - 2008년 현재
    - 175개의 오픈 소스 프로젝트
    - 1125 명의 학생이 참여

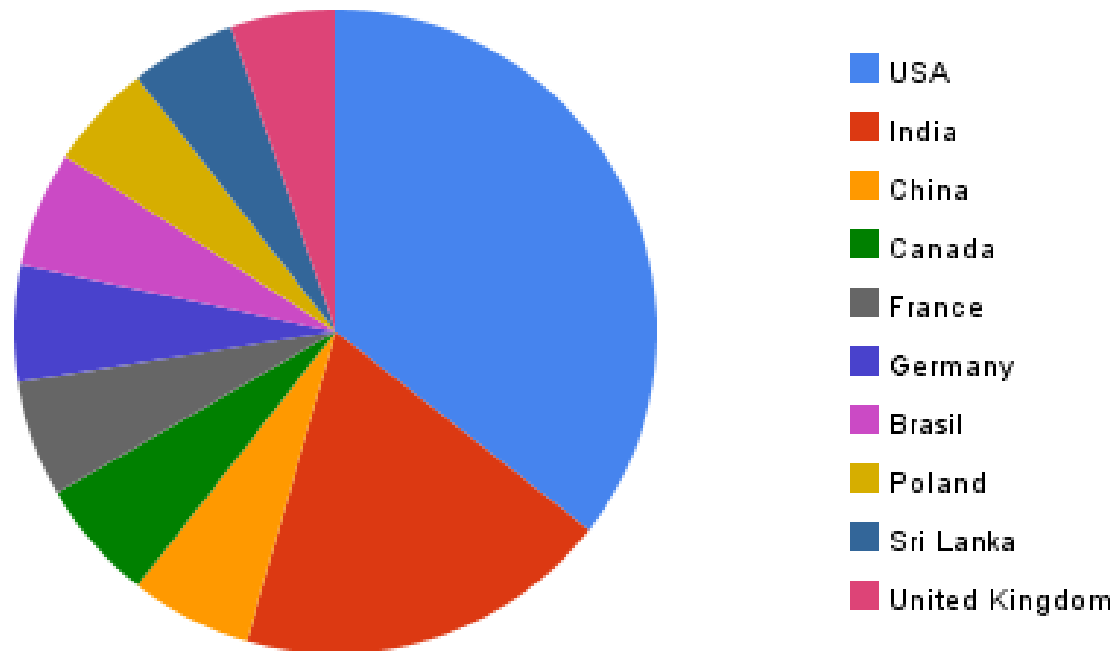
# 받은 것들



# GSOC 공식 통계(1)



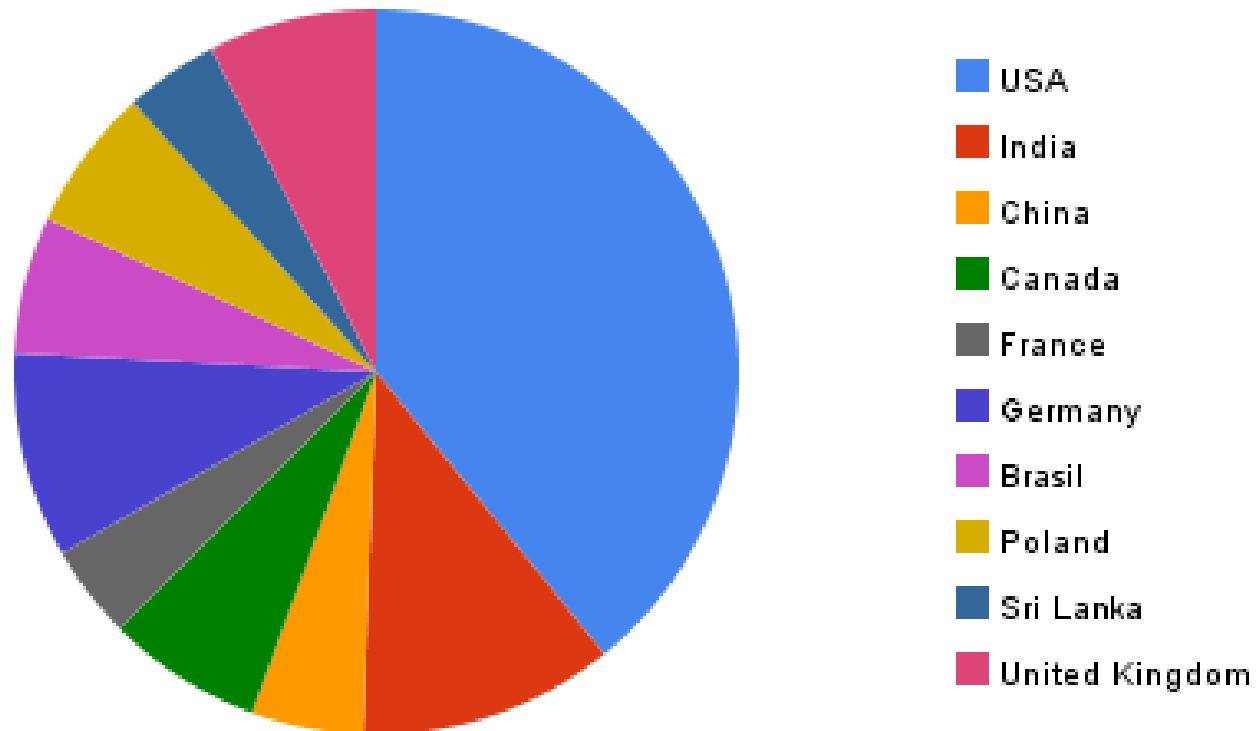
2008 GSoC Applicants: Top 10 Countries



# GSOC 공식 통계(2)



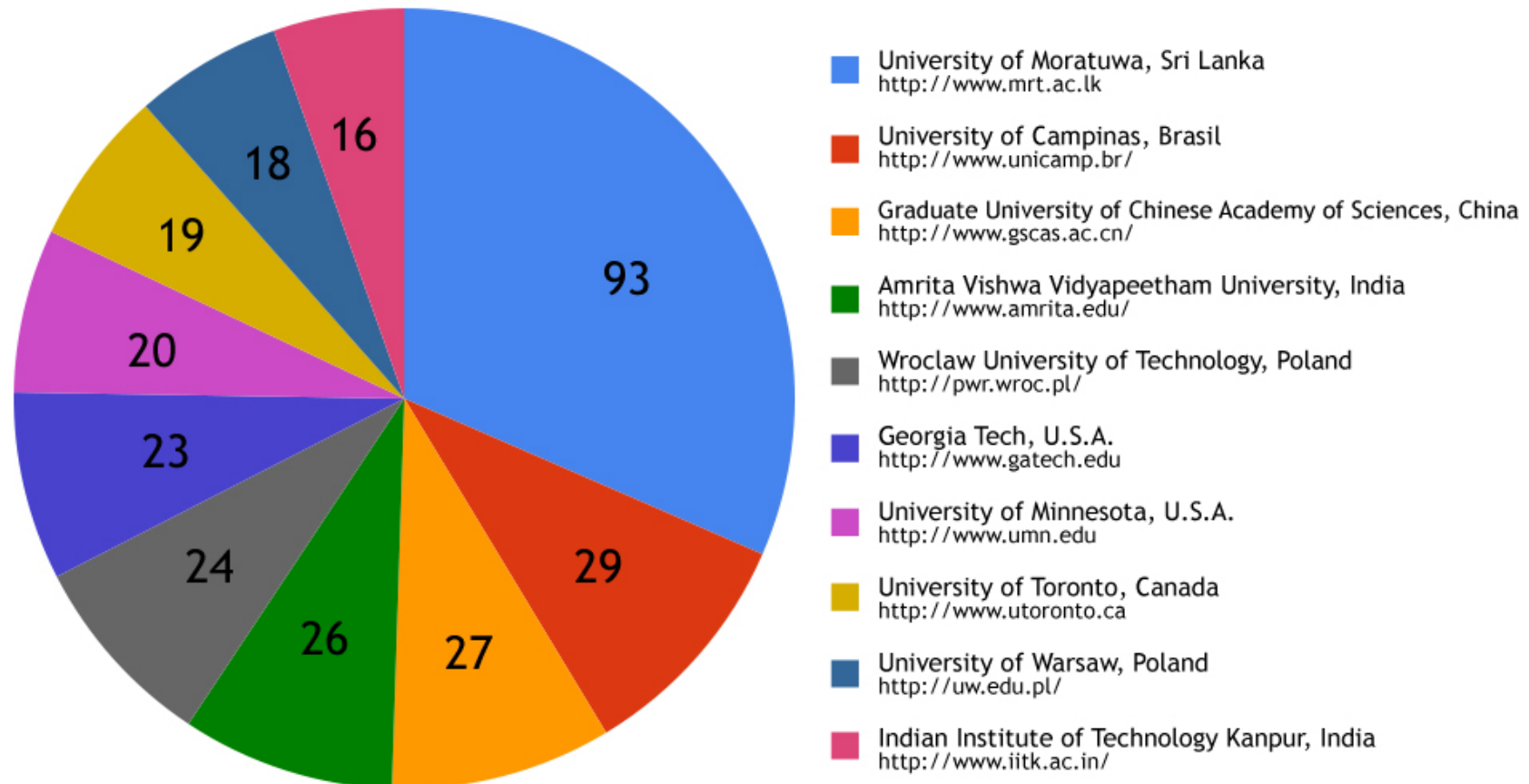
2008 GSoC Accepted Students: Top 10 Countries



# GSOC 공식 통계(3)



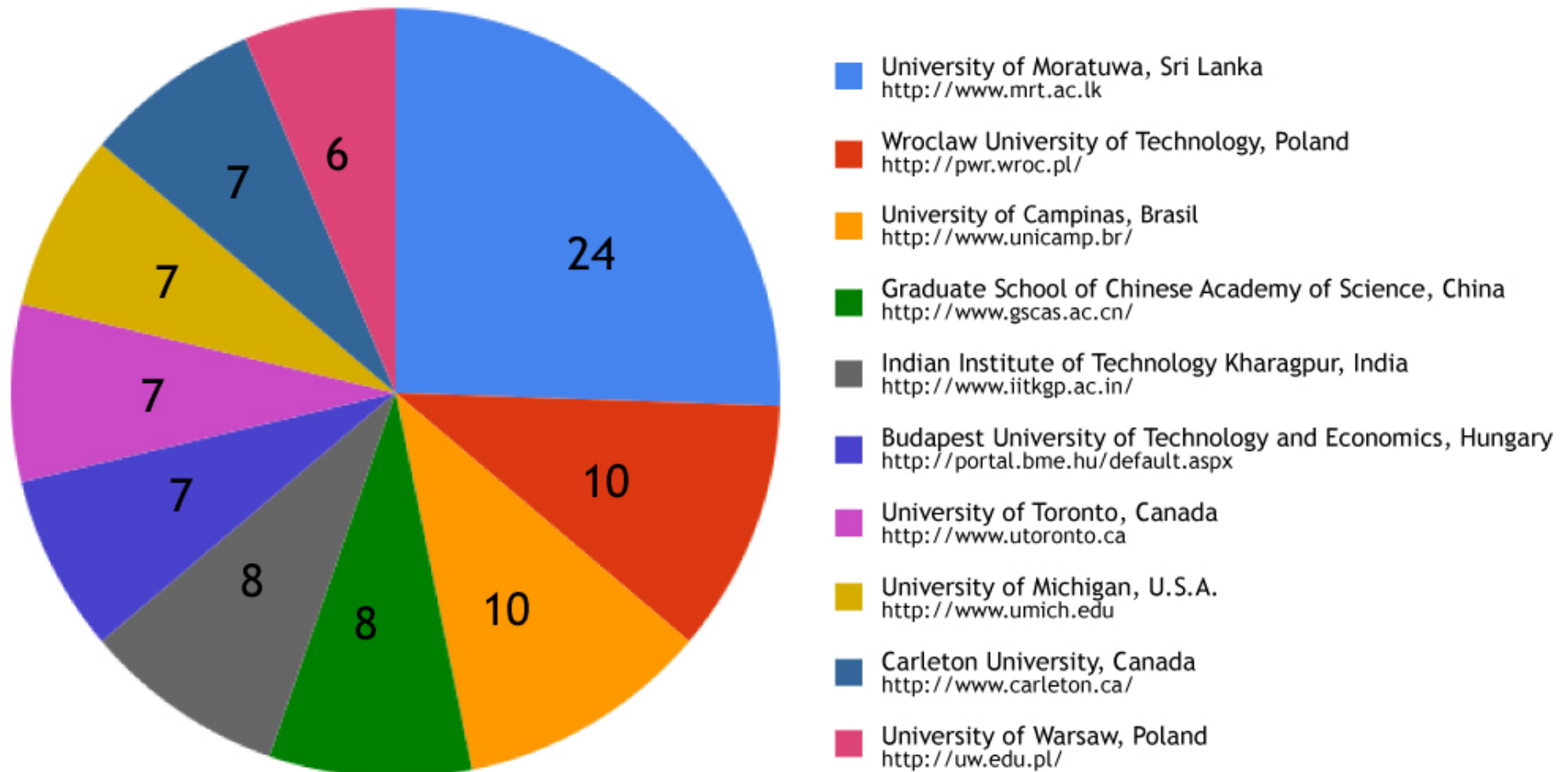
2008 GSoC Applicants: Top 10 Universities



# GSOC 공식 통계(4)



2008 GSoC Accepted: Top 10 Universities







# GSOC 비공식 통계(1)

- 대한민국을 기준
  - 참여 오픈 소스 프로젝트 : 0
  - 참여 mentor : 0
  - 참여 학생 : 6
    - 학위
      - 박사 과정 : 5
      - 학사 과정 : 1
    - 현재 소속
      - 해외 : 4
      - 국내 : 2



## GSOC 비공식 통계(2)

- 특정 국가에서만 개발중인 오픈 소스
  - 3개
- 특정 대학에서 개발중인 오픈 소스
  - 15개
- 특정 대학에서 시작한 오픈 소스
  - 18개



# 대학과 오픈 소스(1)

- 대학을 기반으로 한 오픈 소스
  - 10% 를 약간 넘음
- 우리나라는?
  - 개인이 개발하고 있는 몇몇 오픈 소스는 널리 사용 중
  - 대학에서 공개한 오픈 소스
    - 비 공식적으론 전혀 없음
- 원인은?



## 대학과 오픈 소스(2)

- 원인
  - 대부분의 대학교에서는 국가의 지원을 받음
  - 국가의 지원을 받고 난 후 좋은 평가를 받기 위해선
    - 논문을 많이 쓰면 좋은 평가를 받음
    - 개발한걸 팔 수 있다면 좋은 평가를 받음
    - 특허가 많다면 좋은 평가를 받음
- 대학 기반의 오픈 소스 활성화를 위해서
  - ????

# Model Checking 소개(1)



- ACM, the Association for Computing Machinery, has named **Edmund M. Clarke, E. Allen Emerson, and Joseph Sifakis** the **winners of the 2007 A.M. Turing Award**, widely considered the most prestigious award in computing, for their original and continuing research in a quality assurance process known as **Model Checking**.



# Model Checking 소개 (2)





# Model Checking 소개 (3)

- 모델 체킹
  - 시스템에 오류가 존재하는지 존재하지 않는지 검사하는 정형 검증 (Formal verification) 기술
  - 시스템과 속성(property)을 받아들여서 모든 가능한 경우를 검사
  - 속성의 예제
    - 데드락이 존재하지 않는다, assert 위반이 존재하지 않는다.
  - 결과는
    - 시스템이 속성을 만족
    - 시스템이 속성을 만족하지 않음
  - 오류가 존재할 경우 오류를 찾아줌
  - 모든 가능한 경우를 조사하기 때문에 큰 시스템에 적용하기 힘 듬



# Model Checking 소개(4)

- 테스트
  - 오류가 존재하는 것을 보여줄 수는 있지만  
오류가 존재하지 않는 것을 보여줄 순 없다.
    - 다익스트라
- 모델 체킹
  - 시스템에 오류가 존재하는 것을 보여줄 수 있다.
  - 오류가 존재하지 않는 것을 증명할 수 있다.



## Windows

A fatal exception 0E has occurred at 0020:C0011E36 in UXD UMM(01) + 00010E36. The current application will be terminated.

- \* Press any key to terminate the current application.
- \* Press CTRL+ALT+DEL again to restart your computer. You will lose any unsaved information in all applications.

Press any key to continue

# Microsoft Research 의 Model Checking 도구



SLAM Project - Windows Internet Explorer

http://research.microsoft.com/slam/

Google blue screen 실행 즐겨찾기 124 차단됨 a1 번역 blue screen 설정

SLAM Project

# SLAM

`if(!node->next) { ++vis; proc->end(); node->next = node; }`

"Things like even software verification, this has been the Holy Grail of computer science for many decades but now in some very key areas, for example, driver verification we're building tools that can do actual proof about the software and how it works in order to guarantee the reliability." Bill Gates, April 18, 2002. [Keynote address at WinHec 2002](#)

[What's New?](#) | [Papers](#) | [News](#) | [Presentations](#)  
[Software](#) | [People](#) | [Contact SLAM](#)

## What's New?

- **Static Driver Verifier (SDV) Rule Development Kit (RDK) available**
  - The RDK is an extension to SDV that allows you to adapt SDV to support additional frameworks (or APIs) and write custom SLIC rules for this framework. The goal of the RDK is to allow researchers to experiment with writing SLIC rules for APIs and to gain experience using the SLAM verification engine that underlies SDV.
  - See [README.htm](#) for more information about the RDK.
  - E-mail [rdk\\_req@microsoft.com](mailto:rdk_req@microsoft.com) to request a copy of the RDK.
- **Get the Windows (Vista) Driver Development Kit, which includes SDV**
  - <http://www.microsoft.com/whdc/devtools/WDK/default.msp>
- **Developing Drivers with the Windows® Driver Foundation**, a Microsoft Press book, is now in print, including a chapter about Static Driver Verifier (SDV), which has new rules to enable analysis of drivers written against the Kernel-model Driver Framework API.

## Papers

- **The SLAM Process**
  - **Thorough Static Analysis of Device Drivers** (pdf), Thomas Ball, Ella Bounimova, Byron Cook, Vladimir Levin, Jakob Lichtenberg, Con McGarvey, Bohus Ondrusek, Sriram K. Rajamani and Abdullah Ustuner (Microsoft), *EuroSys 2006*
  - **The SLAM Project: Debugging System Software via Static Analysis**, Thomas Ball, Sriram K. Rajamani, *POPL 2002*, January 2002, pages 1-3 [PS,PDF]. *A three-page high-level overview of SLAM. See the following paper for more details.*
  - **Automatically Validating Temporal Safety Properties of Interfaces** [PS,PDF], Thomas Ball, Sriram K. Rajamani. *SPIN 2001*. Workshop on Model Checking of Software. LNCS 2057. May 2001.

18 완료 인터넷 100%



presented by [WWW.WALLPAPERS.COM](http://WWW.WALLPAPERS.COM)

# NASA 의 모델 체킹 도구




Java PathFinder - Windows Internet Explorer

http://javapathfinder.sourceforge.net/

Google spaceship 실행 즐겨찾기 팝업 허용 a 번역 spaceship 설정

Java PathFinder

 **Java™ PathFinder** .. the (swiss army knife | vulcan mind probe | spiral-of-death | <what do you want it to be today?>) of Java program verification **SOURCEFORGE.NET®**

Home News About Us License & Disclaimers Download, Lists, Bugs etc. Changes Links Documentation

• What is Java PathFinder  
• What can be checked with JPF  
• Model Checking vs Testing  
• Extensibility  
• The State of Affairs  
• History and Credits  
• Prerequisites  
• Obtaining and Installing JPF  
• Configuring JPF Runtime Options  
• Running JPF

**Welcome to the Home of Java PathFinder**

Java PathFinder (JPF) is a system to verify executable Java bytecode programs. In its basic form, it is a Java Virtual Machine (JVM) that is used as an explicit state software model checker, systematically exploring all potential execution paths of a program to find violations of properties like deadlocks or unhandled exceptions. Unlike traditional debuggers, JPF reports the entire execution path that leads to a defect. JPF is especially well-suited to finding hard-to-test concurrency defects in multithreaded programs.

While software model checking in theory sounds like a safe and robust verification method, reality shows that it does not scale well. To make it practical, a model checker has to employ flexible heuristics and state abstractions. JPF is unique in terms of its configurability and extensibility, and hence is a good platform to explore new ways to improve scalability.

JPF is a pure Java application that can be run either as a standalone command line tool, or embedded into systems like development environments. It was mostly developed - and is still used - at the NASA Ames Research Center. Started in 1999 as a feasibility study for software model checking, JPF has found its way into academia and industry, and has even helped detect defects in real spacecraft.

**Headline News**

- 05/01/08: **2008 Workshop slides are up** - in case you were able to attend, you already know it was an exciting event. If you weren't there, you can see what you've missed on the [event schedule page](#)
- 04/15/08: **JPF and Google's Summer of Code** - we were able to get 10 exciting JPF projects approved for this years [Google's summer of code](#). It's going to be a hot summer. Expect to see a lot more traffic on the [javapathfinder-devel mailing list](#).
- 04/04/08: **Fujitsu Japan announces WEAVE framework** - this is a great testimony for JPF's acceptance, esp. since [WEAVE](#) is about verification of web applications. Nobody wants their banking application to crash in mid-transaction, so everybody wants JPF.



# 널리 알려진 모델 체킹 도구

- SLAM
  - Microsoft Research 에서 개발한 모델 체킹 도구
  - 윈도우 디바이스 드라이버를 검사
  - 최근 DDK 에 포함됨
- JavaPathFinder
  - Java 모델 체킹 도구
  - NASA 에서 개발
- BLAST
  - Berkley 대학에서 개발중인 C 언어 모델 체킹 도구

# 제가 GSOC 를 통해 수행 하는 작업

- Java Model Checking 도구인 JavaPathFinder 를
  - 자바 기반 웹 서버에 적용
  - 자바 기반 서블릿에 적용



# 감사합니다.

- 질문은 [taehoon@kgu.ac.kr](mailto:taehoon@kgu.ac.kr)