Lesson 2

Objectives

- Describe how to run the core Flight System (cFS) from the GUI and from the command line
- Provide troubleshooting guidance

Preliminary Information

- The cFS requires some Linux resources that should not require elevated privileges to access but depending on the user's platform configuration the cFS may need to be run with elevated privieges using using SUDO

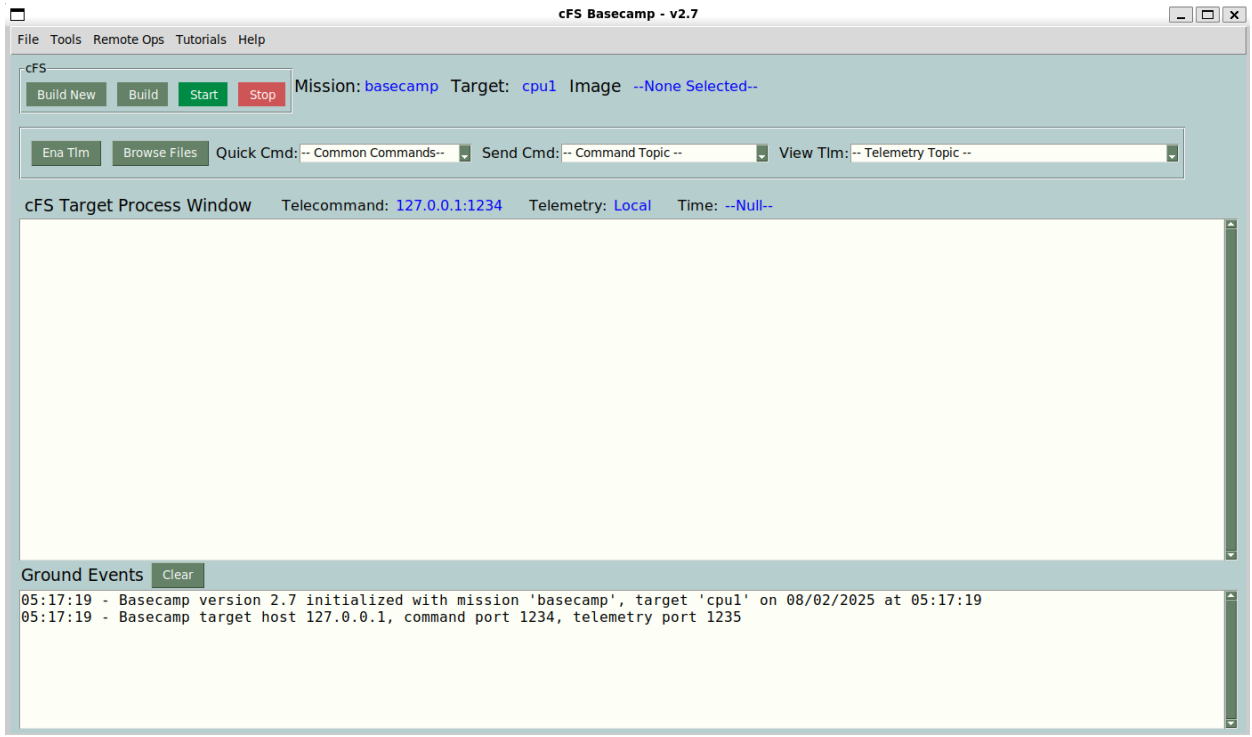
- *Appendix A: cFS Startup Troubleshooting* provides details for solving privilege related runtime issues

- Basecamp's GUI provides a single interface for building and running the cFS, however, you can still build and run from the command line and use the GUI for a command and telemetry interface
  - See *Appendix B: Start/Stop cFS from the Command Line*

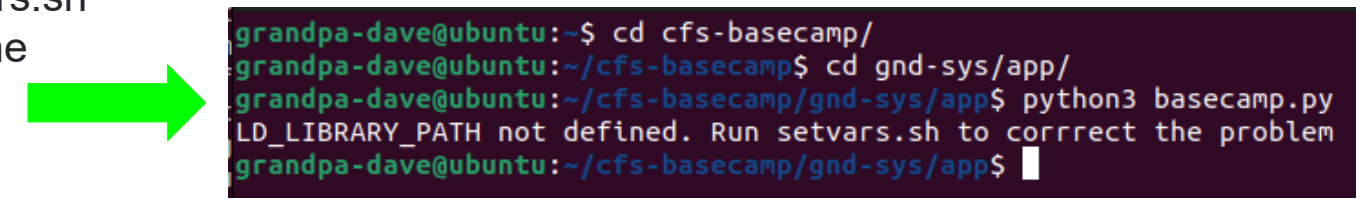
Starting the cFS from the GUI (1 of 3)

Open a terminal window and issue the following commands to start the GUI (assumes you are starting in git clone location):

```
cd cfs-basecamp/gnd-sys/app
. ./setvars.sh
python3 basecamp.py
```



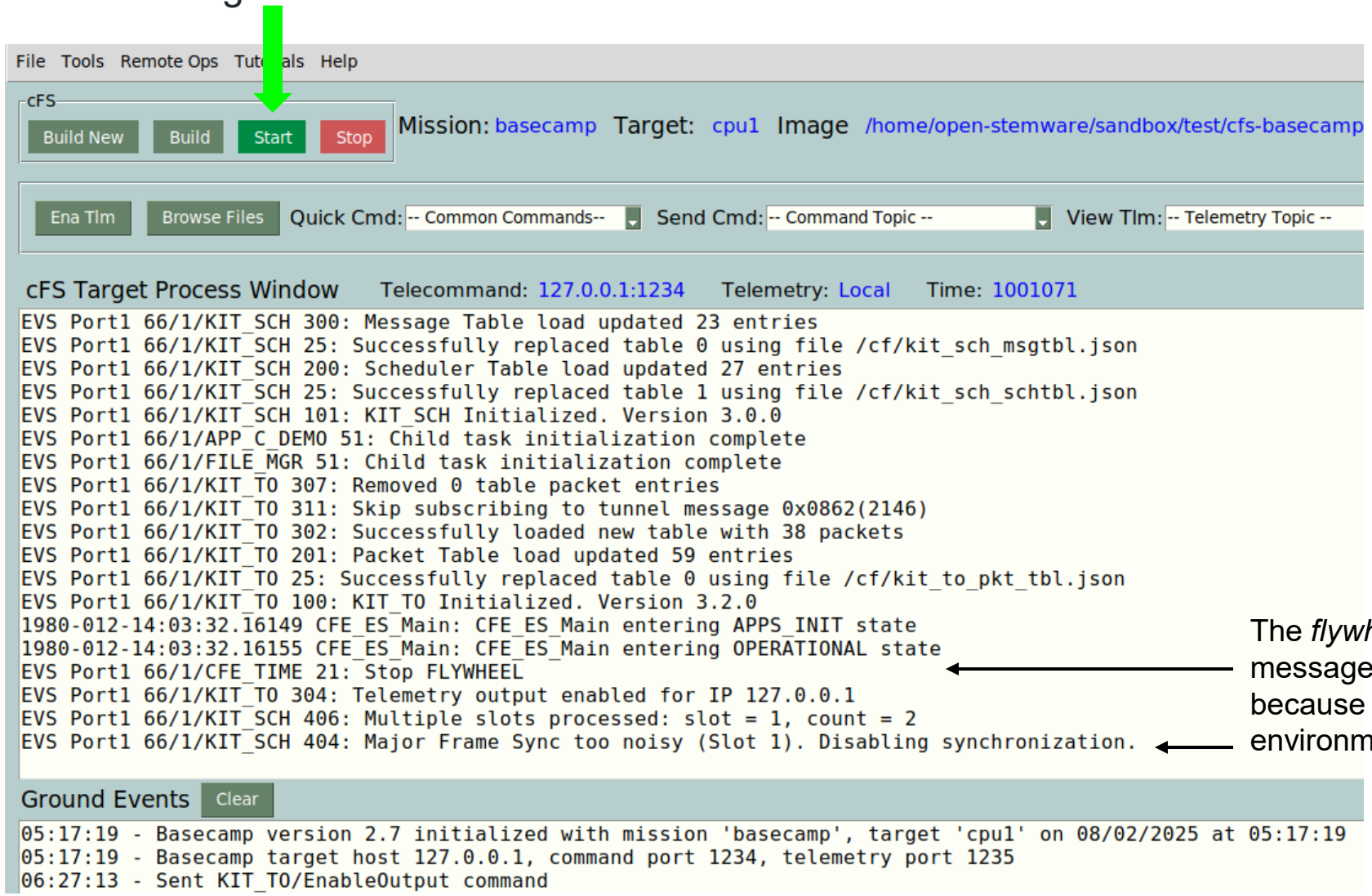
If you see the following error, then either setvars.sh wasn't run or the cFS has not been built and the python libraries don't exist.



Starting the cFS from the GUI (2 of 3)

Click <Start> to run the cFS in a new Linux process

- After a several seconds messages will appear in the 'cFS Process Window' to indicate the cFS is running



cFS Target Monitor Display

The *flywheel* and *sync* events messages are expected because of the non-realtime environment

Starting the cFS from the GUI (3 of 3)

If you scroll up in the cFS Process Window you can find when cfe_es_startup.scr is opened

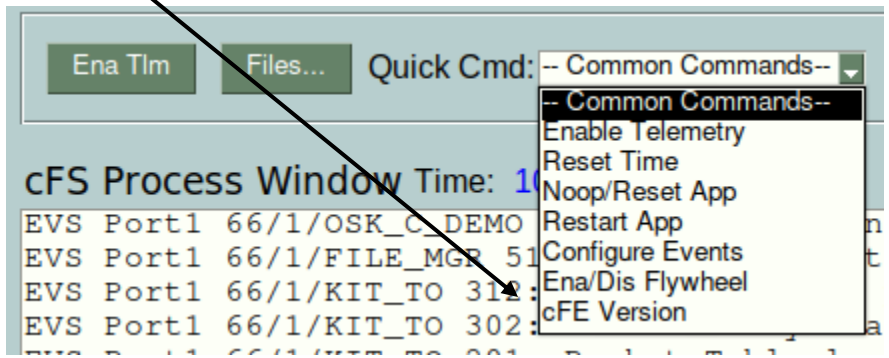
cFS Process Window Time: 1049773

```
1980-013-03:32:10.58743 CFE_ES_StartApplications: Opened ES App Startup file: /cf/cfe_es_startup.scr
1980-013-03:32:10.58762 CFE_ES_ParseFileEntry: Loading shared library: /cf/cfe_assert.so
1980-013-03:32:10.58801 [BEGIN] CFE FUNCTIONAL TEST
1980-013-03:32:10.58802 [BEGIN] 01 CFE-STARTUP
1980-013-03:32:10.58813 CFE_ES_ParseFileEntry: Loading shared library: /cf/osk_c_fw.so
OSK C Application Framework Library Initialized. Version 4.0.0
1980-013-03:32:10.64302 CFE_ES_ParseFileEntry: Loading file: /cf/osk_c_demo.so, APP: OSK_C_DEMO
1980-013-03:32:10.71270 CFE_ES_ParseFileEntry: Loading file: /cf/ci_lab.so, APP: CI_LAB_APP
1980-013-03:32:10.72164 CFE_ES_ParseFileEntry: Loading file: /cf/kit_to.so, APP: KIT_TO
OS_ModuleLoad_Impl():111:Error loading shared library: ./cf/sch_lab.so: cannot open shared object file: No such file or
directory
1980-013-03:32:10.72197 CFE_ES_ParseFileEntry: Loading file: /cf/sch_lab.so, APP: SCH_LAB_APP
1980-013-03:32:10.72201 CFE_ES_LoadModule: Could not load file:/cf/sch_lab.so. EC = -1
1980-013-03:32:10.72209 CFE_ES_ParseFileEntry: Loading file: /cf/file_mgr.so, APP: FILE_MGR
1980-013-03:32:10.72241 CFE_ES_ParseFileEntry: Loading file: /cf/file_xfer.so, APP: FILE_XFER
```

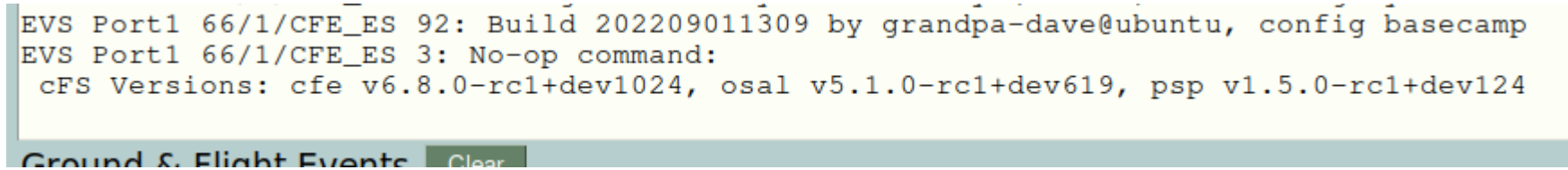
A successful startup includes success messages from each library and app as they are loaded and initialized

Verify Operations

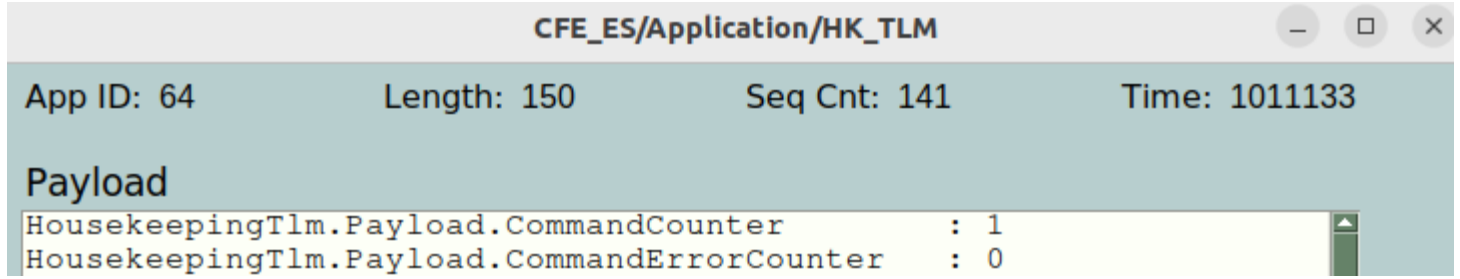
Using the Quick Cmd 'cFE Version' option is a quick way to verify the cFS target is operational



'cFE Version' sends a cFE Executive Service No Op command that reports the current cFE version in an event message



The No Op command also increments ES's command counter that is reported in its Housekeeping telemetry



Stopping the cFS from the GUI

From the GUI click <Stop> to terminate the cFS process

1. The cFS Process terminates
2. Time changes to *Null*
3. Image changes to *None Selected*

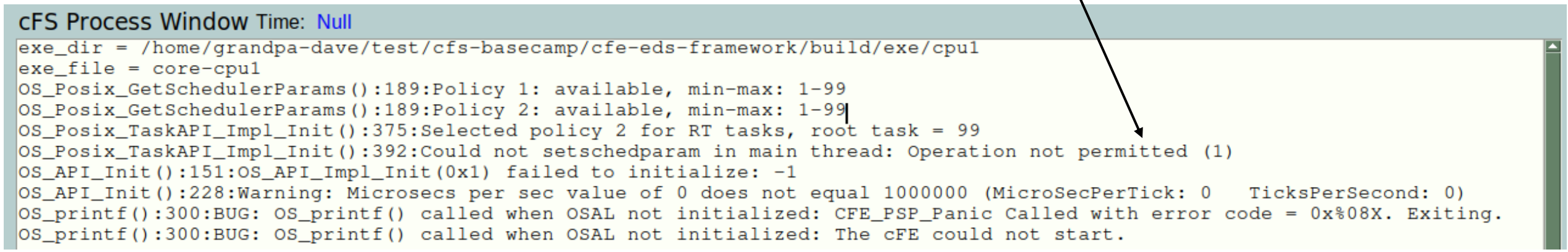
If the cFS was started from a terminal window, the GUI <Stop> button will have no effect

Appendix A

cFS Startup Troubleshooting

Potential Startup Error (1 of 2)

A known potential startup error is a permission denied as shown here



Potential Startup Error (2 of 2)

Potential causes and solutions

- The OSAL has a permissive mode setting that should be set sp resources can be accessed
- To perform a clean build from the command line
  - Issue *'make distclean'* to clear all cmake caches
  - Issue '*make SIMULATION=native prep'* to set OSAL's permissive mode should be set
- Basecamp's GUI <Build New> button performs these steps the two steps above
- If you continue to get the error, you can run in privileged mode
  - The configuration settings are is in cfs-basecamp/gnd-sys/app/basecamp.ini
  - `In [CFS_TARGET] section set SUDO_START_CFS to True or False(default)`
  - `In [APP] section set PASSWORD = <your Linux user account password>; only required if SUDO_START_CFS is set to True`

Basecamp Shell Scripts

- The Basecamp cFS buttons start shell scripts to perform the operation
- If you're comfortable with writing shell scripts you can modify them to suite your needs
- The shell scripts are located in **cfs-basecamp/gnd-sys/app**
- <Build New>
  - *build_cfs_topicids.sh*
- <Build>
  - *build_cfs.sh*
- <Start>
  - *start_cfs.sh* or *sudo_start_cfs.sh* depending on basecamp.ini configuration
- <Stop>
  - *stop_cfs.sh*

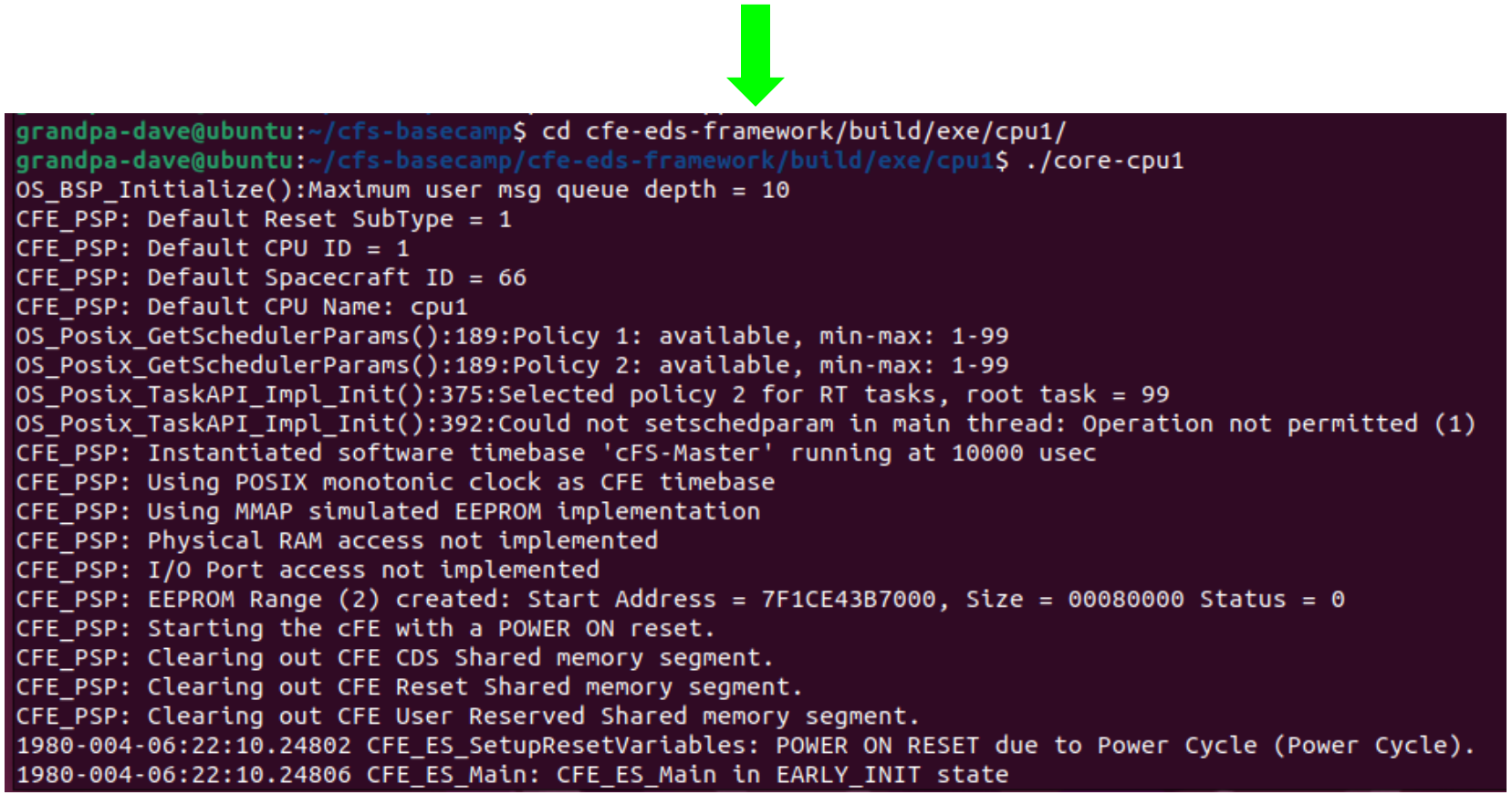
Appendix B

Start/Stop cFS from the Command Line

Starting the cFS from the Command Line (1 of 2)

Open a terminal window and issue the following commands (assumes starting in git clone location):

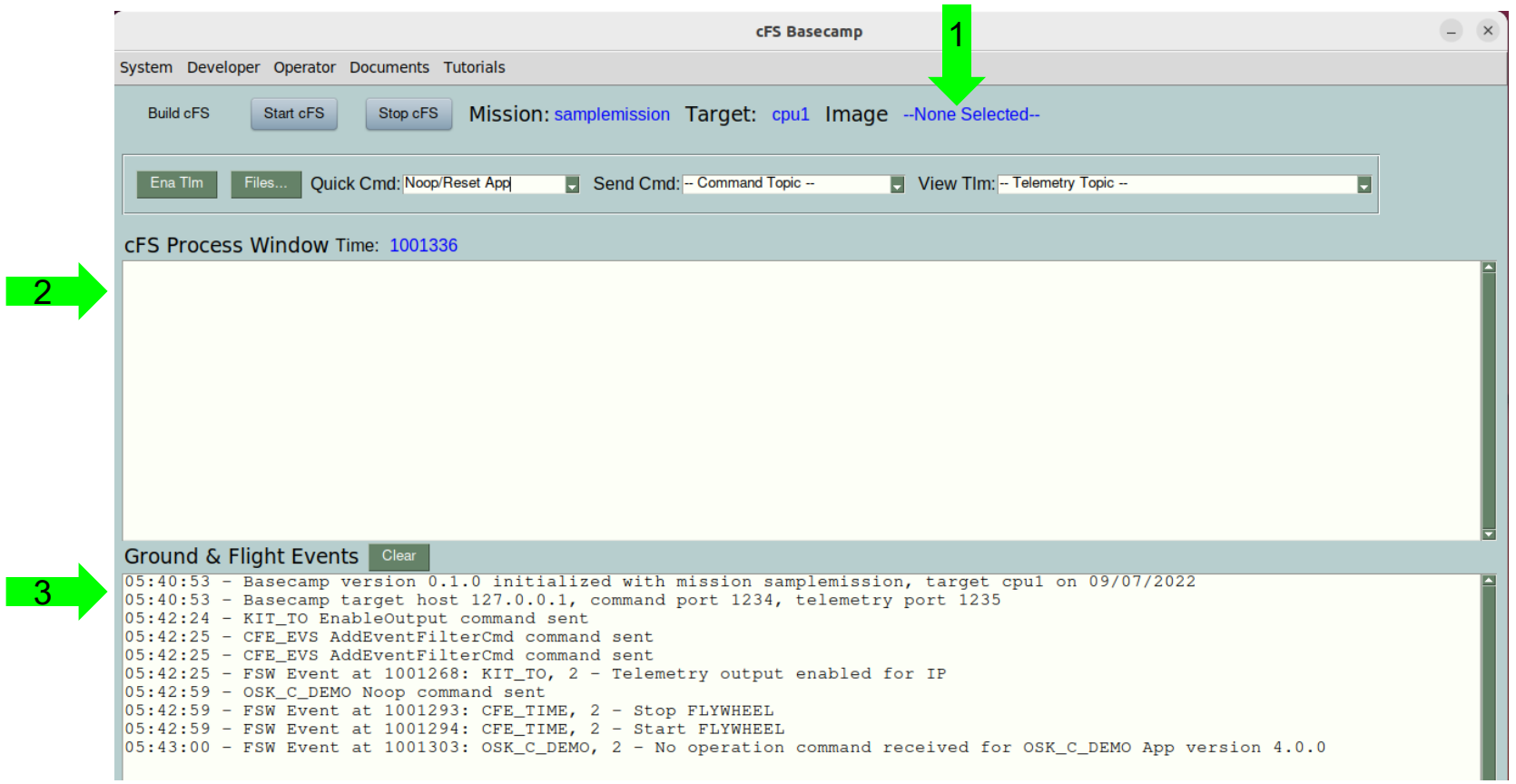
```
> cd cfs-basecamp/cfe-eds-framework/build/exe/cpu1
> ./core-cpu1
```



Starting the cFS from the Command Line (2 of 2)

The GUI can still be used with a cFS process that was not started using the GUI, however

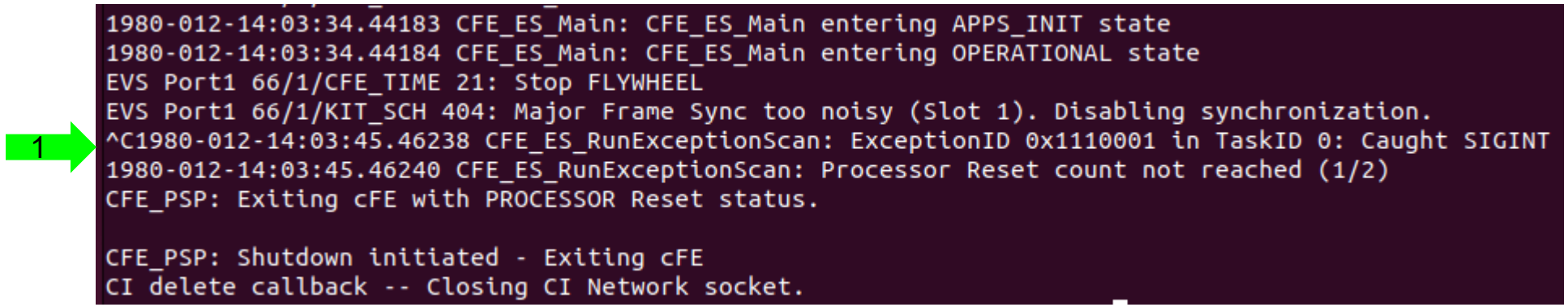
1. The *Image* name will not be loaded
2. The *cFS Process Window* will be blank. The terminal window where the cFS was started serves as the cFS Monitor Display
3. Flight events will still appear in the Ground & Flight Events window



Stopping the cFS from the Command Line

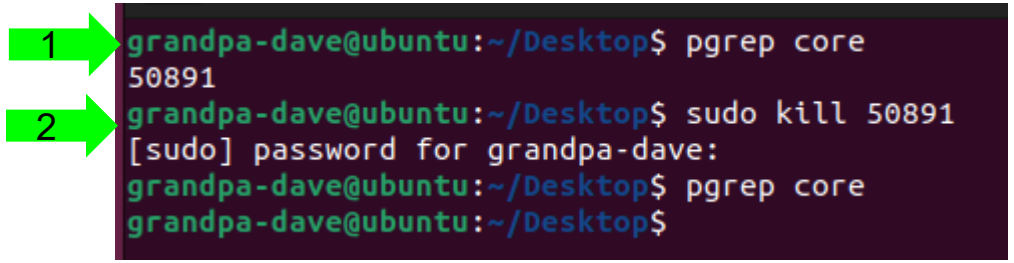
Enter Ctrl-c in the terminal window that was used to start the cFS

1. The exception is caught and cFS process is terminated



Terminate a cFS process from a terminal window not used to start the cFS

1. Determine the cFS process identifier
2. Use elevated privileges to “kill” the process



cFS Target Monitor Terminal