

# Basecamp Introduction Tutorial

## Objectives

- Introduce Basecamp features so users can quickly be productive
- Provide guidance on what to do next based on your goals

## Lesson 1 Objectives

- Describe Basecamp's objectives, components, and terminology

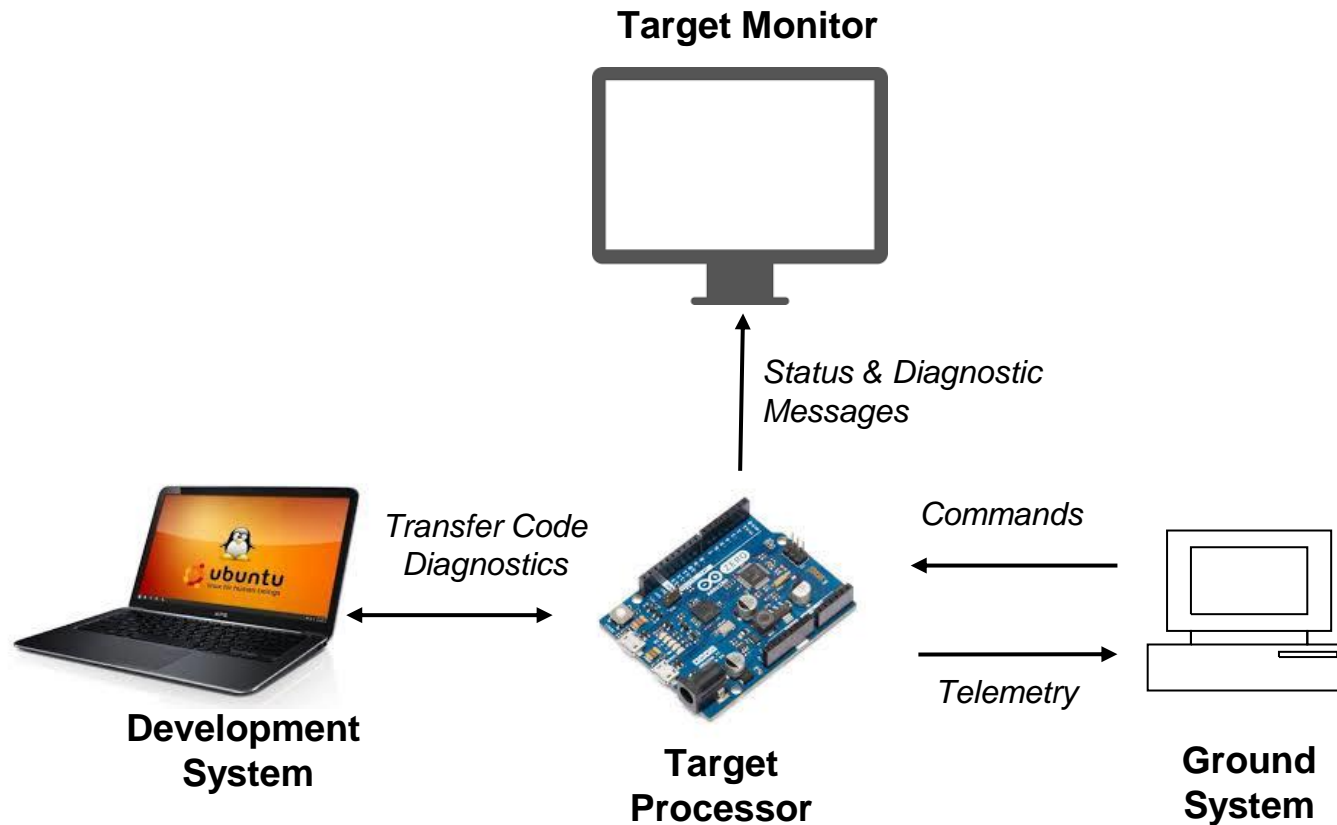
## Notes

- This tutorial describes what is available. Other tutorials and documents provide details on workflows and how to accomplish goals.

# Why Basecamp?

- Basecamp provides a cFS architectural framework, build/runtime tools, and a lightweight GUI that simplify creating, integrating, testing, and deploying cFS applications
- Provides a foundation for users and educators to create cFS-based projects
- Command and telemetry routing design supports interfacing to external systems
- Supports the following application activities
  - Learn the cFS application architectural model
  - Learn Basecamp's application framework (heritage from OpenSatKit)
  - Develop new applications
  - Download apps from the github cfs-apps repositories
  - Integrate apps into Basecamp's cFS target
  - [future] Learn Basecamp's application packaging specification
  - [future] Certify new apps comply with Basecamp's packaging specification
- Not intended to be a fully functional ground system
- Basic command and telemetry GUI/script interfaces provide app development and runtime support
- [future]cFS build tools can be customized to generate command and telemetry definitions for different ground systems

# Embedded Flight Systems Context



## Target Processor

- A processor that runs the cFS target image

## Development System

- Used to build and transfer the cFS target image to the target processor
- Requires a 'cross compiler' if the target process is different than the development system
- May include runtime diagnostic tools

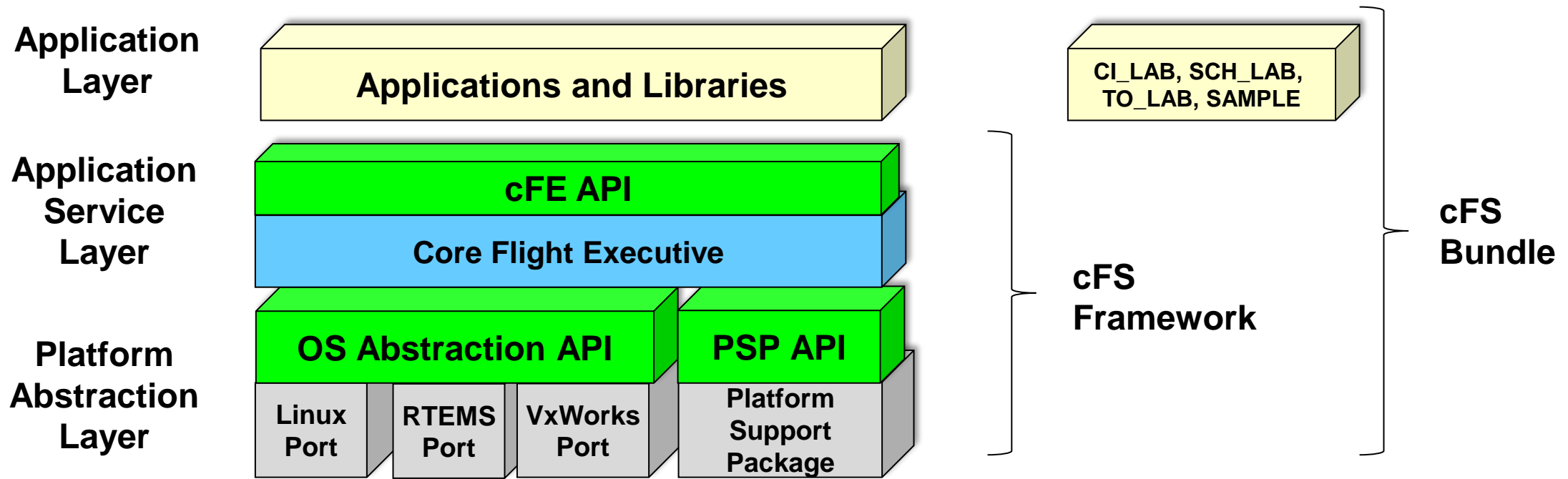
## Target Monitor

- A common diagnostic tool used to help verify the embedded system is operating correctly
- Often a monitor connected over a serial port

## Ground System

- An application that sends command messages to the target and receives telemetry messages from the target
- The command & telemetry communications link may vary between test configurations and operations

# core Flight System Context



- Platform Abstraction Layer ports to different operating systems (OS) / processor combinations
  - Contains the Operating System Abstraction Layer (OSAL) and the Platform Support Package (PSP)
- Application and libraries that only use the cFS APIs are portable across platforms
- The cFS Framework managed by NASA at <https://github.com/nasa/cFE/>
- The cFS bundle provides a starter system with a minimal runtime app suite, <https://github.com/nasa/cFS>

# Basecamp Ecosystem


**Learning Resources**



<https://www.youtube.com/channel/UC2wfvAlkrrgyC4ITwL3zokg>




<https://openmissionstack.com/>

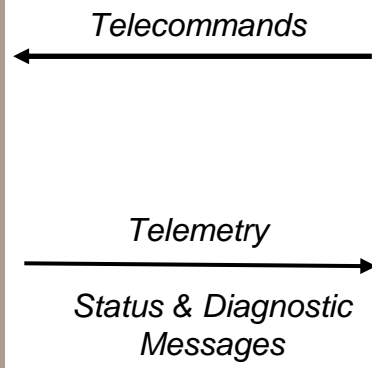


**cFS App Repos**

<https://github.com/orgs/cfs-apps/repositories>



**cFS target runs as a Linux Process**



## Ground System

cFS Basecamp - v1.8

System Developer Operator Documents Tutorials

Build cFS Start cFS Stop cFS Mission: **basecamp** Target: **cpu1** Image **/home/osk/cfs-basecamp/cfe-eds-framework/build/exe/cpu1/core-cf**

Ena Tlm Files... Quick Cmd: -- Common Commands-- Send Cmd: -- Command Topic -- View Tlm: -- Telemetry Topic --

**cFS Target Process Window** Telecommand: **127.0.0.1:1234** Telemetry: **Local** Time: **1001033**

```
EVS Port1 66/1/KIT_TO 306: Removed 0 table packet entries
EVS Port1 66/1/KIT_TO 310: Skip subscribing to tunnel message 0x0862(2146)
EVS Port1 66/1/KIT_TO 301: Successfully loaded new table with 46 packets
EVS Port1 66/1/KIT_TO 201: Packet Table load updated 74 entries
EVS Port1 66/1/KIT_TO 25: Successfully replaced table 0 using file /cf/kit_to_pkt_tbl.json
EVS Port1 66/1/KIT_TO 100: KIT_TO Initialized. Version 3.2.0
1980-012-14:03:20.71667 CFE_ES_Main: CFE_ES_Main entering APPS_INIT state
1980-012-14:03:20.71671 CFE_ES_Main: CFE_ES_Main entering OPERATIONAL state
EVS Port1 66/1/CFE_TIME 21: Stop FLYWHEEL
EVS Port1 66/1/KIT_TO 303: Telemetry output enabled for IP 127.0.0.1
EVS Port1 66/1/KIT_SCH 407: Slots skipped: slot = 2, count = 2
EVS Port1 66/1/KIT_SCH 406: Multiple slots processed: slot = 1, count = 2
EVS Port1 66/1/KIT_SCH 407: Slots skipped: slot = 2, count = 2
EVS Port1 66/1/KIT_SCH 404: Major Frame Sync too noisy (Slot 1). Disabling synchronization.
```

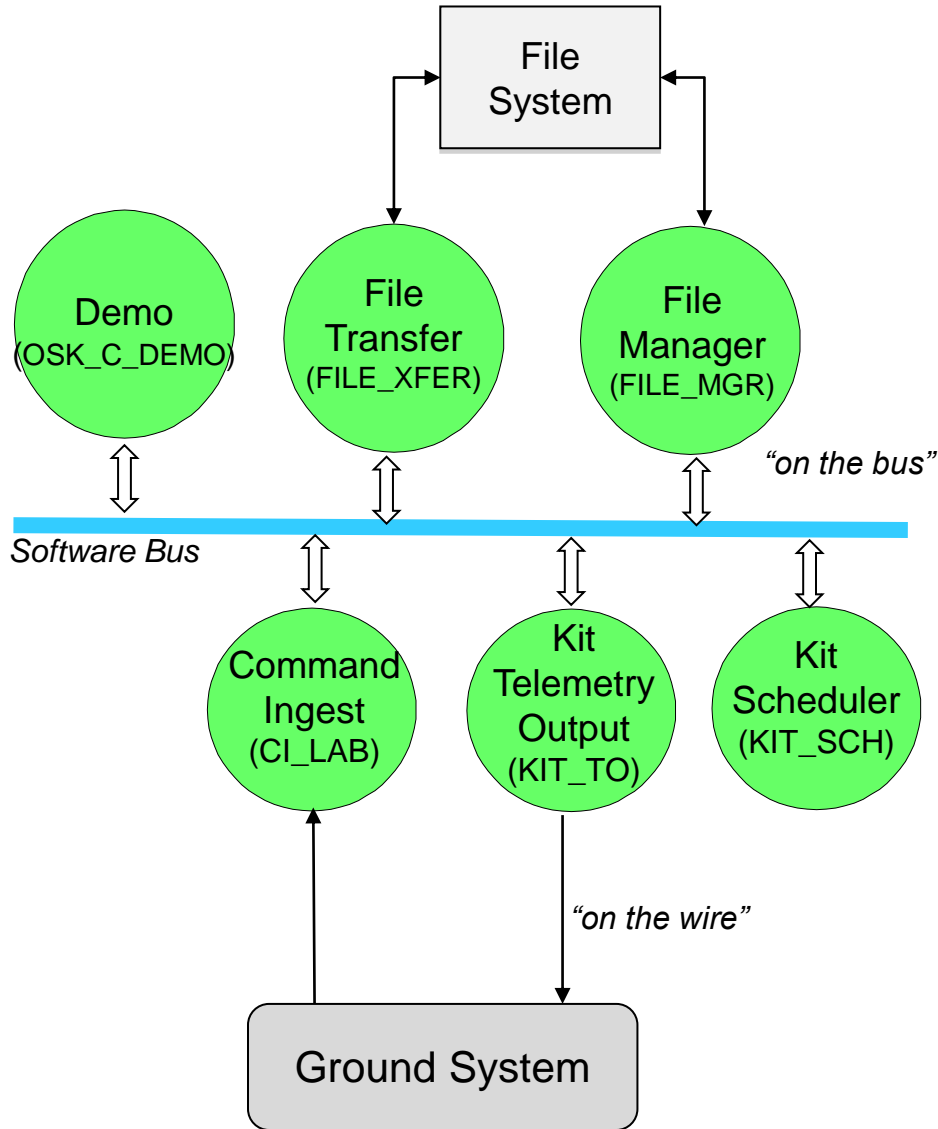
**Ground Events** Clear

```
16:34:52 - Basecamp version 1.8 initialized with mission 'basecamp', target 'cpu1' on 07/02/2023 at 16:34:52
16:34:52 - Basecamp target host 127.0.0.1, command port 1234, telemetry port 1235
16:35:02 - Sent KIT_TO/EnableOutput command
16:35:02 - Sent CFE_EVS/AddEventFilterCmd command
16:35:03 - Sent CFE_EVS/AddEventFilterCmd command
16:35:03 - FSW Event at 1001001: KIT_TO, 2 - Telemetry output enabled for IP 127.0.0.1
16:35:05 - FSW Event at 1001004: KIT_SCH, 3 - Slots skipped: slot = 2, count = 2
16:35:13 - FSW Event at 1001012: KIT_SCH, 2 - Multiple slots processed: slot = 1, count = 2
16:35:14 - FSW Event at 1001013: KIT_SCH, 3 - Slots skipped: slot = 2, count = 2
```

**Target Monitor Display**

# Basecamp cFS Target Apps

8/8



- **Electronic Data Sheets (EDS) specs define command and telemetry messages**
  - “on the wire” → are off card interfaces
  - “on the bus” ↔ are native host definitions
- **Basecamp comes preconfigured with 6 apps**
  - *CI\_LAB* and *KIT\_TO* manage external-to-internal message bus translations
  - *KIT\_SCH* coordinates synchronous application functionality
  - *FILE\_MGR* provides onboard directory and file management services
  - *FILE\_XFER* manage file transfers between flight and ground
  - *APP\_C\_DEMO* is used for educational purposes

# Basecamp Directory Structure

## **cfs-basecamp**

