

How to Contribute to OSS

—

서상현

2015-04-17

N A V E R | L | A | B | S |

오픈 소스 프로젝트에 참여하기

- 정말로 좋은 자료가 있습니다
- Python 프로젝트에 참여한 장혜식님이 만든 자료
- [오픈소스 프로젝트에 참여하기](#)
- [오픈소스 프로젝트 생존 가이드](#)
- [구글 Summer of Code 필승전략](#)

전략

오픈소스는 기술이 아니라 설득이다.
그 당위성을 설득할 수 없다면 패치는 받아들여지지 않는다.

-- Redis 프로젝트에 참여한 강대명님 말씀

무엇을 설득해야 할까요

- 이걸 왜 해야 하는지
 - 이미 알려진 버그를 수정하거나 구현하기로 합의한 기능을 구현한 거라면 넘어가도 괜찮습니다
 - 버그라면 올바른 동작, 현재 동작, 차이점, 차이를 볼 수 있는 재현 방법을 적도록 합시다
- 왜 이렇게 했는지
 - 여러 가지 방법이 있다면 왜 이 방법으로 했는지 말할 수 있어야 하겠습니다
- 앞으로는 어떻게 할 것인지
 - 코드를 고쳤다면 자동화된 테스트를 추가해 주세요
 - 기능을 추가했다면 문서화를 하는 것이 좋습니다

신뢰를 쌓아야 합니다

- 처음에는 작게 시작합니다
- 코드는 유지보수가 필요합니다
 - 작은 코드는 기존 프로젝트에 있던 사람이 맡아줄 수 있습니다
 - 유지보수를 해야 하는 사람은 상대방이므로, 상대방이 요구하는 사항이 있으면 받아주도록 합니다
 - 큰 코드를 공헌하려면 유지보수의 책임을 져야 합니다
 - 문제가 생겼을 때 나몰라라 하지 않을 거라고 처음 보는 사람을 믿기는 어렵습니다

전술

- 받는 사람이 편하도록
- 매뉴얼을 읽읍시다
 - 참여 방법에 대한 매뉴얼이 있다면 한 줄도 빠짐없이 읽도록 합시다
- 선례를 따릅니다
 - 다른 사람은 어떻게 하나 살펴보세요

패치 만들기

- 최신 소스를 받아서
 - 예전 버전에 패치하면 받는 쪽이 불편합니다
 - 다른 곳으로 개발이 옮겨갔는데 이전 소스를 받는 실수 등을 주의
 - 최근 활동을 확인하고 검색을 해봅니다
- 수정하고
 - 수정은 최소한도로 하는 것이 받기 편합니다
 - 자기 스타일대로 포맷을 바꾼다거나 하지 맙시다
 - 수정 사항이 여럿이라면 커밋을 분리
 - 리팩토링을 한다면 리팩토링 커밋은 기능 변경이 없어야 합니다
- 확인
 - 자동화된 테스트가 있다면 통과되는 것을 확인하세요

패치 보내기

- 보내라는 데로 보냅시다
 - 풀 리퀘스트 (GitHub, Bitbucket, ...)
 - 코드 리뷰 (Gerrit, Phabricator, Review Board, ...)
 - 이슈 트래커 (Bugzilla, Trac, ...)
 - 메일링 리스트
 - 개인 메일
- 제목을 잘 붙이세요
- 설명도 정성껏
- 패치 보내기의 목표는 여러분이 패치를 만들며 거쳤던 수정/확인 과정을 받는 쪽에서 재현할 수 있도록 하는 것입니다

패치 집어넣기

- 리뷰 코멘트는 모두 답변
- 최신 소스가 변경되어 충돌이 나면 rebase
- 반응은 빠르게
 - 당장 고칠 수 없더라도, 고치겠다고 말이라도 먼저 합시다
 - 언제까지 하겠다고 쓰는 것도 좋습니다 (지키도록 합니다)
- 실시간으로 논의할 수 있으면 가장 좋습니다
 - 시간대를 확인
 - IRC 등 실시간 채팅을 활용
- 반응이 없을 때는 어떻게?

맺는말

- 고마움을 표현합시다
 - 멋진 프로젝트라면 멋지다고 메일에 쓰고
 - 하는 일에 도움이 되었다면 어떻게 도움이 되었는지 씁시다
 - 받는 쪽에서는 큰 힘이 됩니다
- 패치가 아니라도 좋습니다
 - 문제를 만났다면 직접 고치지 못하더라도 버그 보고를 합시다
 - 인터페이스나 문서를 한국어로 번역하는 것도 좋습니다
 - 블로그에 어떻게 잘 썼다는 글을 쓰는 것도 도움이 됩니다
- 모두에게 도움이 됩니다
 - 코드로 세상을 이롭게
 - 논술 실력과 영작 실력이 쑥쑥
 - 고수 개발자들에게 개인 과외를 받을 수 있는 기회