

- **OPEN** will capture any open of the file or directory named by the path.

The optional **output file** will be used for the timestamped filename of the output from the event (instead of the default **notify.stdout.YYMMDDHHMM**). Each time an event occurs matching a criteria specified in the **notify** file, the **notify_cb.sh** script is invoked (if it exists), passing in the given path and access mode. The script is also provided with the most recent command issued by either the container user, or the root account (whichever is more recent). If there is no **notify_cb.sh** script, then the output consists of the **file_path**, the most recent command, and the associated user (e.g., root). See the **acl** lab for an example.

The output from the notify event is captured in timestamped files, just as those resulting from events described in Section 6.1.5. If the optional output file provided in the **notify** list is given as **precheck**, then events resulting from program invocation, e.g., due to use of a **forcecheck** file, can be recorded in the very same timestamped file as events resulting from a **notify** file. In such cases, output from the former will precede output from the latter within the file. The framework will append the **notify** output to any timestamped **precheck.stdout** file that was created up to two seconds prior to the **notify** event. Inclusion of both outputs into one timestamped file allows the designer to identify events that occurred as part of a single program invocation. Again, see the **acl** lab for an example.

6.1.7 Generating results upon stopping the lab

The lab designer can cause a script to run on selected containers whenever the student stops a lab, or when a student issues the **checkwork** command per 6.4. This is achieved by creating an script or executable program at:

```
trunk/labainers/lab/<lab>/_bin/prestop
```

The stdout of any such program will be written to a timestamped file named **prestop.stdout.timestamp**. The framework ensures that all such scripts on all of the lab containers will complete prior to shutting down any of the containers, and all the timestamps will be the same. Note the Labainers framework generally allows students to achieve their goals at any point in their exploration, and the labs typically do not require the student to leave the system in any particular state. In other words, students should be free to continue experimenting subsequent to getting the correct results. Thus, any use of the prestop feature, (other than for *current state assessment* per 6.4.1), should be accompanied by a lab manual entry advising the student that they may restart a lab after issuing the **stoplab** command. ¹⁷

All **prestop** scripts will timeout after 30 seconds with a SIGTERM. For debugging support, please consider adding signal handling to your prestop scripts. For example, for a bash script, include:

```
trap "echo Timed out; exit" SIGTERM
```

6.1.8 Artifact archives

Artifacts from student labs are combined into a zip file that is placed in the student transfer directory, typically at **/labtainer/xfer/<labname>**. Students provide this file to their instructor for automated assessment, e.g., via email or an LMS.

Other uses for student artifacts are facilitated through use of a script named:

```
labainers/labs/<lab>/bin/postzip
```

If such a script exists, it is executed after all of the student artifacts are zipped up. See the cyberciege lab for an example of postzip processing.

¹⁷Perhaps a **goalsmet** type of command should be added that does nothing but record prestop results without actually stopping the lab?