TEXAS ADVANCED COMPUTING CENTER

WWW.TACC.UTEXAS.EDU

TEXAS
The University of Texas at Austin

# Link Record Generation in XALT

XALT : Link Record

PRESENTED BY:

**Amit Ruhela**

Texas Advanced Computing Center,

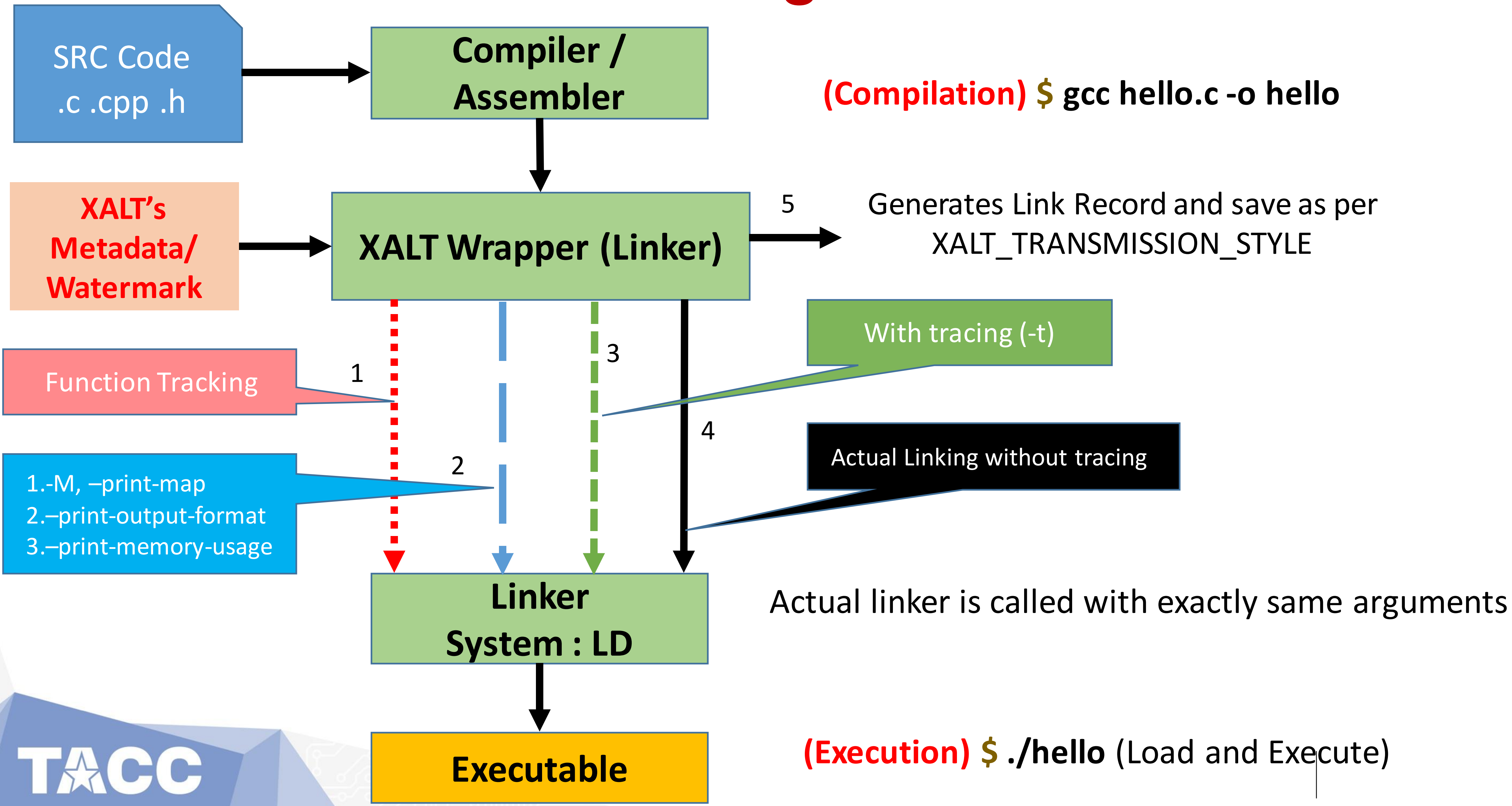
Austin Texas

**User Group Meeting, October 27, 2022**

# Agenda

- Learn how link record is generated
- Code Walkthrough - **xalt/src/linker/xalt_generate_linkdata.C**

# XALT Linking Process



SRC Code
.c .cpp .h

Compiler / Assembler

(Compilation) $ gcc hello.c -o hello

XALT's Metadata/ Watermark

XALT Wrapper (Linker)

5

Generates Link Record and save as per XALT_TRANSMISSION_STYLE

Function Tracking

1

1.-M, –print-map
2.–print-output-format
3.–print-memory-usage

2

3

With tracing (-t)

4

Actual Linking without tracing

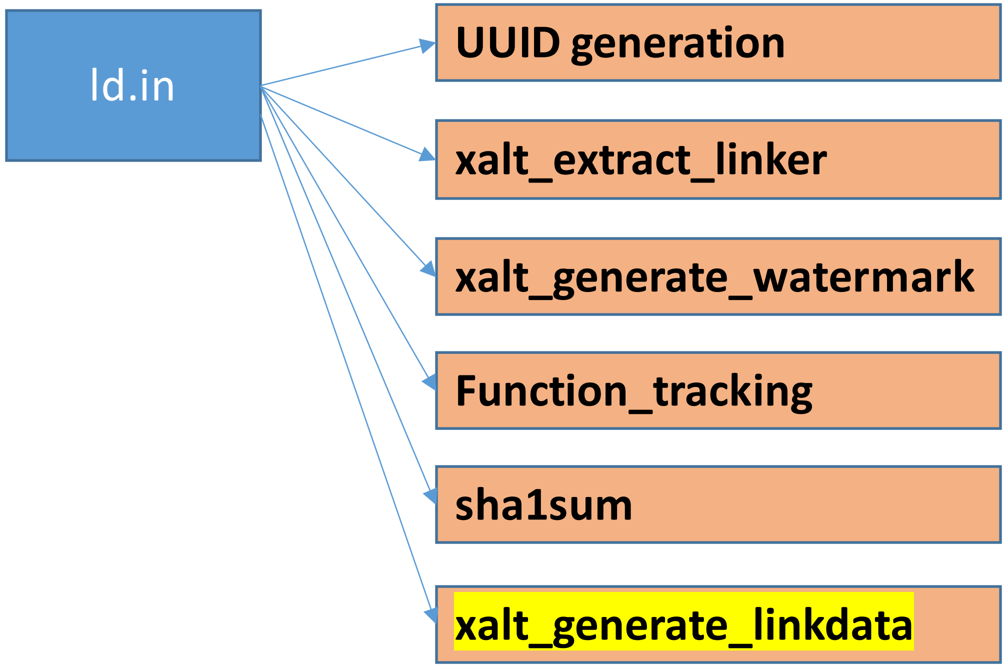
Linker
System : LD

Actual linker is called with exactly same arguments

Executable

(Execution) $ ./hello (Load and Execute)

# ld.in workflow



```
XALT_EXECUTABLE_TRACKING=no
LD_PRELOAD= PATH=@xalt_system_path@
LD_LIBRARY_PATH=$XALT_LD_LIBRARY_PATH
$GEN_LINKDATA
        "$UUID"  # Unique Universal ID
        "`pwd`"  # Working Directory
        "$SYSHOST" # System Name
        "$EXEC"  # Executable Name
        "$SHA1" # Checksum
        "$WATERMARK_O"  # Watermark Object
        "$EPOCH"  # Build Time of Executable
        "$FUNCRAW"  # Functions List
        "$LINKLINE_OUT"  # Link Line
        "$COMP_T”. # Compiler
```

# Workflow : Link Record Generation

1. Initialize with command line arguments
2. Call parseLDTrace on linklineFn seen by ld (generates linkA)
   - Removes lines having xalt.o, *tmp*, *.o, librun_submission.a etc. (Next Slide)
3. Call readFunctionList on funcRawFn (generates funcset)
   - Collect all function name that have “undefined references to”
4. Call parseCompTJsonStr
   - Extract Compiler, Compilerpath, linkline, and parentprocs
   - Create Key-Value pair (resultT) and add uuid, link_program, link_path, build_user, build_epoch, build_syshost, exec_path, hashid, workingdir
5. Create a json datastructure containing CRC, resultT, linkA, funcset, and link_line
6. Transmit data to File, syslog, logger, syslogv1, curl based on XALT_TRANSMISSION_STYLE

# LINKLINE_OUT

```
/lib/../lib64/crt1.o
/opt/apps/gcc/8.3.0/lib/gcc/x86_64-pc-linux-gnu/8.3.0/crtbegin.o
/tmp/icc1UIM3g.o
/opt/intel/compilers_and_libraries_2020.4.304/linux/mpi/intel64/lib/libmpifort.so
/opt/intel/compilers_and_libraries_2020.4.304/linux/mpi/intel64/lib/release/libmpi.so
/lib/../lib64/libdl.so
/lib/../lib64/librt.so
/lib/../lib64/libpthread.so
/lib64/libpthread.so.0
/usr/lib64/libpthread_nonshared.a
//opt/intel/compilers_and_libraries_2020.1.217/linux/compiler/lib/intel64_lin/libimf.a
/opt/intel/compilers_and_libraries_2020.1.217/linux/compiler/lib/intel64_lin/libsvml.a
/opt/intel/compilers_and_libraries_2020.1.217/linux/compiler/lib/intel64_lin/libirng.a
/lib/../lib64/libm.so
/opt/intel/compilers_and_libraries_2020.1.217/linux/compiler/lib/intel64_lin/libipgo.a
/opt/intel/compilers_and_libraries_2020.1.217/linux/compiler/lib/intel64_lin/libdecimal.a
/opt/apps/gcc/8.3.0/lib/gcc/x86_64-pc-linux-gnu/8.3.0//libgcc.a
/opt/apps/gcc/8.3.0/lib/gcc/x86_64-pc-linux-gnu/8.3.0/../../../../lib64/libgcc_s.so
/opt/apps/gcc/8.3.0/lib/gcc/x86_64-pc-linux-gnu/8.3.0/../../../../lib64/libgcc_s.so.1
/opt/apps/gcc/8.3.0/lib/gcc/x86_64-pc-linux-gnu/8.3.0//libgcc.a
/opt/apps/gcc/8.3.0/lib/gcc/x86_64-pc-linux-gnu/8.3.0//libgcc.a
/opt/intel/compilers_and_libraries_2020.1.217/linux/compiler/lib/intel64_lin/libirc.a
/opt/intel/compilers_and_libraries_2020.1.217/linux/compiler/lib/intel64_lin/libsvml.a
/lib/../lib64/libc.so
/lib64/libc.so.6
/usr/lib64/libc_nonshared.a
/lib64/ld-linux-x86-64.so.2
/usr/lib64/libc_nonshared.a
/lib64/ld-linux-x86-64.so.2
/opt/apps/gcc/8.3.0/lib/gcc/x86_64-pc-linux-gnu/8.3.0//libgcc.a
/opt/apps/gcc/8.3.0/lib/gcc/x86_64-pc-linux-gnu/8.3.0/../../../../lib64/libgcc_s.so
/opt/apps/gcc/8.3.0/lib/gcc/x86_64-pc-linux-gnu/8.3.0/../../../../lib64/libgcc_s.so.1
/opt/apps/gcc/8.3.0/lib/gcc/x86_64-pc-linux-gnu/8.3.0//libgcc.a
/opt/intel/compilers_and_libraries_2020.1.217/linux/compiler/lib/intel64_lin/libirc_s.a
/lib/../lib64/libdl.so
/lib/../lib64/libc.so
/lib64/libc.so.6
/usr/lib64/libc_nonshared.a
/lib64/ld-linux-x86-64.so.2
/opt/apps/gcc/8.3.0/lib/gcc/x86_64-pc-linux-gnu/8.3.0/crtend.o
/lib/../lib64/crtn.o
/tmp/aruhela_268c0a33-5684-4521-803e-43aed1bf5f1f_V5F61p/xalt.o
```

Note : Red lines will be filtered out by parseLDTrace function

# Code Walkthrough

https://github.com/xalt/xalt/blob/main/src/linker/xalt_generate_linkdata.C

https://github.com/xalt/xalt/blob/08c78182870860ac9a2ae53057fcd1e0d0267661/src/linker/parseLDTrace.C

https://github.com/xalt/xalt/blob/08c78182870860ac9a2ae53057fcd1e0d0267661/src/util/transmit.c

# Next Meeting

December 15th at 10:00 AM  CST (15:00 UTC )

# Thanks for Listening

# Links

1. XALT Documentation : https://xalt.readthedocs.io/en/latest/index.html#

2. XALT LD: https://github.com/xalt/xalt/blob/master/sh_src/ld.in

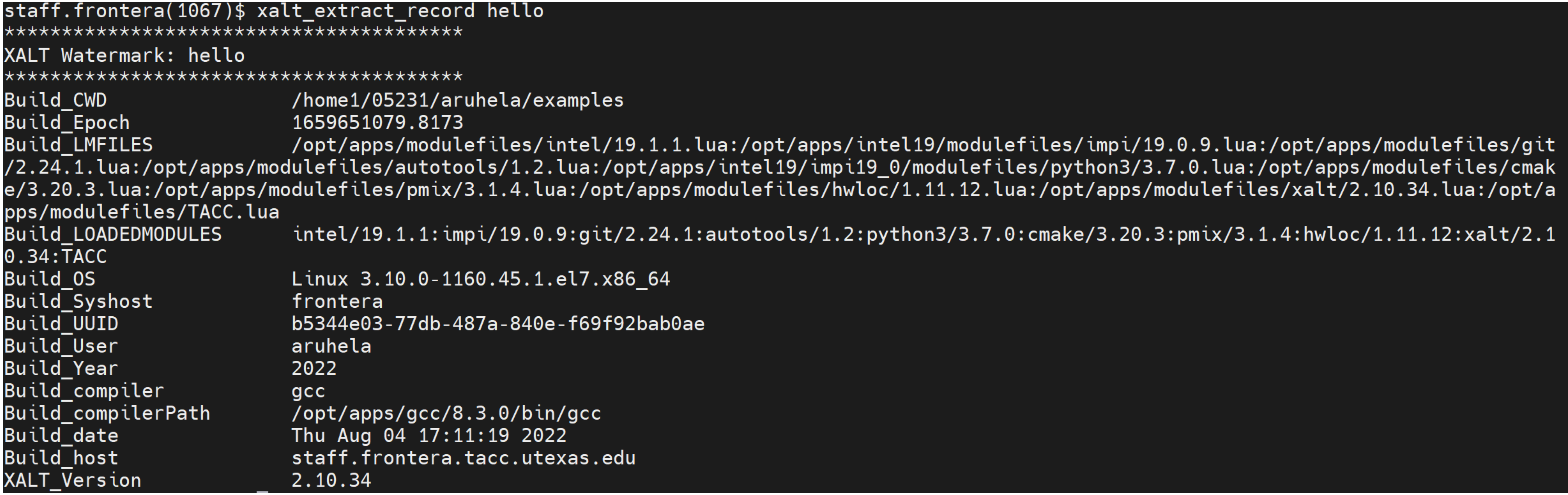
3. Bash https://pubs.opengroup.org/onlinepubs/9699919799/utilities/V3_chap02.html#tag_18_06_02

4. Shared vs Static Libraries : https://www.linuxtopia.org/online_books/an_introduction_to_gcc/gccintro_25.html

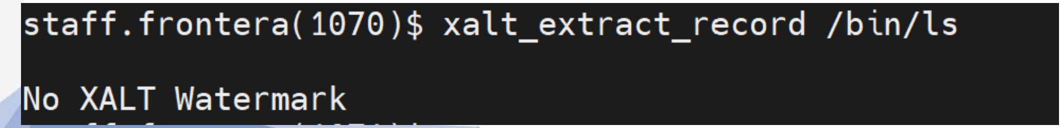
# Backup Slides

# Read Watermarks

$ xalt_extract_record  <file>



```
staff.frontera(1067)$ xalt_extract_record hello
****************************************
XALT Watermark: hello
****************************************
Build_CWD                /home1/05231/aruhela/examples
Build_Epoch              1659651079.8173
Build_LMFILES            /opt/apps/modulefiles/intel/19.1.1.lua:/opt/apps/intel19/modulefiles/impi/19.0.9.lua:/opt/apps/modulefiles/git
/2.24.1.lua:/opt/apps/modulefiles/autotools/1.2.lua:/opt/apps/intel19/impi19_0/modulefiles/python3/3.7.0.lua:/opt/apps/modulefiles/cmak
e/3.20.3.lua:/opt/apps/modulefiles/pmix/3.1.4.lua:/opt/apps/modulefiles/hwloc/1.11.12.lua:/opt/apps/modulefiles/xalt/2.10.34.lua:/opt/a
pps/modulefiles/TACC.lua
Build_LOADEDMODULES      intel/19.1.1:impi/19.0.9:git/2.24.1:autotools/1.2:python3/3.7.0:cmake/3.20.3:pmix/3.1.4:hwloc/1.11.12:xalt/2.1
0.34:TACC
Build_OS                 Linux 3.10.0-1160.45.1.el7.x86_64
Build_Syshost            frontera
Build_UUID               b5344e03-77db-487a-840e-f69f92bab0ae
Build_User               aruhela
Build_Year               2022
Build_compiler           gcc
Build_compilerPath       /opt/apps/gcc/8.3.0/bin/gcc
Build_date               Thu Aug 04 17:11:19 2022
Build_host               staff.frontera.tacc.utexas.edu
XALT_Version             2.10.34
```



```
staff.frontera(1070)$ xalt_extract_record /bin/ls

No XALT Watermark
```

# XALT – Function Tracking

**Goal** : What external functions and subroutines are used by an application?

- Function tracking is performed at Link Time and not Run Time
- Done by dual linking process
    1. First linking done by system linker (ld)
    2. Second linking done by Xalt wrapper that does fail linking by omitting object files and libraries that are found in the reverse map (rmap)
        - Rmap is JSON-formatted file that captures association between libraries, their path and module information, and names of functions/subroutines in them.