

# Day 02 (2019-05-22)

## 1. 브라우저 아키텍처

### (1) 프로세스와 스레드

#### 프로세스

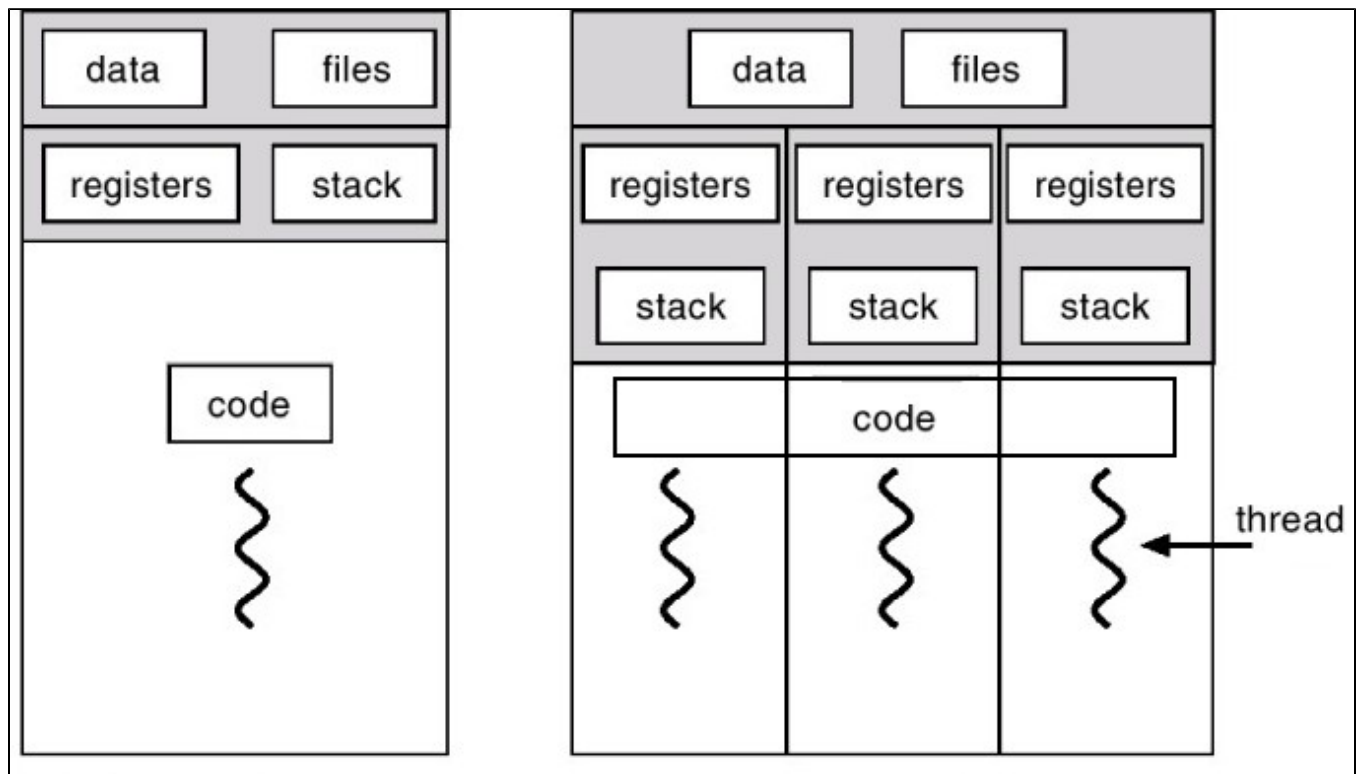
애플리케이션이 실행하는 프로그램

→ 프로세스는 각각의 독립된 메모리 영역을 OS로부터 할당 받는다. 따라서 다른 프로세스의 메모리 영역에 접근할 수 없다.

→ 최소 한개 이상의 스레드를 가지고 있다.

#### 스레드

프로세스 내부에 존재하며 프로세스로 실행되는 프로그램의 실행 흐름 일부들을 나누어 실행하는 주체



→ 한 프로세스의 모든 스레드는 고유한 스택을 가지고 Heap 영역은 공유한다.

→ 공유하는 메모리를 통해서 스레드 간의 통신이 발생한다 (동기화 필요)

*다중 프로세스 환경에서 프로세스간의 통신이 필요할 때*

*다중 스레드 환경이라면 스레드 간의 공유 메모리를 통해 통신이 가능하지만 프로세스는 그렇지 않다.*

*이때 사용하는 것이 IPC (Inter-Process Communication)이다.*

IPC를 통해 서로 다른 프로세스간의 통신으로 상호작용할 수 있으며 그 종류에는 Message passing, Shared memory 방법이있다.

## (2) 브라우저 아키텍처

(Day 01 참조)

구성요소	담당 프로세스	역할
사용자 인터페이스	플러그인 프로세스  (+브라우저 프로세스)	주소 표시줄, 북마크 막대, 뒤로 가기 버튼, 앞으로 가기 버튼 등을 제어  웹 사이트에서 사용하는 플러그인(예: Flash)을 제어한다.
브라우저 엔진	브라우저 프로세스	네트워크 요청이나 파일 접근과 같이 눈에 보이지는 않지만 권한이 필요한 부분도 처리한다.
렌더링 엔진	렌더러 프로세스	탭 안에서 웹 사이트가 표시되는 부분의 모든 것을 제어한다.
	GPU 프로세스	GPU 작업을 다른 프로세스와 격리해서 처리한다. GPU는 여러 애플리케이션의 요청을 처리하고 같은 화면에 요청받은 내용을 그리기 때문에 GPU 프로세스는 별도 프로세스로 분리되어 있다
통신	네트워크 프로세스	네트워크 요청에 따라 실제 통신을 처리한다.  요청에 대한 응답을 받은 후 필요로하는 프로세스에게 알린다.
UI 백엔드	UI 프로세스	콤보 박스와 창 같은 기본적인 장치를 그리는데 사용
자료 저장소	스토리지 프로세스	쿠키, HTML5 스토리지 등 모든 종류의 자원을 저장

## 프로세스 모델

### Single process

각 브라우저의 구성 요소들을 하나의 프로세스에서 실행  
하나의 프로세스 내의 여러 스레드가 각각의 구성요소를 담당

→ 두 개 이상의 tab을 띄웠을 때 하나의 tab에서 응답없음이 발생하면?

## 응답 없는 페이지

페이지가 응답할 때까지 기다리거나 페이지를 종료할 수 있습니다.

제목 없음

대기

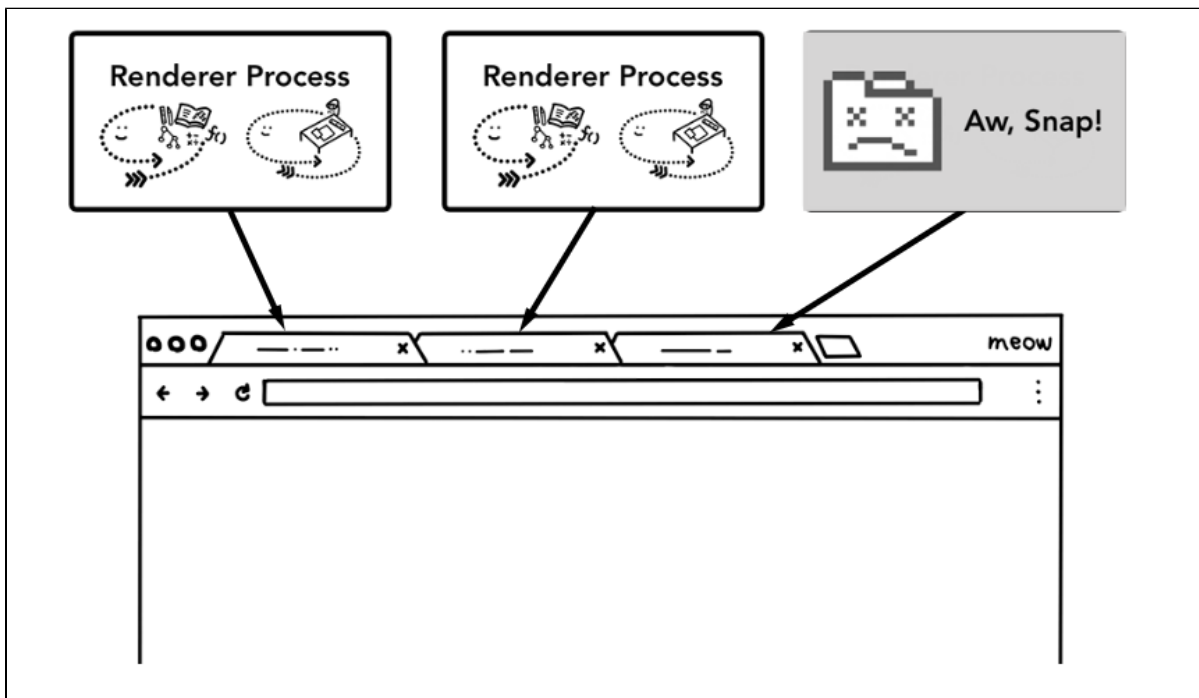
페이지 나가기

### Multi process

#### 1) Process per tab

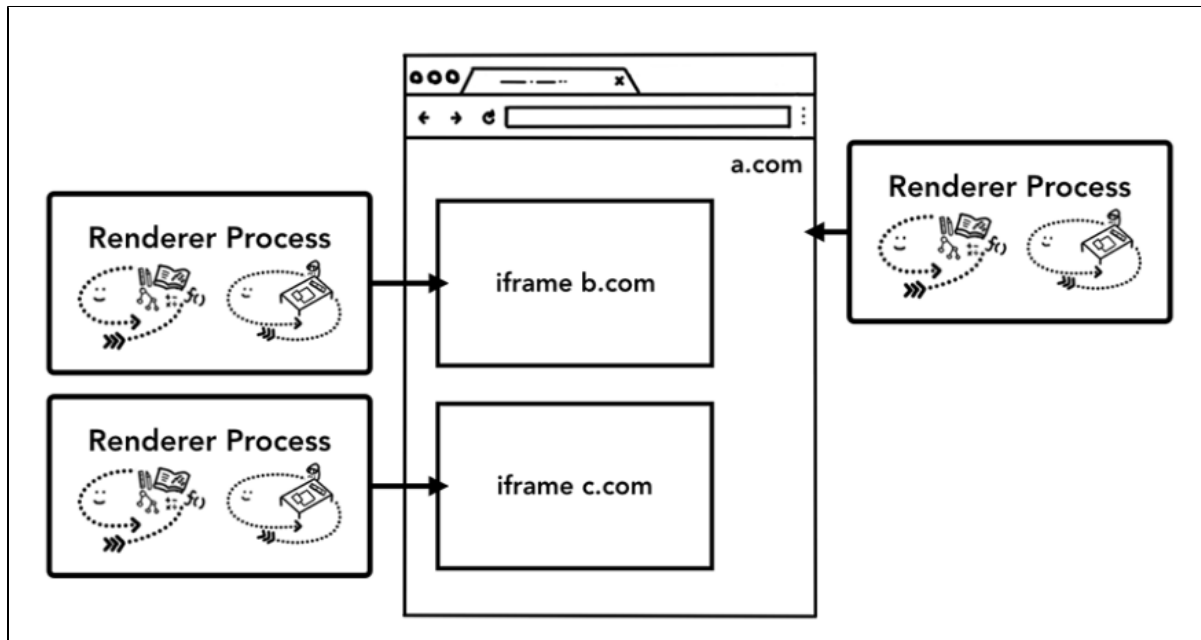
렌더러 프로세스를 각 탭마다 할당

→ 하나의 tab에서 응답없음이 발생하더라도 다른 tab에는 영향 없음



#### 2) Process per site instance

렌더러 프로세스를 사이트의 각 인스턴스마다 할당



→ 사이트 격리를 통해 동일 프로세스에서 여러 사이트간의 메모리를 공유할 수 있는 문제를 회피 **동일 출처 정책(Same-origin policy)**을 지킴

→ Meltdown과 Spectre 문제를 해결 (**Meltdown, Spectre**)

→ 메모리 부하가 매우 심함. 탭을 많이 띄울수록 메모리와 CPU에 과부하가 걸림

### 3) Process per site

렌더러 프로세스를 각 사이트 마다 할당

→ 서로 다른 사이트 별로 분리하지만 동일한 사이트의 모든 인스턴스는 동일 프로세스로 그룹화 함

→ 2번 모델에서 메모리 등의 리소스가 부족하다면 3번으로 전환

※ **멜트다운과 스펙터, CPU 보안이 통째로 무너지다**

CPU의 버그를 통해 해커가 사용자의 암호와 같은 중요 데이터를 훔칠 수 있음.

또한 사용자가 실행중인(CPU가 처리하고 있는) 응용프로그램의 데이터를 다 훔쳐볼 수 있는 버그

현재 최신 버전의 Chrome에선 Process per site instance를 채택하고 있으며 Chrome에서 사용중인 모든 프로세스는 아래와 같이 확인 가능

작업 관리자 - Chrome

작업	메모리 사용량	CPU	네트워크	프로세스 ID
브라우저	206,464K	7.8	0	25340
GPU 프로세스	346,848K	6.2	0	15848
유틸리티: Network Service	29,892K	0.0	0	21196
탭: KT bizmeka EZ	70,704K	1.6	0	18980
탭: Day 02 (2019-05-22) - [TEAM] 사업 3팀 - Confluence	203,728K	7.8	1,144B/s	8512
탭: Day 01 (2019-05-08) - [TEAM] 사업 3팀 - Confluence				
22 탭: CRScube Inc. - 캘린더 - 2019년 5월	108,544K	0.0	0	23868
서브프레임: https://accounts.google.com/	31,664K	0.0	0	12888
서브프레임: https://accounts.google.com/				
탭: cubeCTMS	92,524K	0.0	0	16208
탭: OKKY - All That Developer	55,512K	0.0	0	10260
서브프레임: https://doubleclick.net/	43,132K	0.0	0	3244
서브프레임: https://doubleclick.net/				
서브프레임: https://dable.io/	34,944K	0.0	0	24180
서브프레임: https://2mdn.net/	35,996K	0.0	0	15356
서브프레임: https://googlesyndication.com/	31,272K	0.0	0	14932
탭: Elasticsearch Reference [6.6]   Elastic	107,712K	0.0	0	14772
서브프레임: https://marketo.com/	25,372K	0.0	0	13256
서브프레임: https://olark.com/	42,836K	0.0	0	16608
탭: 10+ Best SMTP Service Providers & Servers 2019 - Woofresh	58,324K	0.0	0	24156
D2 탭: 최신 브라우저의 내부 살펴보기 1 - CPU, GPU, 메모리 그리고 다중 프로세스 아키텍처	98,660K	0.0	0	9836
서브프레임: https://naver.com/				
서브프레임: https://facebook.com/	57,604K	0.0	0	11436
서브프레임: https://facebook.com/				
서브프레임: https://facebook.com/				

프로세스 종료

### (3) 메모리 절약

Chrome은 에버그린 브라우저에 속한다. 따라서 새로운 프로세스 모델이 개발 될 때마다 자동으로 적용이 되는데 사양이 낮은 환경의 컴퓨터에선 Process per site instance와 같은 프로세스 모델은 메모리에 큰 부하를 일으킬 것이다.

→ 프로세스들은 전용 메모리 공간을 사용하므로 V8 엔진과 같이 공통 영역들을 복사해서 따로 가진다.

결과적으로 사용자에게 좋지않은 경험을 남길 수 있는데 Chrome은 메모리 절약과 관련된 전략도 제공한다.

성능이 좋거나 하드웨어 자원이 여유로운 환경에서 실행 중일 때는 Process per site instance 전략을 기본으로 취한다.

반면 성능이 좋지 않은 환경에서는 여러 프로세스를 의미있는 그룹 단위로 묶어서 처리하도록 동작한다.

이것은 성능이 좋은 하드웨어에서 많은 수의 tab을 생성하여 점차 하드웨어 자원이 여유롭지 않은 환경이 될 때에도 동작한다.

사용자가 업데이트 하지 않아도 **자동으로 업데이트 되는** 브라우저를 의미한다

→ 별도의 **재설치가 요구되지 않아** 사용자에게 편의성을 제공하고 빠르게 발전하는 Web 기술을 쉽게 대응할 수 있다는 장점이 있다.

→ IE를 제외한 대부분의 브라우저가 에버그린 브라우저에 속한다

## 참고자료

### 스레드와 프로세스

- <https://jhnyang.tistory.com/24>
- <https://jwprogramming.tistory.com/54>

### 브라우저 아키텍처

- <https://www.chromium.org/developers/design-documents/process-models>
- <https://d2.naver.com/helloworld/2922312>
- <https://developers.google.com/web/updates/2018/02/meltdown-spectre>
- <https://it.donga.com/27265/>