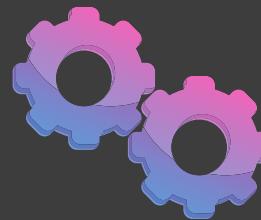


# Operating System Principles

## POWER BROWSER MOZILLA FIREFOX

Presentation by  
SystemsExperts





# Operating System Principles

## MEET THE TEAM

Saalim Shadman



Devin Amalean

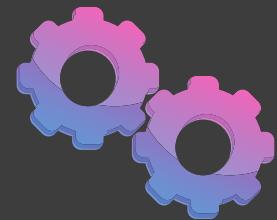


Ryan Le-Nguyen



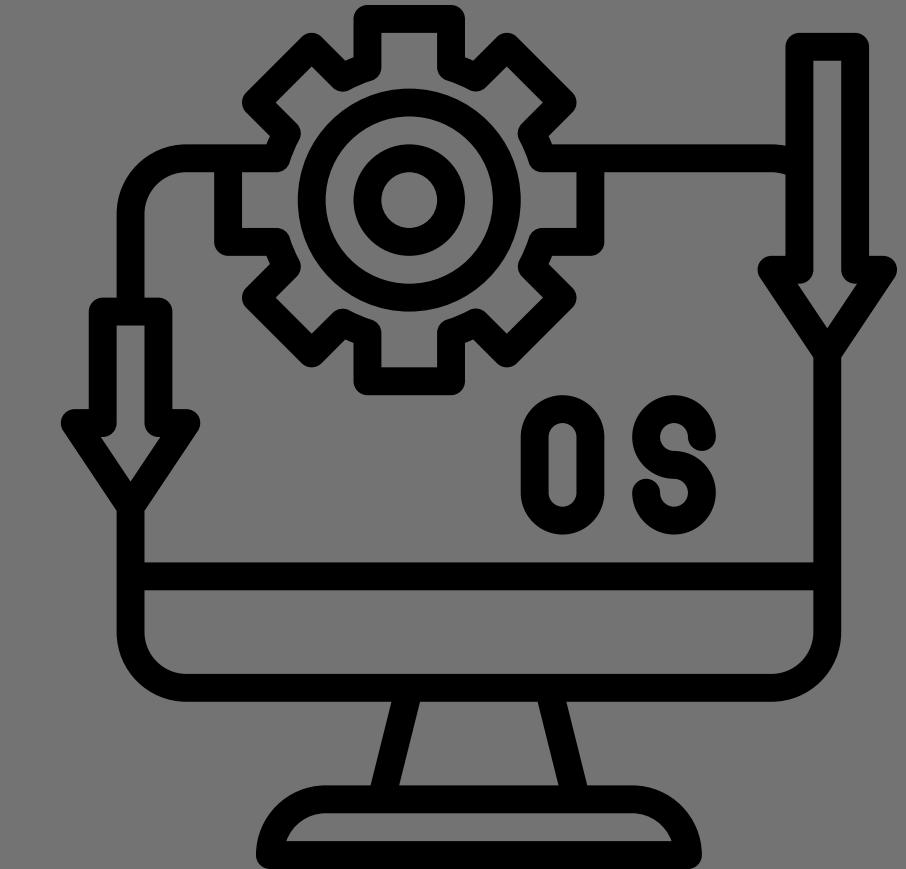
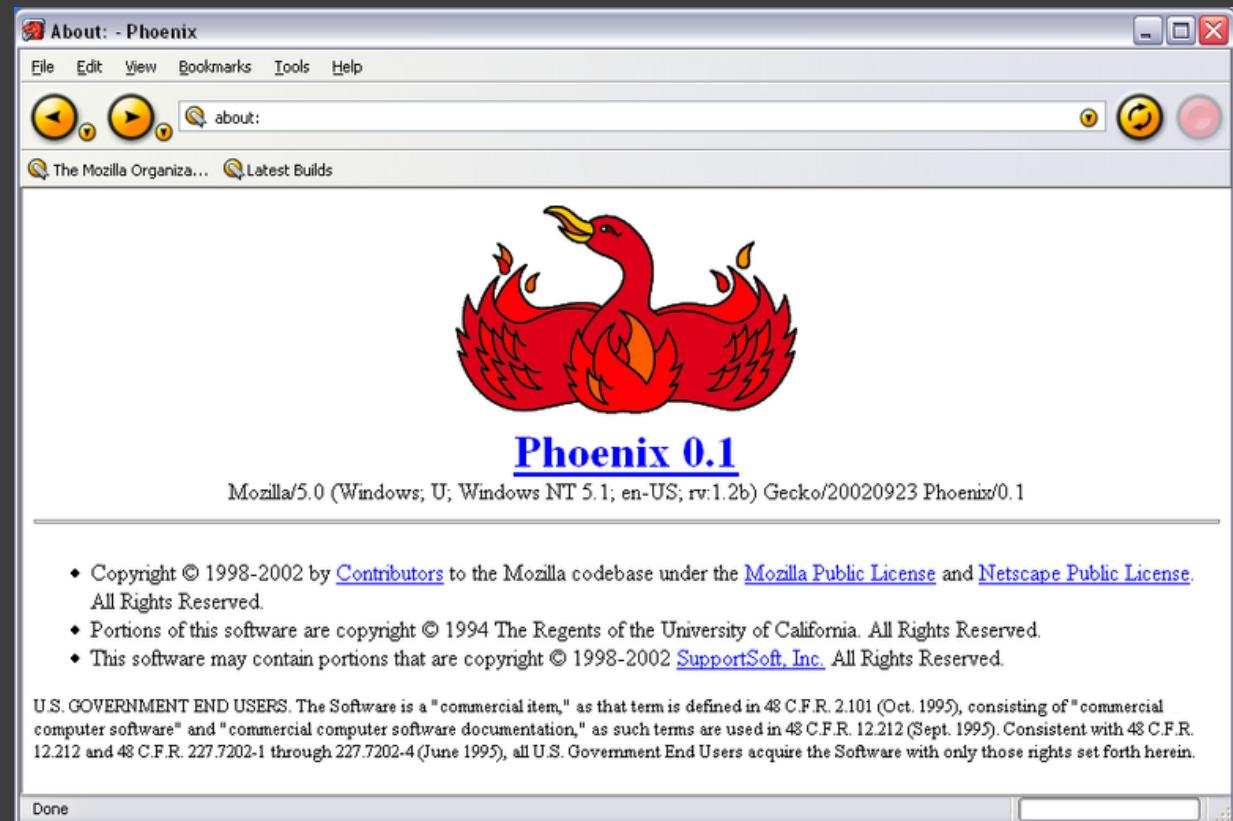
Kowsar Rahman

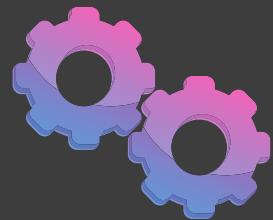




# Operating System Principles

## INTRODUCTION





# Operating System Principles

## INTRODUCTION

**moz://a** Firefox Browsers Products Who We Are Innovation Download Firefox

# 48.0

Firefox Release August 2, 2016

**Version 48.0, first offered to Release channel users on August 2, 2016**  
We'd also like to extend a special thank you to all of the [new Mozillians](#) who contributed to this release of Firefox!

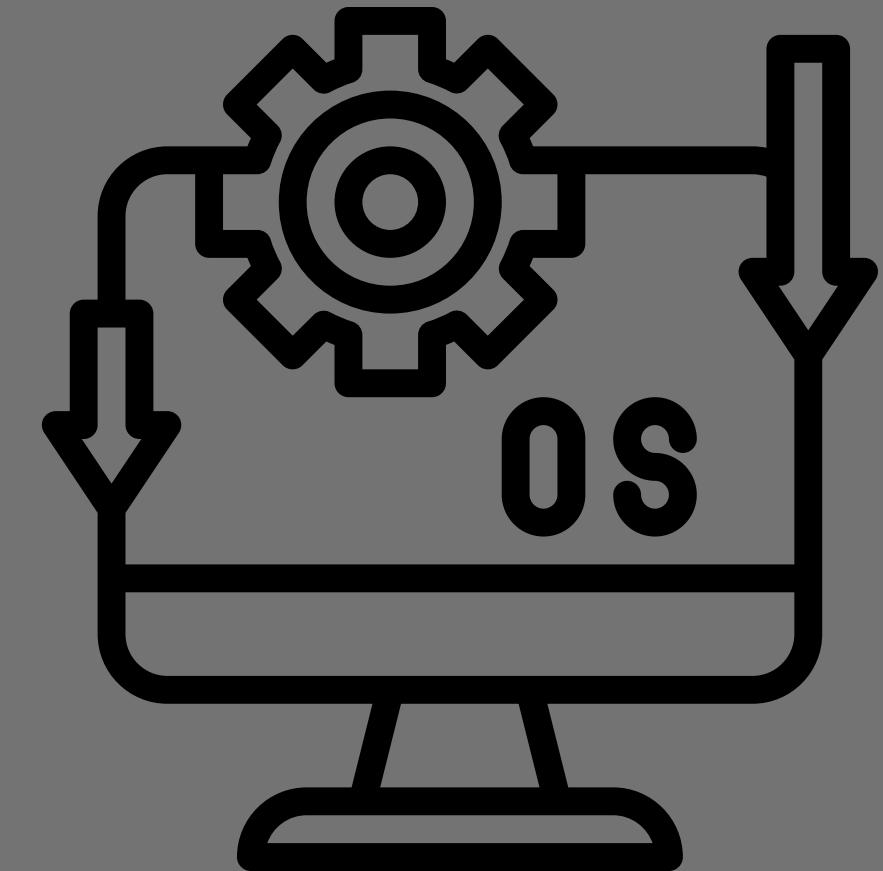
---

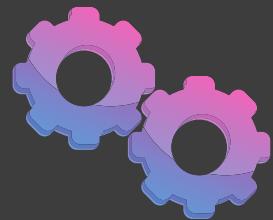
**New** [Process separation](#) (e10s) is enabled for some of you. Like it? Let us know and we'll roll it out to more

Roar for moar [protection against harmful downloads](#)! We've got your back

---

Add-ons that have [not been verified and signed](#) by Mozilla will not load





# Operating System Principles

## INTRODUCTION

**54.0**

Firefox Release  
June 13, 2017

### Version 54.0, first offered to Release channel users on June 13, 2017

Today's release is the first to run Firefox using multiple operating system processes for web page content, making Firefox faster and more stable than ever. Learn more about how multiple processes strike a "just right" balance between performance and memory use on [the Mozilla Blog](#). Dive into the [details \(including performance benchmarks\) on Medium](#).

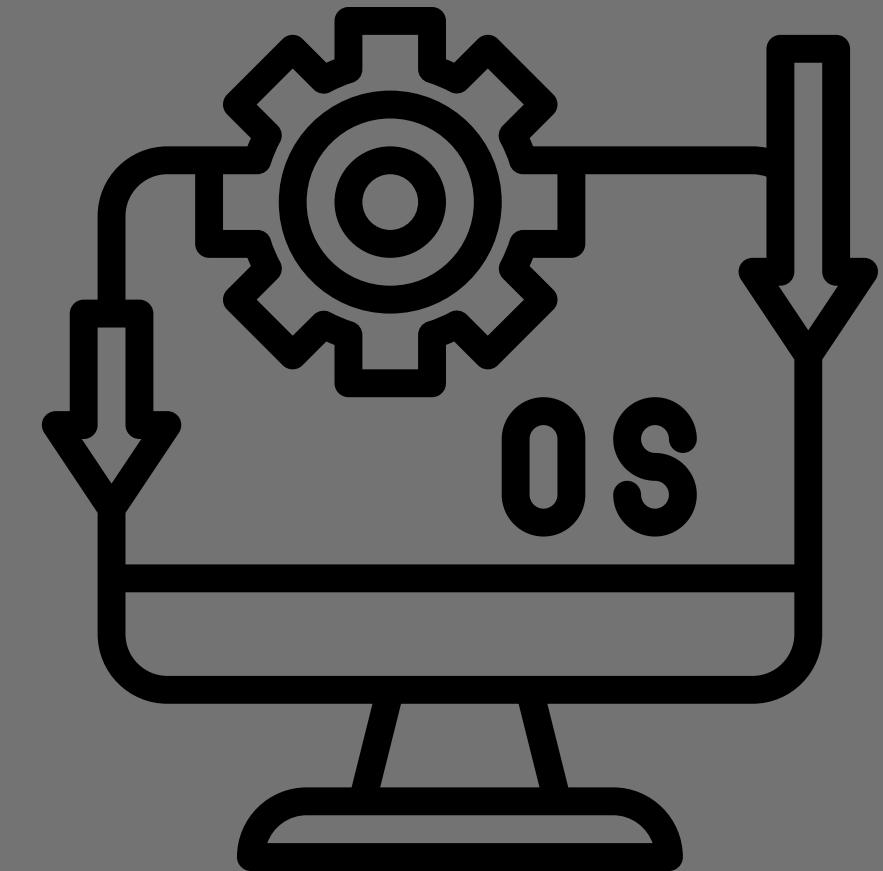
We'd also like to extend a special thank you to all of the [new Mozillians](#) who contributed to this release of Firefox!

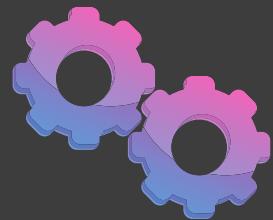
★ New

Added Burmese (my) locale

Added support for [multiple content processes](#) (e10s-multi)

Simplified the download button and download status panel





# Operating System Principles

## INTRODUCTION

**105.0**

Firefox Release  
September 20, 2022

Version 105.0, first offered to Release channel users on September 20, 2022

★ New

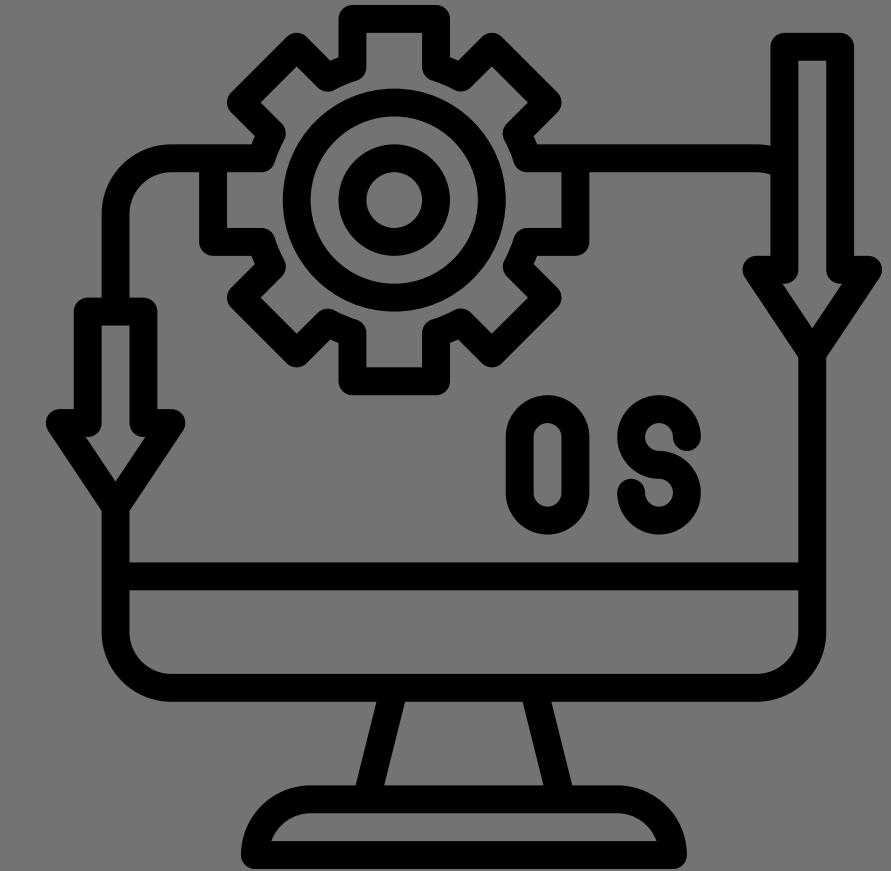
Added an option to print only the current page from the print preview dialog.

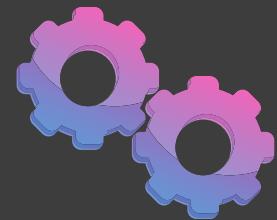
Firefox now supports partitioned service workers in third-party contexts. You can register service workers in a third-party iframe and it will be partitioned under the top-level domain.

Swipe to navigate (two fingers on a touchpad swiped left or right to perform history back or forward) on Windows is now enabled.

Firefox is now compliant with the User Timing L3 specification, which adds additional optional arguments to the `performance.mark` and `performance.measure` methods to provide custom start times, end times, duration, and attached details.

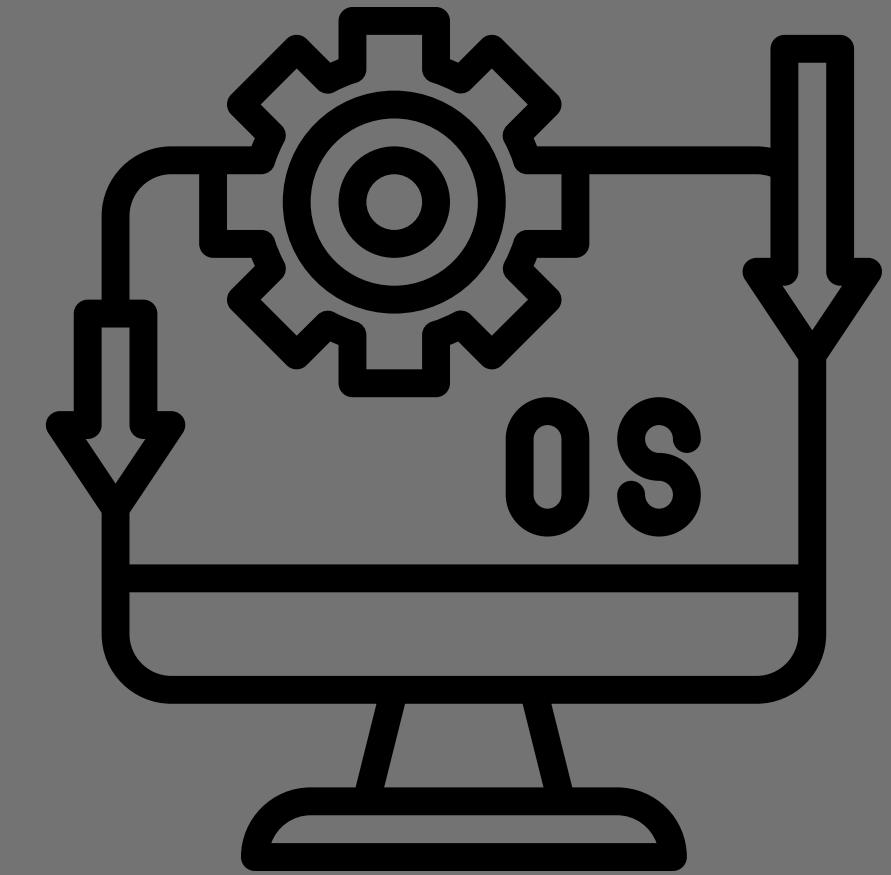
Searching in large lists for individual items is now 2x faster. This performance enhancement replaces `array.includes` and `array.indexOf` with an optimized SIMD version.

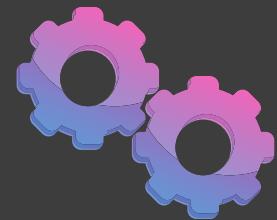




## INTRODUCTION

- Multi-Processing
- Multi-Threading
- Semaphores
- Mutex Implementations



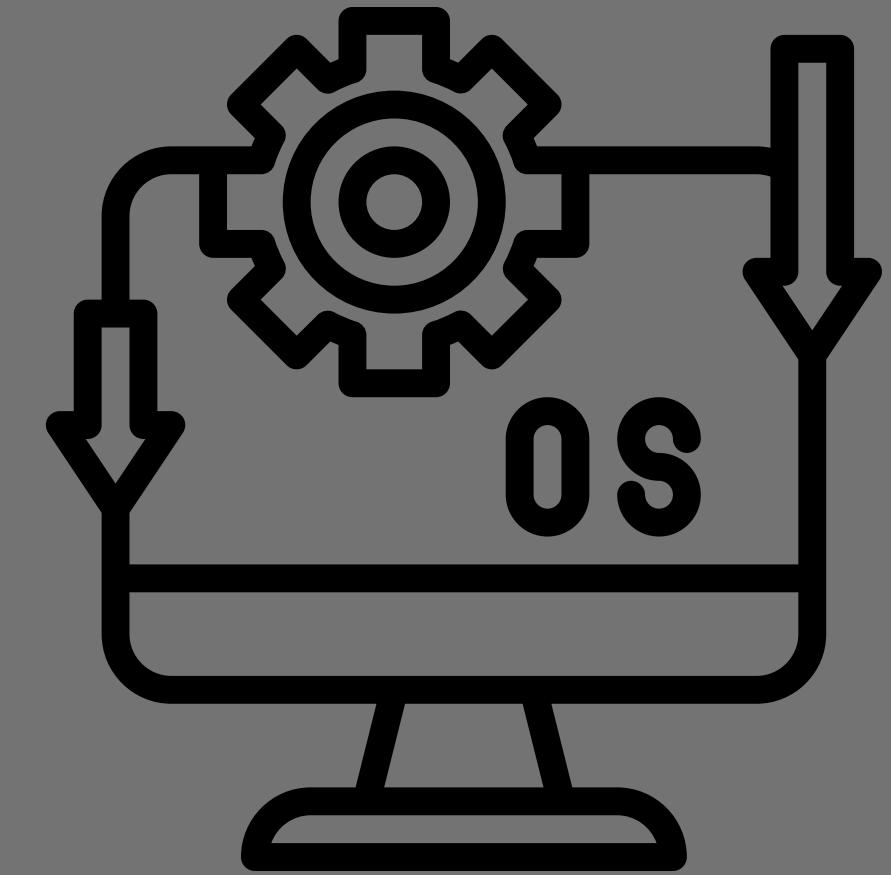


## INTRODUCTION

**Ryan:** Project build and environment

**Kowsar:** Code demo and OSP relevance

**Shadman:** Challenges faced



# PLATFORM AND BUILD ENVIRONMENT



# THE BUILD SYSTEM: MECHA

## 3 Components

- Core
- Commands
- Driver

› ./mach --help  
usage: mach [global arguments] command [command arguments]

Global Arguments:

-v, --verbose	Print verbose output.
-l FILENAME, --log-file FILENAME	Filename to write log data to.
--log-interval	Prefix log line with interval from last message rather than relative time. Note that this is NOT execution time if there are parallel
--no-interactive	Automatically selects the default option on any interactive prompts. If the output is not a terminal, then --no-interactive is assumed
--log-no-times	Do not prefix log lines with times. By default, mach will prefix each output line with the time since command start.
-h, --help	Show this help message.
--debug-command	Start a Python debugger when command is dispatched.
--profile-command	Capture a Python profile of the mach process as command is dispatched.
--settings FILENAME	Path to settings file.

Build Commands:

Interact with the build system

build	Build the tree.
build-backend	Generate a backend used to build the tree.
cargo	Invoke cargo in useful ways.
clobber	Clobber the tree (delete the object directory).
compare-locales	Run source checks on a localization.
configure	Configure the tree (run configure and config.status).
hazards	Commands for running the static analysis for GC rooting hazards

Post-build Commands:

Common actions performed after completing a build.

artifact	Use pre-built artifacts to build Firefox.
buildsymbols	Produce a package of Breakpad-format symbols.
devtools-css-db	Rebuild the devtool's static css properties database.
geckodriver	Run the WebDriver implementation for Gecko.
package	Package the built product for distribution as an APK, DMG, etc.
package-multi-locale	Package a multi-locale version of the built product for distribution as an APK, DMG, etc.
resource-usage	Show information about system resource usage for a build.
show-log	Display mach logs
warnings-list	Show a list of compiler warnings.
warnings-summary	Show a summary of compiler warnings.

Testing:

Run tests.

# MECURIAL

## Version Control System

- Alternative to git
- Source code hosting



mercurial

# BUID PROCESS

## INSTALL MECURIAL

```
python3 -m pip install --  
user mercurial
```

## CLONE

```
curl https://hg.mozilla.org/mozilla-central/raw-  
file/default/python/mozboot/bin/bootstrap.py -O  
python3 bootstrap.py
```

## BUILD

```
cd mozilla-unified  
.mach build  
.mach run
```

# DEMO

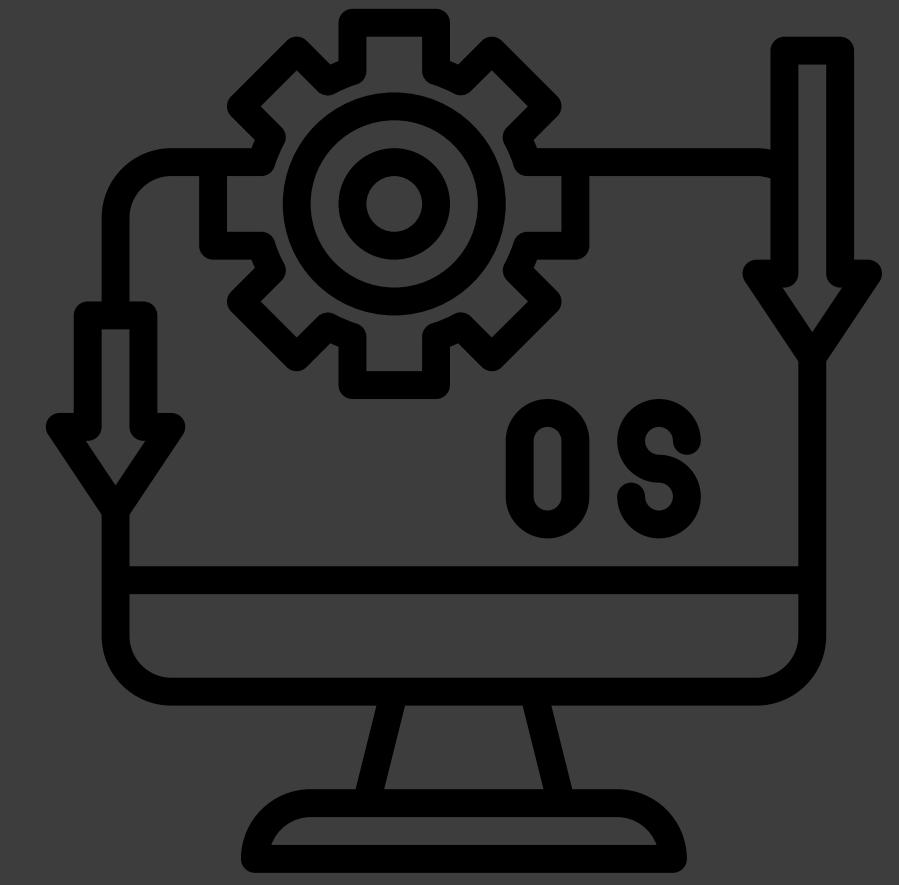
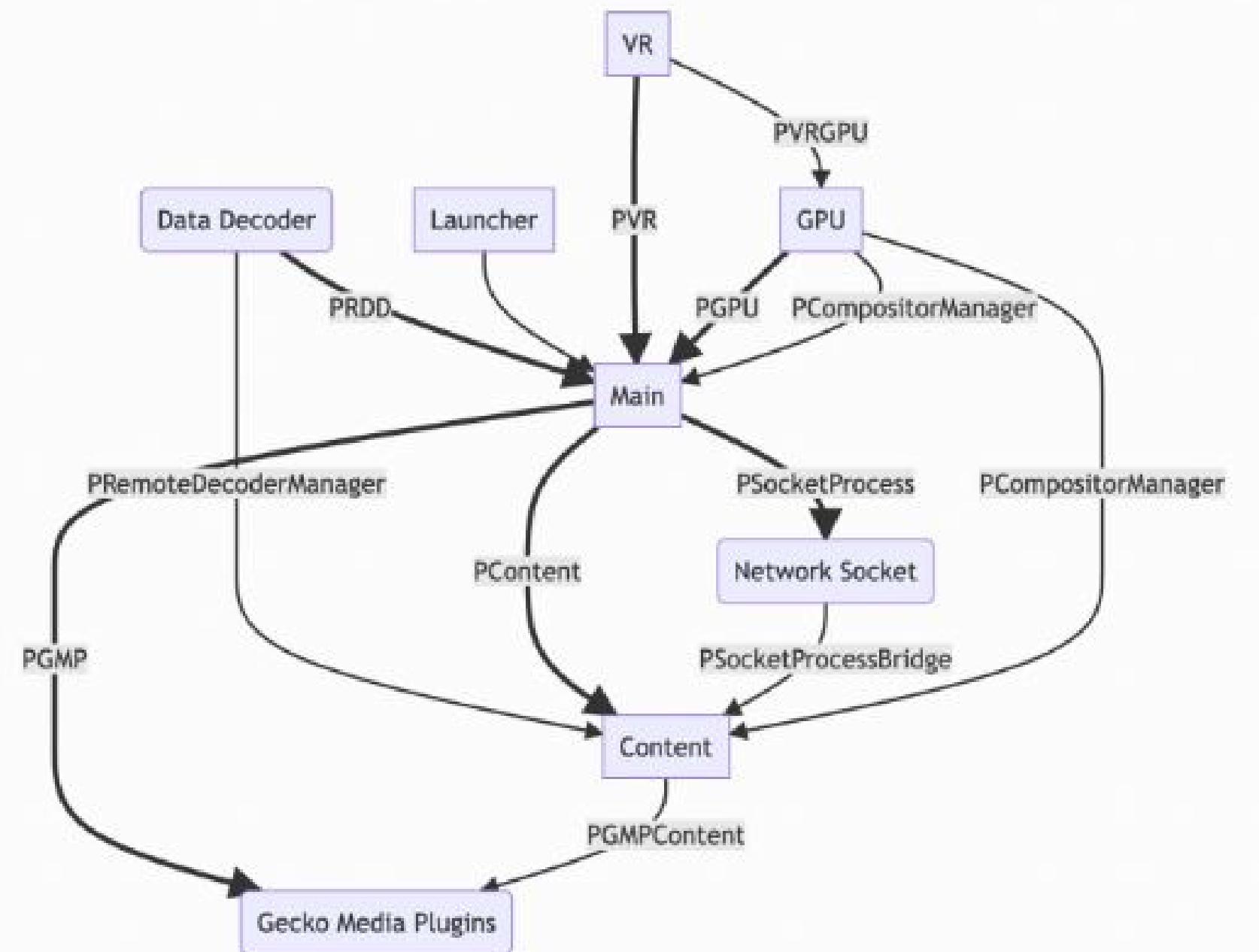
Activities Terminal Oct 20 22:59

```
kowsar@kowsar-VirtualBox: ~/mozilla-unified
gnu/dist/bin/browser/features/webcompat-reporter@mozilla.org: Kept 1 existing;
Added/updated 2; Removed 0 files and 0 directories.
1:42.07 Elapsed: 0.61s; From /home/kowsar/mozilla-unified/obj-x86_64-pc-linux-gnu/dist/bin/browser/localization: Kept 98 existing; Added/updated 0; Removed 0 files and 0 directories.
1:42.34 Elapsed: 0.13s; From /home/kowsar/mozilla-unified/obj-x86_64-pc-linux-gnu/dist/bin/gmp-clearkey/0.1: Kept 0 existing; Added/updated 1; Removed 0 files and 0 directories.
1:42.53 Elapsed: 0.02s; From /home/kowsar/mozilla-unified/obj-x86_64-pc-linux-gnu/dist/bin/gmp-fake/1.0: Kept 1 existing; Added/updated 0; Removed 0 files and 0 directories.
1:42.68 Elapsed: 0.02s; From /home/kowsar/mozilla-unified/obj-x86_64-pc-linux-gnu/dist/bin/gmp-fakeopenh264/1.0: Kept 1 existing; Added/updated 0; Removed 0 files and 0 directories.
1:42.94 Elapsed: 0.14s; From /home/kowsar/mozilla-unified/obj-x86_64-pc-linux-gnu/dist/bin/localization: Kept 67 existing; Added/updated 0; Removed 0 files and 0 directories.
2:04.24 Packaging [REDACTED] specialpowers@mozilla.org.xpi...
2:04.61 Packaging mozscreenshots@mozilla.org.xpi...
2:04.76 0 compiler warnings present.
2:04.98 Overall system resources - Wall time: 115s; CPU: 57%; Read bytes: 8388
33152; Write bytes: 233324544; Read time: 218309; Write time: 10508
2:04.98 Swap in/out (MB): 0.18359375/207.3984375
To view resource usage of the build, run |mach resource-usage|.
2:05.08 Your build was successful!
To take your build for a test drive, run: |mach run|
For more information on what to do now, see https://firefox-source-docs.mozilla.org/setup/contributing_code.html
kowsar@kowsar-VirtualBox:~/mozilla-unified$
```

# PROCESS ARCHITECTURE

## Firefox Process Hierarchy

This diagram shows the primary process types in Firefox.



# PROCESS

```
bool LinuxCoreDumper::BuildProcPath(char* path, pid_t pid,
                                     const char* node) const {
    if (!path || !node)
        return false;

    size_t node_len = my_strlen(node);
    if (node_len == 0)
        return false;

    size_t procfs_path_len = my_strlen(procfs_path_);
    size_t total_length = procfs_path_len + 1 + node_len;
    if (total_length >= NAME_MAX)
        return false;

    memcpy(path, procfs_path_, procfs_path_len);
    path[procfs_path_len] = '/';
    memcpy(path + procfs_path_len + 1, node, node_len);
    path[total_length] = '\0';
    return true;
}
```



Used in crash reporting

Works for all three platforms

# THREADS

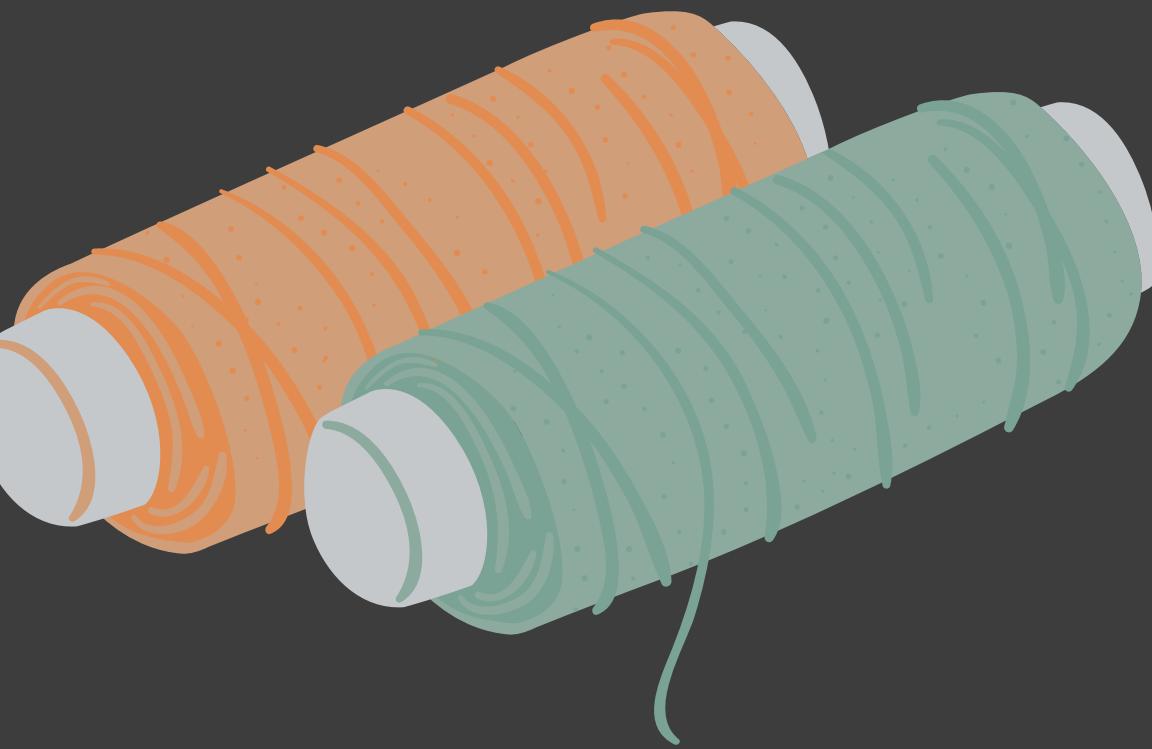
```
void CrashGenerator::CreateThreadsInChildProcess(unsigned num_threads) {
    *GetThreadIdPointer(0) = getpid();

    if (num_threads <= 1)
        return;

    // This method does not clean up any pthread resource, as the process
    // is expected to be killed anyway.
    ThreadData* thread_data = new ThreadData[num_threads];

    // Create detached threads so that we do not worry about pthread_join()
    // later being called or not.
    pthread_attr_t thread_attributes;
    if (pthread_attr_init(&thread_attributes) != 0 ||
        pthread_attr_setdetachstate(&thread_attributes,
                                    PTHREAD_CREATE_DETACHED) != 0) {
        fprintf(stderr, "CrashGenerator: Failed to initialize thread attribute\n");
        exit(1);
    }

    pthread_barrier_t thread_barrier;
    if (pthread_barrier_init(&thread_barrier, NULL, num_threads) != 0) {
        fprintf(stderr, "CrashGenerator: Failed to initialize thread barrier\n");
        exit(1);
    }
}
```



Creating threads in child processes

# MUTEX LOCKS

```
class RTC_LOCKABLE RawMutexTryLocker {
public:
    explicit RawMutexTryLocker(Mutex& mutex) : mutex_(mutex) {}
    void Lock() RTC_EXCLUSIVE_LOCK_FUNCTION() {
        while (!mutex_.TryLock()) {
            YieldCurrentThread();
        }
    }
    void Unlock() RTC_UNLOCK_FUNCTION() { mutex_.Unlock(); }

private:
    Mutex& mutex_;
};
```



Used in RTC part of the Web Browser

Used for video and audio streaming

Prevents race conditions

# SEMAPHORES

```
bool CrossProcessSemaphore::Wait(const Maybe<TimeDuration>& awaitTime) {
    MOZ_ASSERT(mSemaphore, "Improper construction of semaphore.");
    HRESULT hr = ::WaitForSingleObject(
        mSemaphore, awaitTime.isSome() ? awaitTime->ToMilliseconds() :
    return hr == WAIT_OBJECT_0;
}

void CrossProcessSemaphore::Signal() {
    MOZ_ASSERT(mSemaphore, "Improper construction of semaphore.");
    ::ReleaseSemaphore(mSemaphore, 1, nullptr);
}
```

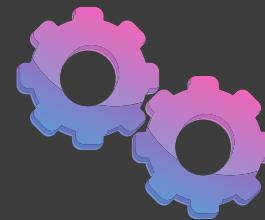


010  
010  
101  
0101001  
1111101  
0101010  
0100100101001  
1110101111101  
0100100101001

Prevents the critical section problem

Helps achieve process synchronization

Two operations: Signal() and Wait()



## DIFFICULTIES & ISSUES



1

### **Excessive Memory Space Use**

The size of the project was huge, hence cloning the entire repository consumed 8.47 GB of memory in our local environments

2

### **Bad Time Complexity**

Since the project was large, cloning the project took roughly two hours which we thought was a long time.

3

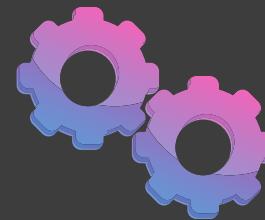
### **Code Comprehension Challenges**

Going through the code was challenging as there was a lot going on which we were not fully able to understand.

4

### **Understanding New Version Control Tool**

Understanding and using a new version control tool called Hg was something we were unfamiliar



## PROJECT CITATIONS



- Dan Callahan, Belén Albeza Posted on June 13. "Firefox 54: E10s-Multi, WebExtension Apis, CSS Clip-Path – Mozilla Hacks – the Web Developer Blog." Mozilla Hacks – the Web Developer Blog, 13 June 2017, <https://hacks.mozilla.org/2017/06/firefox-54-e10s-webextension-apis-css-clip-path/>.
- Wikipedia Contributors. "Firefox." Wikipedia, Wikimedia Foundation, 7 Dec. 2019, [en.wikipedia.org/wiki/Firefox](https://en.wikipedia.org/wiki/Firefox).
- "Electrolysis – MozillaWiki." Wiki.mozilla.org, wiki.mozilla.org/Electrolysis. Accessed 20 Oct. 2022.
- "Mozilla Firefox Release Notes." Mozilla, [www.mozilla.org/en-US/firefox/releases/](https://www.mozilla.org/en-US/firefox/releases/).
- "History of the Mozilla Project." Mozilla, [www.mozilla.org/en-US/about/history/#:~:text=The%20Mozilla%20project%20was%20created](https://www.mozilla.org/en-US/about/history/#:~:text=The%20Mozilla%20project%20was%20created).
- "Firefox 48.0, See All New Features, Updates and Fixes." Mozilla, [www.mozilla.org/en-US/firefox/48.0/releasenotes/](https://www.mozilla.org/en-US/firefox/48.0/releasenotes/). Accessed 20 Oct. 2022.
- "Gecko Processes — Firefox Source Docs Documentation." Firefox-Source-Docs.mozilla.org, [firefox-source-docs.mozilla.org/ipc/processes.html#creating-the-new-process](https://firefox-source-docs.mozilla.org/ipc/processes.html#creating-the-new-process). Accessed 20 Oct. 2022.