

ooRexx **FOR** ANDROID

Thomas Kahr
Seminar aus BIS
Thema 19



Overview

Part 1. Overview and Terminology

Part 2. Building ooRexx from Source Code

Part 3. Setup Android

Part 4. Porting ooRexx

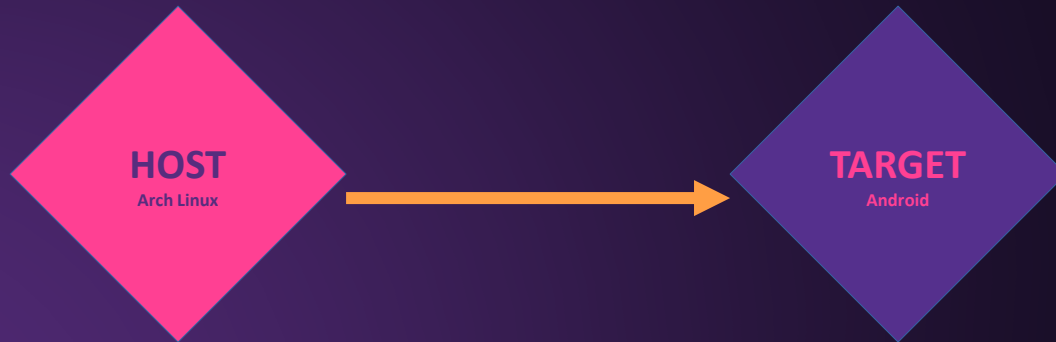
Part 5. Status and Outlook

PART 1 | Overview

1.1. Environment

1.2. install vs. build from source

1.1. Environment



- Source Files
- Tools to Build
- Emulator

1.2. Install vs. Build from Source

Available Installer for:

- MacOS
- Windows
- Ubuntu

Not for: Arch Linux

Build from Source:

- 1) Download source Files
- 2) Build ooRexx
- 3) Install ooRexx



PART 2 | **Building ooRexx from Source Code (Linux)**

2.1. Tools needed

2.2. Build process

2.1. Tools needed

Subversion

Central version control system
Server Side project history
Comparable to GIT

Compiler

GCC
compile Source Code to
Executables

CMake

Tool to manage the build Process
in big Projects

2.3. Tools needed | Compiler

Compiling a program with GCC

cpp file

```
ls  
calculator  calculator.cpp  working_hello
```

Compiling Command

```
~/coding/cpp  
11:01:11 > g++ calculator.cpp -o calc
```

Compiled Output

```
~/coding/cpp  
11:01:26 > ls  
calc  calculator  calculator.cpp  working_hello  
  
~/coding/cpp  
11:01:28 > ./calc  
Put in your first number: 10  
Your number is: 10  
Put in your second number: 10  
Your second number is: 10  
SUM = 20  
  
~/coding/cpp  
11:01:41 >
```

Compiler Steps

Preprocessing

Compilation

Assembling

Linking

2.4. CMake

Compiling a program with CMake

- CMake creates Makefiles
- More complex than Makefile but less complicated
- Especially useful for crossbuilds
(Toolchain)

```
nvim CMakeLists.txt /home/tjk  
cmake_minimum_required(VERSION 3.6.0)  
project(calculator)  
add_executable(calculator calculator.cpp)
```

2.4. CMake

CmakeLists.txt ooRexx

```
# Build for Apple macOS                                     (done)
if (BUILD_DMG)
# The installer is built in a post-process script build_macOS_dmg.sh.in
# initialised from platform/unix/macOS/install/build_macOS_dmg.sh.in
# using settings from here                                     (main)

# Always 64 bit for macOS                                     (done)
set(OS_DIST "macOS")

# This is copied from DEB & RPM Installers                   (done)
if (DEFINED PROO)
set(CPACK_PACKAGE_RELEASE 1)
else ()
set(CPACK_PACKAGE_RELEASE ${ORX_BLD_LVL})
endif ()

if (DEFINED OS_DIST)
set(CPACK_PACKAGE_FILE_NAME "${CPACK_PACKAGE_NAME}-${ORX_MAJOR}.${ORX_MINOR}.${ORX_MOD_LVL}-${CPACK_P
CKAGE_RELEASE}.${OS_DIST}.${CMAKE_HOST_SYSTEM_PROCESSOR}")
else ()
set(CPACK_PACKAGE_FILE_NAME "${CPACK_PACKAGE_NAME}-${ORX_MAJOR}.${ORX_MINOR}.${ORX_MOD_LVL}-${CPACK_P
CKAGE_RELEASE}.${CMAKE_HOST_SYSTEM_PROCESSOR}")
endif ()

message(STATUS "CPACK_PACKAGE_FILE_NAME ${CPACK_PACKAGE_FILE_NAME}")

# Be sure to do this AFTER you set the CPACK_PACKAGE_FILE_NAME
configure_file( ${CMAKE_SOURCE_DIR}/platform/unix/macOS/install/build_macOS_dmg.sh.in ${CMAKE_BINARY_DIR
/build_macOS_dmg.sh IMMEDIATE @ONLY )

endif ()

# Do not move the following statement!! It must be the last statement in this file.
include(CPack)
CMakeLists.txt
```

Frequency	Percentage
Never	10%
Sometimes	40%
Often	50%

Download

```

A main/trunk/1/bakedef.cmp
A main/trunk/Utilities
A main/trunk/Utilities/platform
A main/trunk/Utilities/platform/windows build directory
A main/trunk/Utilities/platform/windows/rexhide
A main/trunk/Utilities/platform/windows/rexhide/rexhide.cpp
A main/trunk/Utilities/platform/windows/rexhide/rexhide.mak
A main/trunk/Utilities/platform/windows/rexpaws
A main/trunk/Utilities/platform/windows/rexpaws/rexpaws.cpp
A main/trunk/Utilities/platform/windows/rexpaws/rexpaws.mak
A main/trunk/Utilities/platform/unix
A main/trunk/Utilities/rexx
A main/trunk/Utilities/rexx/platform
A main/trunk/Utilities/rexx/platform/unix
A main/trunk/Utilities/rexx/platform/windows/rexx.cpp
A main/trunk/Utilities/rexx/platform/windows/rexx.cpp really only rexx
A main/trunk/Utilities/rexxc
A main/trunk/Utilities/rexxc/platform
A main/trunk/Utilities/rexxc/platform/unix
A main/trunk/Utilities/rexxc/platform/windows/rexxCompiler.cpp
A main/trunk/Utilities/rexxc/platform/windows/RexxCompiler.cpp
A main/trunk/Utilities/rexximage
A main/trunk/Utilities/rexximage/rexximage.cpp
A main/trunk/Utilities/rxqueue
A main/trunk/Utilities/rxqueue/platform
A main/trunk/Utilities/rxqueue/platform/unix
A main/trunk/Utilities/rxqueue/platform/unix/rxqueue.cpp
A main/trunk/Utilities/rxqueue/platform/windows
A main/trunk/Utilities/rxqueue/platform/windows/rxqueue.cpp
A main/trunk/Utilities/rxsubcom
A main/trunk/Utilities/rxsubcom/platform
A main/trunk/Utilities/rxsubcom/platform/unix
A main/trunk/Utilities/rxsubcom/platform/unix/rxsubcom.cpp
A main/trunk/Utilities/rxsubcom/platform/windows
A main/trunk/Utilities/rxsubcom/platform/windows/rxsubcom.cpp
A main/trunk/CHANGES To install the version you just built
A main/trunk/README version
A main/trunk/CONTRIBUTORS
A main/trunk/files.html same source materials
A main/trunk/CPV1.0.txt
U main/trunk Optionally you can create a package
Checked out revision 12264.

```

CMake

```

coding/test/main/trunk via △v3.20.3
13:07:02 } cmake -DBUILD_DEB=1 -DOS_DIST=ubuntu1604 -DCMAKE_BUILD_TYPE=RELEASE
-- CMake version is 3.20.3
-- The C compiler identification is GNU 11.1.0
-- The CXX compiler identification is GNU 11.1.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/gcc - skipped
-- Detecting C compile features
-- Detecting C compile features - done

-- Looking for sys/types.h
-- Looking for sys/types.h - found
-- Looking for sys/time.h
-- Looking for sys/time.h - found
-- Looking for sys/utsname.h
-- Looking for sys/utsname.h - found
-- Looking for sys/wait.h
-- Looking for sys/wait.h - found
-- Performing Test HAVE_UNION_SEMUN
-- Performing Test HAVE_UNION_SEMUN - Failed
-- Looking forunistd.h
-- Looking forunistd.h - found
-- Looking for wsyncup in /usr/lib/libcurses.so
-- Looking for wsyncup in /usr/lib/libcurses.so - found
-- Looking for cbreak in /usr/lib/libcurses.so
-- Looking for cbreak in /usr/lib/libcurses.so - found
-- Looking for nodelay in /usr/lib/libcurses.so
-- Looking for nodelay in /usr/lib/libcurses.so - found
-- Found Curses: /usr/lib/libcurses.so
-- CURSES HAVE NCURSES_H is /usr/include/ncurses.h

```

```
~/coding/test/main/trunk via Δv3.28.3
13:15:03 > cmake -build ./
[ 1%] Generating rxsubcom.1.gz
[ 1%] Built target rxsubcom.man
[ 2%] Generating rxqueue.1.gz
[ 2%] Built target rxqueue.man
[ 2%] Generating rexxc.1.gz
[ 2%] Built target rexxc.man
[ 2%] Generating bin/rxrexp.cls
[ 2%] Generating bin/rxftp.cls
[ 2%] Generating bin/csvStream.cls
[ 3%] Generating bin/json.cls

ect testbinaries/CMakeFiles/orxinvoke.dir/orxinvoke.cpp.o
ed library ../lib/liborxinvoke.so
invoke
ect testbinaries/CMakeFiles/orxfun.dir/orxfun.cpp.o
ed library ../lib/liborxfun.so
fun
ect testbinaries/CMakeFiles/rexxinstance.dir/rexxinstance.cpp.o
utable ../bin/rexxinstance
xinstance
ed testbinaries/CMakeFiles/orxclassic.dir/orxclassic.cpp.o
ed library ../lib/liborxclassic.so
classic

nk via Δv3.28.3 took 1m17s
```

PART 3 | Android

3.1. Android Studio, SDK, NDK

3.2. Android Toolchain

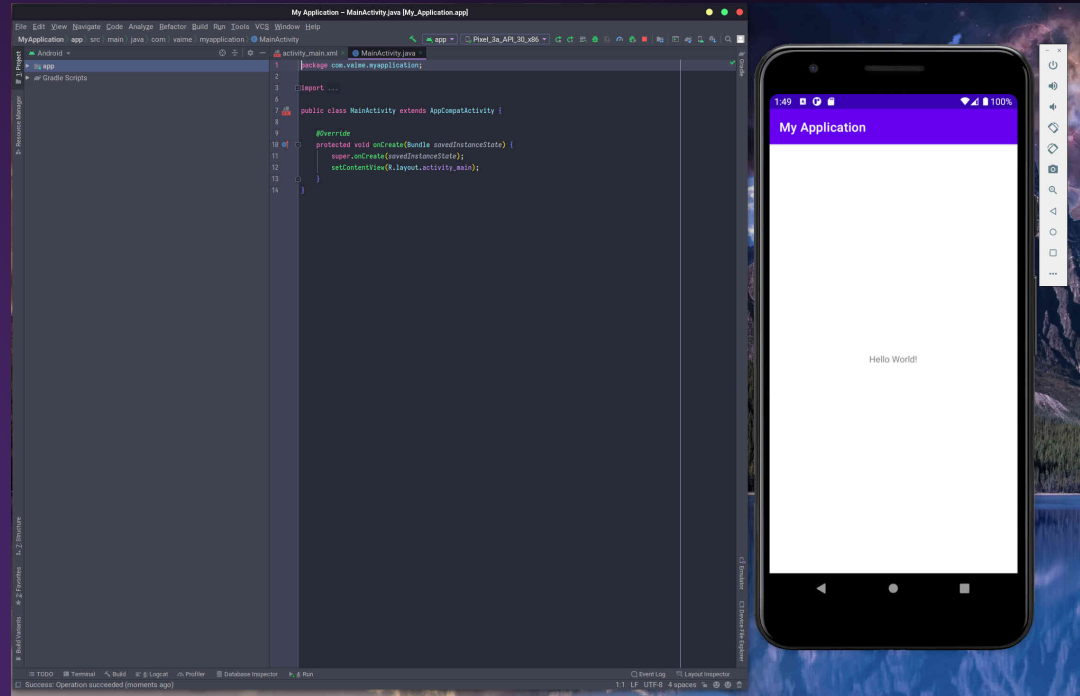
3.3. Emulator, Shell

3.4. Run Executable on Device

3.1. Android Software Tools

Android Studio

- IDE – Integrated Development Environment



3.1. Android Software Tools

SDK

Android Software Development Kit

- necessary to build Programs with Android Studio
- Build in Android Studio
- Provides API libraries and developer tools

NDK

Android Native Development Kit

- Toolset for porting C/C++ project to Android
- includes Android Toolchain since ~2018

- Needs to be installed explicitly

ADB

Android Debug Bridge

- Connecting with the Device over Terminal
- Pushing files to Device



3.2. Android Toolchain File

Android Toolchain

- Comes together with Android NDK
- Before 2018 -Developer Project
- Used together with CMake to build for Android target Systems

```
# https://github.com/android-ndk/ndk/issues/323
if(ANDROID_NDK_TOOLCHAIN_INCLUDED)
    return()
endif(ANDROID_NDK_TOOLCHAIN_INCLUDED)
set(ANDROID_NDK_TOOLCHAIN_INCLUDED true)

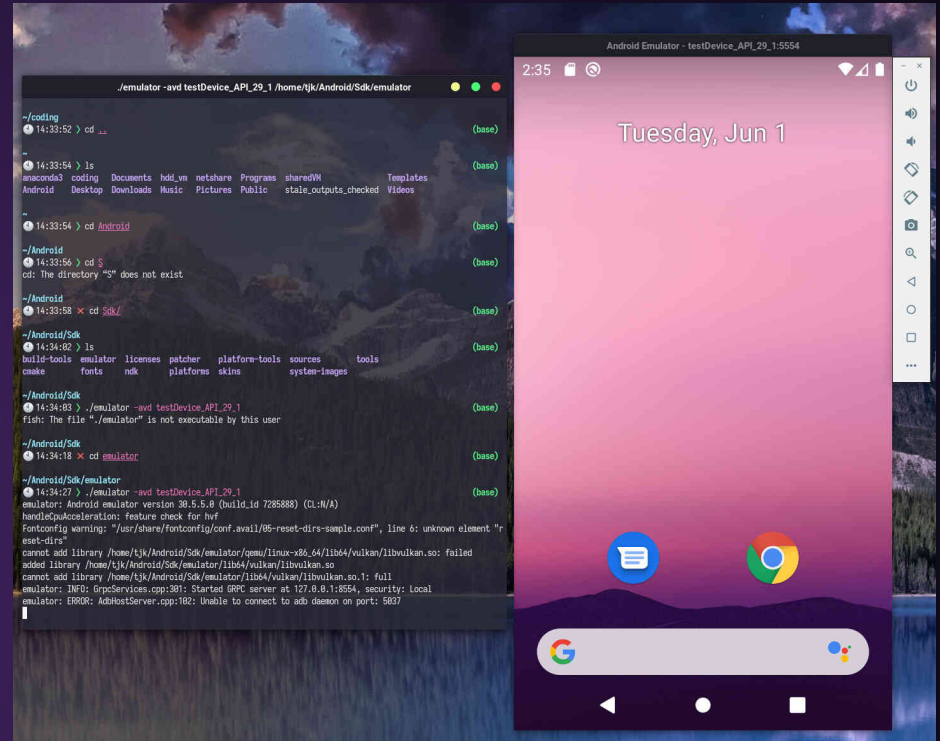
# Android NDK
get_filename_component(ANDROID_NDK_EXPECTED_PATH
    "${CMAKE_CURRENT_LIST_DIR}/../.." ABSOLUTE)
if(NOT ANDROID_NDK)
    set(ANDROID_NDK "${ANDROID_NDK_EXPECTED_PATH}")
else()
    # Allow the user to specify their own NDK path, but emit a warning. This is an
    # uncommon use case, but helpful if users want to use a bleeding edge
    # toolchain file with a stable NDK.
    # https://github.com/android-ndk/ndk/issues/473
    get_filename_component(ANDROID_NDK "${ANDROID_NDK}" ABSOLUTE)
    if(NOT "${ANDROID_NDK}" STREQUAL "${ANDROID_NDK_EXPECTED_PATH}")
        message(WARNING "Using custom NDK path (ANDROID_NDK is set): ${ANDROID_NDK}")
    endif()
endif()
unset(ANDROID_NDK_EXPECTED_PATH)
file(TO_CMAKE_PATH "${ANDROID_NDK}" ANDROID_NDK)

# Android NDK revision
# Possible formats:
# * r16, build 1234: 16.0.1234
# * r16b, build 1234: 16.1.1234
# * r16 beta 1, build 1234: 16.0.1234-beta1
#
# Canary builds are not specially marked.
file(READ "${ANDROID_NDK}/source.properties" ANDROID_NDK_SOURCE_PROPERTIES)
```

3.3. Emulator, Shell 1

Start Emulator over Terminal

- Navigate to SDK Folder
- Command: **emulator -awd deviceName**



3.3. Emulator, Shell 2

Use ADB to transfer files

- Push files to Device: `/adb push fileName`
- Connect to Shell: `/adb shell`
- “su” for root (#) privileges

```
./platform-tools/adb shell /home/tjk/Android/Sdk
generic_x86:/ # ls -l
total 128
dr-xr-xr-x 70 root root 0 2021-05-22 17:04 acct
-rw-r--r-- 1 root root 723 2020-07-21 02:41 adb_keys
drwxr-xr-x 14 root root 280 2021-05-22 17:04 apex
lrw-r--r-- 1 root root 11 2020-07-21 02:56 bin -> /system/bin
lrw-r--r-- 1 root root 50 2020-07-21 02:56 bugreports -> /data/user_de/0/com.android.shel
drwxrwx--- 2 system cache 4096 2020-07-21 02:18 cache
lrw-r--r-- 1 root root 19 2020-07-21 02:56 charger -> /system/bin/charger
drwxr-xr-x 4 root root 0 2021-05-22 17:04 config
lrw-r--r-- 1 root root 17 2020-07-21 02:56 d -> /sys/kernel/debug
drwxrwx--x 46 system system 4096 2021-05-03 19:55 data
drwxr-xr-x 2 root root 4096 2020-07-21 02:18 debug_ramdisk
lrw----- 1 root root 23 2020-07-21 02:56 default.prop -> system/etc/prop.default
drwxr-xr-x 18 root root 1240 2021-05-22 17:04 dev
lrw-r--r-- 1 root root 11 2020-07-21 02:56 etc -> /system/etc
lrwxr-x--- 1 root shell 16 2020-07-21 02:56 init -> /system/bin/init
-rwxr-x--- 1 root shell 1653 2020-07-21 02:18 init.environ.rc
-rwxr-x--- 1 root shell 33152 2020-07-21 02:41 init.rc
-rwxr-x--- 1 root shell 7690 2020-07-21 02:40 init.usb.configfs.rc
-rwxr-x--- 1 root shell 5649 2020-07-21 02:40 init.usb.rc
-rwxr-x--- 1 root shell 563 2020-07-21 02:40 init.zygote32.rc
drwx----- 2 root root 16384 2020-07-21 02:57 lost+found
drwxr-xr-x 2 root root 4096 2020-07-21 02:18 metadata
drwxr-xr-x 12 root system 260 2021-05-22 17:04 mnt
drwxr-xr-x 2 root root 4096 2020-07-21 02:18 odm
drwxr-xr-x 2 root root 4096 2020-07-21 02:18 oem
dr-xr-xr-x 194 root root 0 2021-05-22 17:04 proc
lrw-r--r-- 1 root root 15 2020-07-21 02:56 product -> /system/product
lrw-r--r-- 1 root root 24 2020-07-21 02:56 product_services -> /system/product_services
drwxr-xr-x 3 root root 4096 2020-07-21 02:41 res
drwxr-x--- 2 root shell 4096 2020-07-21 02:18 sbin
lrw-r--r-- 1 root root 21 2020-07-21 02:56 sdcard -> /storage/self/primary
drwxr-xr-x 5 root root 100 2021-05-22 17:05 storage
dr-xr-xr-x 12 root root 0 2021-05-22 17:04 sys
drwxr-xr-x 13 root root 4096 2020-07-21 02:56 system
-rw-r--r-- 1 root root 2608 2020-07-21 02:18 ueventd.rc
drwxr-xr-x 8 root root 4096 2020-07-21 02:42 vendor
generic_x86:/ #
```

3.4. Run Executable on Device

! After running through Cmake and the Android Toolchain it's not possible to execute the File on the Host System

```
~/coding/cpp/working_hello via △v3.20.3
14:56:48 > ls
calc CMakeCache.txt CMakeFiles cmake_install.cmake CMakeLists.txt

~/coding/cpp/working_hello via △v3.20.3
14:56:50 > ./calc
Failed to execute process './calc'. Reason:
The file './calc' does not exist or could not be executed.

~/coding/cpp/working_hello via △v3.20.3
14:56:55 ✖
```

```
./platform
generic_x86:/data/local/tmp/test # ls
calc
generic_x86:/data/local/tmp/test # ./calc
Put in your first number: 10
Your number is: 10
Put in your second number: 20
Your second number is: 20
SUM = 30
generic_x86:/data/local/tmp/test #
```

PART 4 | Porting ooRexx

4.1. CMake Command

4.2. CMake Output & Problems

4.3. Solving Problems

4.1. CMake Command

```
cmake -DCMAKE_TOOLCHAIN_FILE=~/.coding/android-ndk-r22b/build/cmake/android.toolchain.cmake -DANDROID_ABI="x86" -DCMAKE_BUILD_TYPE=Release \
      -DCMAKE_C_COMPILER=$CC -DCMAKE_CXX_COMPILER=$CXX -DANDROID_NATIVE_API_LEVEL=28
```

4.1. CMake Output & Problems

```
-- Looking for wordexp
-- Looking for wordexp - not found
-- Looking for wordexp.h
-- Looking for wordexp.h - not found
-- Looking for sys/xattr.h
-- Looking for sys/xattr.h - found
-- Looking for catopen
-- Looking for catopen - found
-- Looking for dlfcn.h
-- Looking for dlfcn.h - found
-- Looking for features.h
-- Looking for features.h - found
-- Looking for filehdr.h
-- Looking for filehdr.h - not found
-- Looking for getpwuid
-- Looking for getpwuid - found
-- Looking for IDtouser
-- Looking for IDtouser - not found
-- Looking for nl_types.h
-- Looking for nl_types.h - found
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Looking for pthread_mutexattr_t
-- Looking for pthread_mutexattr_t - not found
-- Looking for pthread_mutex_timedlock
-- Looking for pthread_mutex_timedlock - not found
-- Performing Test HAVE_PTHREAD_MUTEX_ERRORCHECK
-- Performing Test HAVE_PTHREAD_MUTEX_ERRORCHECK - Failed
-- Performing Test HAVE_PTHREAD_MUTEX_RECURSIVE
-- Performing Test HAVE_PTHREAD_MUTEX_RECURSIVE - Failed
-- Performing Test HAVE_DLADDR
-- Performing Test HAVE_DLADDR - Success
-- Looking for _PC_CASE_SENSITIVE
-- Looking for _PC_CASE_SENSITIVE - not found
-- Looking for FNM_CASEFOLD
-- Looking for FNM_CASEFOLD - found
-- Looking for KDMKTONE
```

```
[ 71%] Building CXX object CMakeFiles/rexx.dir/interpreter/platform/unix/SysFileSystem.cpp.o
[ 71%] Building CXX object CMakeFiles/rexx.dir/interpreter/platform/unix/SysInterpreterInstance.cpp.o
[ 71%] Building CXX object CMakeFiles/rexx.dir/interpreter/platform/unix/SysRexxUtil.cpp.o
[ 72%] Building CXX object CMakeFiles/rexx.dir/interpreter/platform/unix/SystemCommands.cpp.o
/home/tjk/coding/oorexx/oorexxsvn/main/trunk/interpreter/platform/unix/SystemCommands.cpp:874:9: error: '
    posix_spawn_file_actions_init(&action);
    ^
/home/tjk/coding/android-ndk-r22b/toolchains/llvm/prebuilt/linux-x86_64/sysroot/usr/include/spawn.h:51:44
typedef struct __posix_spawn_file_actions* posix_spawn_file_actions_t;
                                         ^
/home/tjk/coding/oorexx/oorexxsvn/main/trunk/interpreter/platform/unix/SystemCommands.cpp:892:13: error:
    posix_spawn_file_actions_adddup2(&action, input[0], 0); // stdin reads from pipe
    ^
```

Android Bionic

- POSIX Support limited
- API Level 28 (Android 9, “PIE”) added 70 Functions
- No Support for glibc Functions so far: crypt(), wordexp

4.1. CMake Output & Problems 2

```
[ 84%] Linking CXX executable bin/rxapi
[ 84%] Built target rxapi
[ 84%] Building CXX object CMakeFiles/rexximage.dir/utilities/rexximage/rexximage.cpp.o
[ 85%] Linking CXX executable bin/rexximage
[ 85%] Built target rexximage
[ 86%] Generating bin/PlatformObjects.orx
[ 86%] Generating bin/CoreClasses.orx
[ 86%] Generating bin/StreamClasses.orx
[ 86%] Generating lib/rexx.img
/bin/sh: line 1: /home/tjk/coding/oorexx/oorexxsvn/main/trunk/bin/rexximage: No such file or directory
make[2]: *** [CMakeFiles/rexx_img.dir/build.make:77: lib/rexx.img] Error 127
make[1]: *** [CMakeFiles/Makefile2:487: CMakeFiles/rexx_img.dir/all] Error 2
make: *** [Makefile:156: all] Error 2

coding/oorexx/oorexxsvn/main/trunk via Δv3.28.3 took 1m20s
```

Solution:

- Move rexximage onto Device
- Execute it on the Device
- Move rexx.img into *lib*/rexx.img

PART 5 | Status and Outlook

5.1. Status Quo & Outlook

5.2. Discussion

5.1. Status Quo & Outlook

- Nearly functional Build for Android
- Outlook Bachelor Thesis
 - Nutshell Examples with working Android Build
- Outlook beyond
 - Business Programming Students are able to build Android Apps with ooRexx and BSF4Rexx (Java)

5.2. Discussion

- Questions?