문제 해결을 즐기는 개발자 최재영입니다

Email: 000111000e@gmail.com **Phone:** 01083405932

GitHub: https://github.com/cloudchamb3r

문제에 대해 심층적인 탐구를 탐구를 통해 근본 원인을 파악하고 해결책을 제시하는 능력을 가지고 있습니다.

Skills

Categogry	Skill
Language	`Java(下)` / `Kotlin(下)` / `C++(下)` / `NodeJS(下)` / `Python3(下)` / `TS/JS(下)` / `C#(下)` / `Go(下)`
Frameworks & Libraries	`Spring Boot(下)` / `NestJS(下)` / `ExpressJS(下)` / `React(下)` / `Tailwind(下)` / `WPF(下)` / `VueJS(下)`
Database	`SQLServer(下)` / `MySQL(下)` / `Postgresql(下)`
etc	`Debugging(中)`

Experience

SofwareMaestro 15기 (2024 - 현재)

- 현재 코드리뷰 보조 솔루션 Codection 프로젝트 진행 중

Acorn Academy (2023 - 2024)

- 실시간 다대다 채팅, 화상 공유를 지원하는 Dotori 프로젝트 진행
- 학업 우수자로서 수료

Medical Standard (2021 - 2022)

- 사내 웹 프로젝트의 보안 취약점 발견 및 개선 방안 제시
- C++ Dicom Library 를 node-gyp 이용하여 Typescript로 포팅
- PACS Client 메모리 누수 해결 및 성능 개선
- 개발 편의성 향상을 위한 TDS SQL Query Sniffer 개발

42Seoul 2기 (2019 - 2020)

- 기수 내 최단 기간 공통 과정 돌파

Projects (Problem & Solution)

Codection

오픈 소스 프로젝트인 gitea 를 기반으로 코드 리뷰 서비스 개발

문제 상황 및 해결 경험

비동기 요청이 정상적으로 이루어지지 않는 문제
Codection 에서는 Pull Request 가 올라오면, 자동적으로 OpenAI 의 API 를 활용하여 AI 응답을 받는 로직이 존재합니다. 이때 이 API 는 상당한 시간이 걸리기에 goroutine 을 이용하여 비동기적으로 응답을 받고, 응답결과를 데이터베이스에 저장하도록 작성 되어있습니다.
하지만 어떤 이유에서인지, 정상적으로 작동하지 않았고 버그 해결에 도움을 달라는 팀원의 요청에 의해 관련 코드를 분석하였습니다. 내부 코드를 검토하다, goroutine 에 context를 전달하는데 이 부분에서 부모의 컨텍스트와 자식의 컨텍스트의 예상되는 수명 차이를 문제일것이라 추측하였습니다.

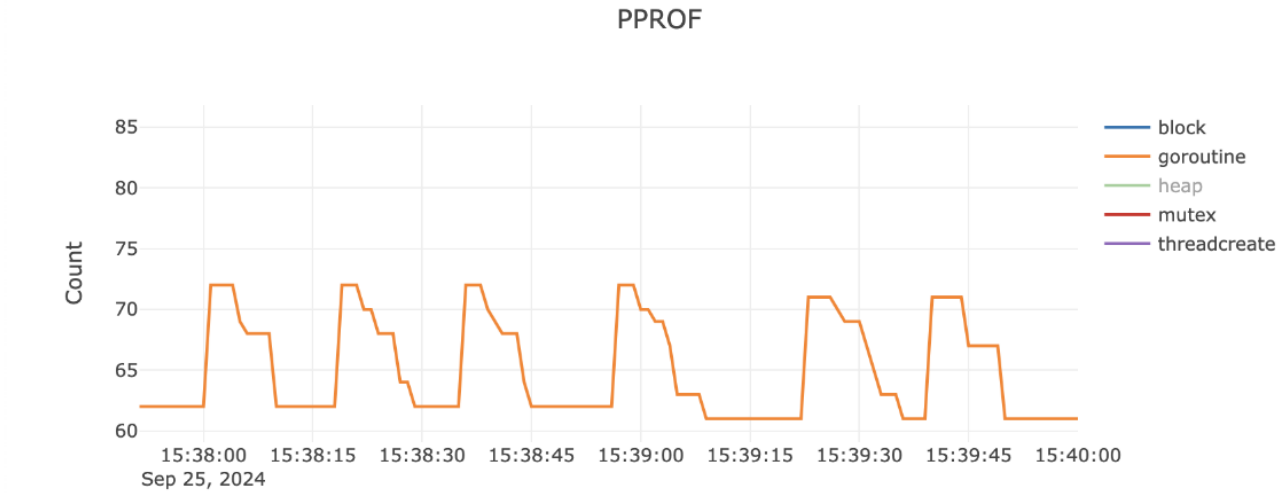
디버깅을 통해 저의 예상대로 부모와 자식의 요구되는 context 수명의 불일치가 일어나는게 원인이였다는 것을 검증하였고, 이를 해결하는 간단한 코드를 작성해 이 문제를 해결하였습니다.

```go
// 다음의 Context 는 표준 라이브러리 Context의 Wrapper입니다
func (c Context) NewChildContext() *Context {
    c.Base.originCtx = context.WithoutCancel(c.Base.originCtx)
    return &c
}
```

또한 혹시나 이렇게 복제된 context 및 goroutine이 계속 남아 문제가 되지 않을까라는 염려가 팀 내에 존재했기에 PPROF 를 이용하여 프로파일링을 진행하였고 시각화 라이브러리를 통해 문제가 될만한 leakage가 존재하지 않음을 보였습니다.

cloudchamb3r yesterday Maintainer edited ...

Original comment in Korean - Translate to English



약 6번의 샘플 코드 요청 결과입니다.
그래프에서 보여주듯, 딱히 문제가 될만한 goroutine leak은 보이지 않습니다

Dotori

WebRTC 기술을 이용하여 Full Mesh P2P 를 통해 다대다 실시간 화상 공유 기능 구현

문제 상황 및 해결 경험

미흡한 권한 체크로 인한 취약점
Dotori 프로젝트에는 채널과 토픽이라는 개념이 존재했습니다. 하나의 채널 아래에는 여러개의 토픽이 존재할 수 있고, 채널에 대한 권한이 있다면 자동적으로 하위에 있는 여러 토픽에 대한 권한을 갖습니다.

```
Channel#1
  ㄴ topic1
  ㄴ topic2
  ㄴ topic3

Channel#2
  ㄴ topic1
  ㄴ topic2
  ㄴ topic3
```

이때, 이전에 분석 및 제보하였던 모 상용 서비스의 취약점과 비슷한 취약점이 Dotori에도 존재할 수 있다고 생각하였고, 코드를 분석하여 본 결과 동일한 문제가 존재하고 있었습니다. 문제가 되는 코드는 다음과 같습니다. (설명의 편의상 핵심 로직을

제외한 부분은 모두 제거하였습니다)

```
@DeleteMapping("/channel/{channelId}/topic/{topicId}")
void removeTopic(@PathVariable int channelId, @PathVariable int topicId,
Authentication auth) {
    var name = auth.getName();
    // channel에 대한 권한 체크
    if (!channelService.hasPermission(channelId, name) {
        throw new ResponseStatusException(HttpStatus.FORBIDDEN);
    }
    // channel에 대한 권한이 있다면 topic 제거
    topicService.removeTopic(topicId);
}
```

겉보기에는 위 코드는 이상이 없어보이지만 상당한 문제가 존재합니다.
그 문제는 channel 에 대한 권한 검증은 이루어지나 topic 과 channel 이 적절한 연관 관계인지는 검증하지 않는다는 점이였습니다.
따라서 문제가 되는 부분의 코드에 연관관계에 따라서만 제거하도록 로직을 변경해 발생하는 취약점을 해결하였습니다.

```
@DeleteMapping("/channel/{channelId}/topic/{topicId}")
void removeTopic(@PathVariable int channelId, @PathVariable int topicId,
Authentication auth) {
    var name = auth.getName();
    if (!channelService.hasPermission(channelId, name) {
        throw new ResponseStatusException(HttpStatus.FORBIDDEN);
    }
    topicService.removeTopic(channelId, topicId);
}
```

사내 PACS 솔루션

전 직장에서 작업을 진행했던 프로덕트입니다. 보안 및 법적인 우려로 내부 코드에 대한 자세한 설명은 할 수 없음을 미리 양해 드립니다.

문제 상황 및 해결 경험

10년 넘게 방치 되던 메모리 누수 문제
사내 프로덕트에서 작업 중 프로그램을 켜둠에 따라 점차적으로 컴퓨터 및 소프트웨어가 느려지는 현상을 체감하였습니다.
이를 해결하기 위해 프로그램의 메모리 사용량을 측정하는 툴을 사용하여 메모리 증가가 되는 상황을 재현 및 확인하였고,
CRT 라이브러리의 `_CrtDumpMemoryLeaks()` 를 통해 메모리 누수가 일어나는 지점을 파악하고 해결하였습니다. 이를 통해 메모리 효용성의 상당한 증가를 이루었습니다.

OSS Activity

Gitea 이슈 리스트 렌더링 버그 해결

Link: #32081 Fix Bug in Issue/Pull list

Codection 프로젝트 진행 중 기반이 되는 프로젝트 gitea 의 버그를 발견하였습니다. 디버깅을 통해 컨텍스트에 올바른 데이터가 전달 되지 않는다는 사실을 파악하고 issue 를 올리고, boolean 값을 전달하도록 수정하여 pull request 를 제출하였습니다.
메인테이너의 optional 한 값들도 고려를 해보는게 어떻겠냐는 의견에 따라 optional 한 값들도 핸들링할 수 있게 수정하여 PR 을 다시 제출하여 main 브랜치에 성공적으로 머지할 수 있게 되었습니다.

HarfBuzz 한글 개선 안 제시

Link: #4850 ligatures with hangul does not working correctly

VSCode 사용 중 font ligature 옵션이 한글에 대해서 유난히 이상하게 작동하는 현상을 발견하였습니다. 이후 추가적으로 확인해본 결과 vscode 뿐만 아닌 크롬 등에도 문제가 발견하는것을 확인하였습니다. 처음엔 크로미움, 특히 렌더링 엔진 Blink 의 문제라 생각하고 장시간에 걸쳐 크로미움 빌드와 디버그를 진행하였습니다.
디버깅의 결과로 단순 렌더링 엔진의 문제가 아닌 크로미움 내부에서 사용되는 font shaping engine 인 Harfbuzz가 원인이였다는것을 파악하고, Harfbuzz 라이브러리에 대해 추가적인 디버깅을 진행하였습니다. 몇번의 삽질 끝에 hangul shaper 에 `calt` 기능이 비활성화 된 것이 문제가 되었다는 것을 파악했고, 이를 비활성화 시킨다면 한글에서만 유독 이상하게 작동하는 것을 해결 할 수 있음을 알게 되었습니다.
단, 이를 활성화하게 된다면, 일부 폰트에서 옛 한글 렌더링에 비정상적인 행동이 발생할 수 있음을 파악하고 이에 대한 내용을 추가하여 이슈를 올렸습니다. 이 이슈는 현재 Discussion 으로 전환되어 메인테이너들과 함께 해결 방안에 대한 논의를 진행하고 있습니다.

Awards

대회명	수상 내역
Hacking Camp CTF 24th	대상 (1등)