



Breathe ideas.

M3 – System Concepts

Android Property

- Value used to describe current system status and configurations
- getprop/setprop

```
root@android:/ # getprop
[ARGH]: [ARGH]
[dalvik.vm.heapsize]: [48m]
[dalvik.vm.stack-trace-file]: [/data/anr/traces.txt]
[init.svc.adbd]: [running]
[init.svc.bootanim]: [running]
[init.svc.console]: [running]
...
```

- C/C++ - cutils/properties.h
LOCAL_SHARED_LIBRARIES += libcutils
- Java android.os.SystemProperties

Service

- Background application
- C/C++ - used to perform long background operations, usually as system app (daemon)
- Java – High level operations that doesn't require user interaction

Init

- First process that is started in the system (Linux)
- Starts other services through rc script
- Enables
 - Service startup
 - Setting properties (prop)
 - Basic OS file system operations (mkdir, chmod, ...)
 - Performing operations based on different startup phases (boot, early boot, fs init, ...)

Binder

- Communication mechanism between two separate processes
- Defined with client-server architecture
- Solution is not specific to one programming language:
 - C/C++ - C/C++ (native – native)
 - C/C++ - Java (native – high lvl)
 - Java - C/C++ (high lvl – native)
 - Java – Java (high lvl - high lvl)
- Parcel
- aidl
- **LOCAL_SHARED_LIBRARIES += libbinder**

Exercise 1

- Implement exercises from practicum

Upload to Canvas:

- Source code of native application that is setting rtrk.android property and screenshot as confirmation it is working as intended.
- Source code of Android (Java) application that is setting rtrk.android property and screenshot as confirmation it is working as intended.
- Source code of exd native application and screenshot as confirmation it is working as intended.
- init.goldfish.rc file, logcat and screenshot that shows exd is automatically started by init process.

Exercise 2

- Implement exercises from practicum

Upload to Canvas:

- Source code of binder server and client written in C/C++, logcat and screenshot as confirmation it is working as intended.
- Source code of binder server written in C/C++ and client written in Java which are exchanging basic data types, logcat and screenshot as confirmation it is working as intended.

Exercise 2 – optional

- Implement exercises from practicum

Upload to Canvas:

- Source code of binder server written in C/C++ and client written in Java which are exchanging complex data types, logcat and screenshot as confirmation it is working as intended.
- Source code of binder server written in C/C++ and client written in Java which are exchanging a callback, logcat and screenshot as confirmation it is working as intended.

Ashmem

- Binder supports maximally 1MB of data to be transferred in one session
- Binder supports 15 clients max
- Ashmem offers a mechanism to share memory between two processes which can be used for sharing data

JNI

- JNI is a Java interface mechanism that allows access to native code
- Keyword native is used to mark JNI methods to the compiler
- type Java_full_package_name_class_method (JNIEnv *, jobject, params...);
 - type – type of return value, jint, jvoid, jlong...
 - full_package_name – full package name where . is replaced with _
 - class – name of Java class in application
 - method – name of method inside previous class
 - JNIEnv *, jobject – mandatory parameters
 - params – list of arguments based on types known to JNI compiler
 - jclass javaLocalClass = env->FindClass("com/package/name/Example");

Exercise 3

- Implement exercises from practicum

Upload to Canvas:

- Source code of ashmem C/C++ server and C/C++ client, logcat and screenshot as confirmation it is working as intended.

Exercise 3 - optional

- Implement exercises from practicum

Upload to Canvas:

- Source code of ashmem C/C++ server and Java client, logcat and screenshot as confirmation it is working as intended.

Exercise 4

- Implement exercises from practicum

Upload to Canvas:

- Source code of shared library and Android JNI application, logcat and screenshot as confirmation it is working as intended.

Exercise 4 - optional

- Implement exercises from practicum

Upload to Canvas:

- Source code of Android Java application, JNI library and native service that implements ashmem mechanism, logcat and screenshot as confirmation it is working as intended.

NIT

Breathe ideas.

Contact us

Institute for information technologies NIT

Radnicka 30a, 21000 Novi Sad, Serbia

info@nit-institute.com

www.nit-institute.com

+381 64 01 64 724