

[| 자랑 게시판 | >](#)

[스압] 옹? 누가 내 변수 바꿨어!! ~ 당신의 변수는 안녕하십니까 ~



윤 챗봇 입문자 🌱 1:1 채팅

2020.06.06. 01:12 조회 219

💬 댓글 20 URL 복사 ⋮

예아~ 가입한지 3일만에 작성한 모든 글을 인기글로 올리는 중인 윤입니다.



사실 지금 술 좀 마셨습니다. 그러니 평소보다 문맥의 앞뒤가 맞지 않을 수 있는 것을 주의하시기 바랍니다.

이 글이 작성된 계기는 [여기](#)서 시작됩니다. 이 글에서 다중 스레드와 유일 자료의 관리 방법에 대해 설명하려 했지만... 답글조차 못 다는 상황입니다.

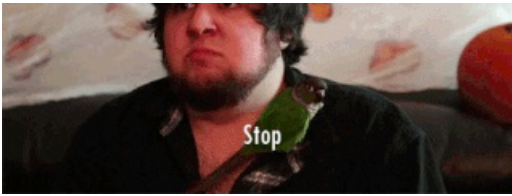
하지만 제가 새싹을 포기하지 못 해서 댓글을 포기했습니다. 대신, 그에 맞는 모듈을 하나 만들어 왔죠.

바로 유텍스라는 모듈입니다.

- 유텍스란
- 왜, 유텍스를 꼭 써야해?
- 코벨소 엔진과 **response**가 불리는 구조
- **JS**에서 구현한 방법 및 모듈 설명

□ 유텍스란





우리는 멍청하게 싱글쓰레드로 모든걸 해결하지 않습니다.
하다못해 C언어조차 쓰레드를 만듭니다.

~~그런데 JS는 싱글스레드입니다.~~

유크스는, 여러 쓰레드에서 하나의 자원 (변수 등) 을 접근하는데에 사용하는 프로그래밍 기법입니다.

□ 웹, 유크스를 꼭 써야해?

쓰레드는 여러개인데, 전역 변수는 단 하나의 메모리를 가지고 있습니다.
몇 개의 쓰레드가 살아있든, 접근할 수 있는 메모리가 하나라는 것은 결국 선착순으로 줄을 서야한다는 얘기입니다.



그딴거 모르겠고, 일단 난 변수를 수정해야겠어!

여러분, 누군가가 새치기를 하면 기분이 좋지 않아요?
어땠! 저 사람이 내가 10시간 기다린 줄을 무시하고 앞에 서줬어!! 행복해!!

그러지 않을겁니다.
기계도 그래요. 새치기를 당하면, 기분이 나빠서 아주 프로그램을 엉망진창으로 만들어 버립니다.

다음 소스를 봅시다.

```

1  var a = 5;
2
3  (function() {
4      setTimeout(() => {
5          console.log(a);
6      }, 1000);
7  })();
8
9  (function() {
10     setTimeout(() => {
11         a = 10
12     }, 500);
13 })();
14

```

원래는 다른 비동기 또는 멀티쓰레드를 예시로 들어야 하지만, 간단한 예제로 `setTimeout` 함수를 사용했습니다.

먼저 호출된 즉시 함수는, `setTimeout` 을 실행하고 `a` 를 출력합니다.

저는 `a` 가 5로 초기화 된 것을 알기에, 5가 출력됐으면 좋겠습니다.

인생이 호락호락하지 않은 것도 벅찬데, 컴퓨터 또한 호락호락하지 않습니다.

나같은 초보는 어디 서러워서 살겠나.

```

> var a = 5;

(function() {
  setTimeout(() => {
    console.log(a);
  }, 1000);
})();

(function() {
  setTimeout(() => {
    a = 10
  }, 500);
})();

```

← undefined

10

두 번째 실행된 즉시함수의 `setTimeout` 이 더 먼저 끝나서, `a` 가 10으로 바뀌고 이후 출력을 하기 때문에 10이 출력되게 됩니다.

뮤택스는, 원하는 자원을 질서있게 사용하기 위하여 사용됩니다.

□ 코볼소 엔진과 **response**가 불리는 구조



사실 저도 잘 모릅니다.

카톡 봇 시작한지 얼마나 됐다고 뉴비가 엔진의 구조까지 알고 있겠습니까.
단지 제 경험과 JS의 특성상,

각 프로젝트는 하나의 컨텍스트고,
1 **response** 호출시 쓰레드 1개가 생성된다고 보면 될 것 같습니다.

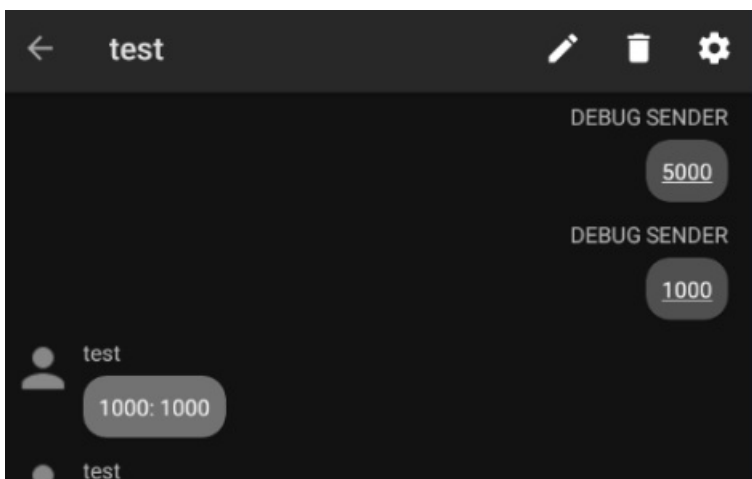
그럼 채팅 두 개가 오면 두 개의 쓰레드가 생성되겠죠?
바로 증명해 보도록 하겠습니다.

```
1 var num = 0;
2 function response(room, msg, sender, isGroupChat, replier, imageDB, packageName) {
3   p = parseInt(msg);
4   java.lang.Thread.sleep(p);
5   num = p;
6   replier.reply(msg + ":" + num.toString());
7 }
```

Colored by Color Scripter

만약, 봇 구조가 **response** 마다 컨텍스트가 새로 생성되어서, **num** 변수의 메모리가 전부 다르다면,
입력한 숫자값이 그대로 나올 것입니다.

또는 큐 형태로 제공되어서, 먼저 들어간 **response**가 종료되어야 작동하는 경우도 마찬가지로 있을 것입니다.



5000: 1000

하지만 이후 들어온 1000의 값이 찍히는 것을 볼 수 있습니다.
이로써 저의 가설이 맞았다는 것이 증명되었습니다.

□ JS에서 구현한 방법 및 모듈 설명

사실 정확한 뮤텝스를 구현한 것은 아닙니다.
진짜 뮤텝스는, Lock이 걸리면 코드 진행을 "대기" 하니까요.

제가 구현한 소스는 순차 비동기 작업 queue를 구현했다고 볼 수 있습니다.

```
1  "use strict";
2
3  function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { thro
4
5  function _defineProperties(target, props) { for (var i = 0; i < props.length; i++) { var descrip
6
7  function _createClass(Constructor, protoProps, staticProps) { if (protoProps) _defineProperties(
8
9  module.exports = /*#__PURE__*/function () {
10     function _class() {
11         _classCallCheck(this, _class);
12
13         this.mutexWorks = [];
14         this.working = false;
15     }
16
17     _createClass(_class, [{
18         key: "__polling",
19         value: function _polling() {
20             var _this = this,
21                 _this$mutexWorks;
22
23             if (this.working) {
24                 setTimeout(function () {
25                     _this.__polling();
26                 }, 50);
27                 return;
28             }
29
30             if ((this === null || this === void 0 ? void 0 : (_this$mutexWorks = this.mutexWorks) ===
31                 this.working = true;
32             var mutex = this.mutexWorks.shift();
33             mutex.work(function () {
34                 var _this$mutexWorks2;
35
36                 _this.working = false;
37
38                 if ((_this === null || _this === void 0 ? void 0 : (_this$mutexWorks2 = _this.mutexWor
39                     setTimeout(function () {
40                         _this.__polling();
41                     }, 50);
42                 }
43             }));
44         }
45     }]);
46 }
```

```

45     }
46   }, {
47     key: "Lock",
48     set: function set(work) {
49       var _this2 = this;
50
51       if (typeof work === "function") {
52         var _this$mutexWorks3;
53
54         this.mutexWorks.push({
55           work: work,
56           id: new Date().getTime()
57         });
58
59         if ((this === null || this === void 0 ? void 0 : (_this$mutexWorks3 = this.mutexWorks) =
60           setTimeout(function () {
61             _this2.__polling();
62           }, 0);
63         }
64       }
65     }
66   }]);
67
68   return _class;
69 }();
70

```

이것을 mutex.js 로 저장하고, require 로 불러들일 것입니다.

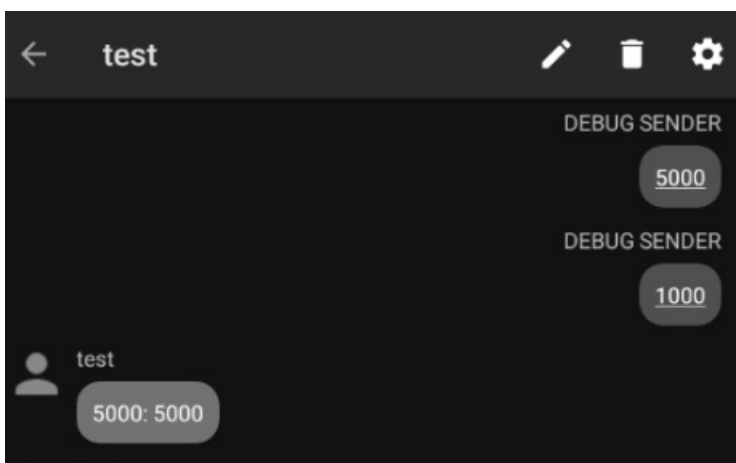
```

1  const Mutex = require('mutex');
2  const globalMutex = new Mutex();
3
4  var num = 0;
5  function response(room, msg, sender, isGroupChat, replier, imageDB, packageName) {
6    globalMutex.Lock = (Unlock) => {
7      p = parseInt(msg);
8      java.lang.Thread.sleep(p);
9      num = p;
10     replier.reply(msg + ": " + num.toString());
11     Unlock();
12   }
13 }

```

Colored by Color Scripter

Lock 에 뮤텍스를 추가한 함수는, Unlock 인자를 받아서 뮤텍스 종료시 Unlock(); 해줘야 합니다.
문법은 함수 대입이지만, class 의 set 프로퍼티이므로, 함수취급 받아서 push 합니다.







순차적으로 데이터에 접근할 수 있습니다.

읽어주셔서 감사합니다.


하트 눌러주세요.




 [윤님의 게시글 더보기 >](#)

 좋아요 11  댓글 20

 공유 |  신고

댓글 등록순 최신순 

댓글알림 



Dark Tornado

C로 컴퓨터VS사람 게임만드는 과제 할 때, 뮤텍스의 존재는 아는데 쓸 줄은 몰라서 while+sleep으로 대기시켰던건 안비밀

2020.06.06. 01:19 [답글쓰기](#)



윤 작성자

사실 뮤텍스의 이론은 "대기"이기 때문에 실질적으로 구현해 보면 while (lock); 입니다.

그것도 맞쵸!

2020.06.06. 01:20 [답글쓰기](#)



ngus0103

오오...

2020.06.06. 03:33 [답글쓰기](#)



윤 작성자



읽어주셔서 감사합니다

2020.06.06. 09:56 답글쓰기



rrrrt

밥먹고 코딩만 하시는 분인가... 공부가 끊임없이네 ㄷㄷ

2020.06.06. 05:41 답글쓰기



윤 작성자

현직 개발자입니다

2020.06.06. 09:56 답글쓰기



a8M9uQ2WRV

아침부터 질 좋은 게시글 보여서 기분이 좋습니다!

2020.06.06. 06:38 답글쓰기



윤 작성자

술 마시고 쓴 글을 맘에 들어 해 주셔서 영광입니다.

2020.06.06. 09:56 답글쓰기



도미 doami2005

레벨이 다르다

2020.06.06. 07:42 답글쓰기



윤 작성자

전 언제 도미님만큼 잘 해질까요

2020.06.06. 09:57 답글쓰기



백작선생

제목, 내용, 정리... 매번 윤님 강의 볼때마다 경이롭다는생각이..

2020.06.06. 08:11 답글쓰기



윤 작성자

이렇게 쓸 수 있기까지 많은 똥글이 있었죠

2020.06.06. 09:59 답글쓰기



성빈



2020.06.06. 08:12 답글쓰기



윤 작성자



2020.06.06. 10:00 답글쓰기



재승

ㅎㄷ

2020.06.06. 09:02 답글쓰기



윤 작성자



2020.06.06. 10:01 답글쓰기



rbos

라이노에 기본적으로 동기화함수 있지 않나요?

2020.06.06. 10:31 답글쓰기



윤 작성자

앗 아아..... 초보라서 모르겠습니다...

2020.06.06. 10:35 답글쓰기



choi0108

이렇게 해결하는 방법도 있군요
좋은글 감사합니다 ㅎㅎ

2020.06.08. 13:32 답글쓰기



윤 작성자

읽어주셔서 감사합니다.

2020.06.08. 13:33 답글쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP

'I 자랑 게시판 I' 게시판 글

이 게시판 새글 구독하기 ☐

카톡봇 버립니다. 🤖 [10]	백작선생	2020.06.06.
[미니게임] 타자 게임 🎮 [8]	재승	2020.06.06.
[스압] 앵? 누가 내 변수 바꿨어!! ~ 당신의 변수는 안녕하십니까 ~ 🤖 [20]	윤	2020.06.06.
처음 만들어본 업다운 🎮 [13]	HuOka	2020.06.06.
추억의 다마고치 🎮 [25]	deadmau5	2020.06.05.

이 카페 인기글



아키네이터

기본 개념에 대한 강좌 | 조건문과 반복문

OtakoidTony
♡0 💬6

채팅횟수 및 출석 봇 만들기 강좌

OtakoidTony
♡1 💬5

계산기 소스

JSR
♡0 💬15

꺼무위키 하다가 발견했는데, 나무뉴스라는 게 있더라고요...?

OtakoidTony
♡2 💬2

간단 계산기 입니다

getSchoolFood

도미 doami2005
♡0 💬5

(수정5)카카오오픈채팅 공식 봇관련 정

보?
이발
Hibot
카카오오픈채팅
♡0 💬21