

LO52 A2018 - Examen final

Consignes :

- Seul le support de cours fourni par l'enseignant est **autorisé**
- Vous pouvez utiliser une copie si vous le souhaitez
- Vous pouvez répondre directement sur la feuille

1 Partie 1 : QCM (5 points, 0,25 points par question)

1. Quel est l'équivalent de la commande ***make clean***?
 - (a) make clobber
 - (b) `rm -rf out/target/*`
 - (c) `rm -rf out/`
 - (d) make mrproper
2. **Pourquoi** utilise-t-on la compilation croisée pour les **systèmes embarqués** ?
 - (a) Car ça fait classe auprès des copains de Génie Mécanique
 - (b) Car ce type de système n'a généralement que très peu d'espace de stockage disponible
 - (c) Car le processeur est plus lent que sur une station de travail
 - (d) Car on ne souhaite pas installer les outils de développement sur la cible
3. Dans une fonction JNI, **deux paramètres** sont obligatoires :
 - (a) Un pointeur sur JNIEnv et un jint
 - (b) Un pointeur sur JNIEnv et un jobject
 - (c) Un pointeur sur JNIEnv et un jstring
 - (d) Un jobject et un jstring
4. Laquelle ou lesquelles de ces affirmations est/sont vraie(s) ?
 - (a) Dans Android, "tout" est "package"
 - (b) Une librairie tierce-partie est un composant
 - (c) WebKit est ce qu'on appelle un NAOS (Non Android Open Source Software)
 - (d) On distingue les composants dits Internes des composants Externes
5. Comment appelle t-on les **services asynchrones pouvant être stoppé par n'importe qui**?
 - (a) Les Async Services
 - (b) Les Started Services
 - (c) Les Bounded Services
 - (d) Aucune de ces réponses. Un Service ne peut que être synchrone sur Android

6. Laquelle ou lesquelles de ces affirmations est/sont vraie(s) ?
- (a) repo est l'outil pour la récupération du code source Android
 - (b) repo est écrit en Python
 - (c) repo permet d'aggréger les commandes Git
 - (d) repo permet de faire des différentiels sur les différents projets Git de l'AOSP
7. Laquelle de ces réponses n'est pas un outil pour vérifier la compatibilité Google ?
- (a) Le CDD
 - (b) Le CDI
 - (c) Le CTS
 - (d) L'AOSP
8. Quel outil permet de faire des "code review" ?
- (a) Garrit
 - (b) Gerrit
 - (c) Code Collaborator
 - (d) Jira
9. Quel gestionnaire de sources est utilisé pour l'AOSP ?
- (a) Subversion
 - (b) Git
 - (c) Mercurial
 - (d) Clearcase UCM
10. Sous quelle forme se présente la configuration du Kernel ?
- (a) Un fichier .config avec un contenu de type "clé=valeur"
 - (b) Un fichier .Kconfig avec un contenu de type "clé=valeur"
 - (c) Un fichier config.kernel avec un contenu de type "clé=valeur"
 - (d) Un fichier .config avec un contenu de type "clé;valeur"
11. Une PlatformLibrary est :
- (a) Une librairie Java
 - (b) Une archive Java brute
 - (c) Associée à un fichier de permissions XML
 - (d) Un composant sans contraintes ou restrictions particulières
12. L'option permettant de définir la Kernel Command Line dans la configuration du noyau s'appelle :
- (a) CONFIG_KCMDLINE
 - (b) CONFIG_CMD_LINE
 - (c) CONFIG_CMDLINE
 - (d) CONFIG_K_CMDLINE

13. Le répertoire des sources où se trouve les **applications** s'appelle :
- (a) apps
 - (b) packages
 - (c) frameworks
 - (d) prebuilt
14. Le répertoire **res** dans une application Android peut contenir :
- (a) des animations
 - (b) des "9-patch"
 - (c) des fichiers bruts (images)
 - (d) des labels
15. **GCC** signifie :
- (a) GNU Compiler Collection
 - (b) GNU Code Collection
 - (c) GNU Compiler Computer
 - (d) GNU Code Computer
16. Laquelle **de ces directives** permet de produire un fichier .jar ?
- (a) BUILD_JAVA_SHARED_LIBRARY
 - (b) BUIL_JAVA_STATIC_LIBRARY
 - (c) BUILD_JAVA_LIBRARY
 - (d) BUILD_JAR_LIBRARY
17. Laquelle ou lesquelles de ces affirmations est/sont fausse(s) ?
- (a) Une application Android vit dans un contexte isolé du système
 - (b) Une application Android a son propre user/group
 - (c) Une application Android utilise l'unique instance de Dalvik/ART du système
 - (d) Une application Android a seulement accès à ce dont elle a besoin
18. Laquelle de ces valeurs ne représente pas **l'état d'une activité** ?
- (a) Killed
 - (b) Paused
 - (c) Running
 - (d) Started
19. En **quelle année** a été créé le Kernel ?
- (a) 1989
 - (b) 1990
 - (c) 1991
 - (d) 1992

20. Quel fichier d'un produit décrit les directives liées au matériel ?
- (a) BoardOfDirectors.mk
 - (b) BoardConfiguration.mk
 - (c) BoardConfigCommon.mk
 - (d) BoardConfig.mk

2 Partie 2 : Exercice "Android dans la nouvelle Tesla d'Elon Musk" : TesAA (15 points)

Vous voilà aujourd'hui en ST50 au sein du fabricant "**Tesla**" qui souhaite mettre en place un système Android sur leur nouvelle collection de nouvelles voitures connectées qui va intégrer les dernières technologies de Android Auto (désigné par AA).

Ce système embarqué pour voiture intégrera un ensemble de composants classiques tels que lecteur de Musique **TesMusic**, un lecteur de vidéo **TesVideo**, un GPS **TesGPS**...

La carte à partir de laquelle vous allez travailler est une réalisation spécifique au projet "**TesAAuto**" dont le nom de référence est "**tesaauto**" mais dont l'essentiel des composants électroniques et des caractéristiques sont les mêmes que la **Google AAuto (device aauto)** pour une architecture **32 bits**.

Votre employeur vous précise que vous travaillerez à partir des sources d'Android Oreo de chez **Google**. Les différents fichiers concernant la Google AA se trouvent sous l'arborescence **/device/google/aauto** :

- *vendorsetup.sh*
- *AndroidProducts.mk*
- *BoardConfigAAuto.mk*
- *BoardConfigAAuto_64.mk*
- *aauto_common.mk*
- *aauto_common_64.mk*
- *full_aauto.mk*
- *full_aauto_64.mk*

Consignes :

- Vous définirez votre nouveau produit sous l'arborescence **device/tesla/tesaauto**
- Celui-ci **héritera** de la **Google AAuto** dans chaque fichier où cela sera possible
- On souhaite **trois** types de cibles : une cible dite **engineering**, une cible dite **userdebug** et une cible **user**
- Tous les composants externes iront directement dans le répertoire **device/tesla/tesaa/external** sous la forme :
 - **libtaa-<nom>** pour la librairie
 - **taa-<nom>-bin** pour les binaires/services
 - **libtaa-<nom>-jni** pour les interfaces JNI
 - et **taa-<nom>-fw-ext** pour les extensions de framework
- Les "Makefiles" Android des composants iront directement à la racine des répertoires cités précédemment
- Pour copier directement un fichier dans l'image finale de votre système, on utilise la directive **PRODUCT_COPY_FILES**(:= ou +=₄ selon le cas) **path_fichier_source:path_final**.

5. On cherche à remplacer les fichiers suivants pour notre cible sans altérer le code source d'Android :

- *strings.xml*
- *layout.xml*
- *menus.xml*

Sachant que l'on travaille pour une collection de voitures qui ira dans un premier temps sur les marchés américain et canadien et dont l'écran le plus petit est 640dp x 480dp . Cette résolution d'écran est considérée comme grande. Les fichiers indiqués au dessus sont présents dans les répertoires suivants (sous **frameworks/base/core/res/res/**) :

- **drawable-small-en-nodpi/**,
- **drawable-normal-en-nodpi/**,
- **drawable-large-en-nodpi/**,
- **drawable-xlarge-en-nodpi/**,
- **drawable-xxlarge-en-nodpi/**,
- **drawable-small-fr-nodpi/**,
- **drawable-normal-fr-nodpi/**,
- **drawable-large-fr-nodpi/**,
- **drawable-xlarge-fr-nodpi/**,
- **drawable-xxlarge-fr-nodpi/**,
- **drawable-small-zh-nodpi/**,
- **drawable-normal-zh-nodpi/**,
- **drawable-large-zh-nodpi/**,
- **drawable-xlarge-zh-nodpi/**
- et **drawable-xxlarge-zh-nodpi/**

(a) Quel(s) **mécanisme(s)** va être utilisé pour réaliser cette opération de surcharge ? (0,5 points)

```
-- Réponse --
```

```
-- Réponse --
```

(b) Quelles **ressources** vont être impactées par ce mécanisme ? Expliquez comment. (0,5 points)

```
-- Réponse --
```

```
-- Réponse --
```

(c) Donnez la ligne que vous allez ajouter à votre fichier **tesaa.mk** : (1 point)

```
-- tesaa.mk --
```

```
-- tesaa.mk --
```

6. Le prestataire de services a fourni une application de lecture audio **TesMusic.apk** pour le système TesAAuto comme vous lui avez demandé dans les spécifications externes. Vous avez décidé de copier celle-ci sous **packages/apps/** sur votre système Android durant l'étape de production de votre système.

(a) Où allez-vous stocker cette application dans les sources ? (0,5 points)

```
-- Réponse --

-- Réponse --
```

(b) Quelle ligne allez-vous ajouter au **Makefile** de votre produit ? (0,5 points)

```
-- tesaa.mk --

_____ \
_____ :_____
-- tesaa.mk --
```

7. Celui-ci souhaite aussi que vous personnalisiez entre autres les propriétés **tesm.vol_min**, **tesm.vol_max** avec les valeurs recommandées par le constructeur du chipset "**0x00 pour la valeur minimale et 0xff pour la valeur maximale**". Quelle(s) ligne(s) allez-vous ajouter à votre **Makefile** de produit ? (0,5 points)

```
-- tesaa.mk --

_____ \
_____ =_____ \
_____ =_____
-- tesaa.mk --
```

8. Vous allez maintenant intégrer une librairie externe pour votre application TesGPS (librairie partagée) **libgpst**. En suivant les consignes (**nommage, localisation...**) données en préambule de l'exercice et en sachant que :

- Les fichiers sources présents sont *libtaa-gpst/main/main.c*, *libtaa-gpst/src.c*, *libtaa-gpst/external/threads* et *libtaa-gpst/external/threads_x64.c*
- Les en-têtes se situent dans les répertoires *libtaa-gpst/inc*, *libtaa-gpst/ext/inc*, *libtaa-gpst/ext/inc_x64* ainsi que le fichier *libtaa-gpst/main/config.h*
- Il faut compiler la librairie avec le **CFLAG NOPOSIX_THREADS**

(a) Quel est le fichier dont vous allez avoir besoin absolument pour produire votre librairie ? (0,5 points)

```
-- Réponse --

-- Réponse --
```

(b) Complétez son contenu :(1,5 points)

```
-- Makefile --  
  
_____  
_____  
_____ $(CLEAR_VARS)  
  
_____  
_____ := \  
_____  
_____  
_____  
  
_____  
_____  
_____  
_____  
_____  
  
LOCAL_C_FLAGS _____ - _____  
  
_____  
_____ libtaa-gpst  
  
_____  
  
LOCAL_PRELINK_MODULE:= false  
  
_____  
  
-- Makefile --
```

(c) Complétez la ligne du Makefile du produit (1 point)

```
-- tesaa.mk --  
  
_____  
_____ \  
libtaa-gpst  
  
-- tesaa.mk --
```

9. Vous allez maintenant écrire une JNI qui va répondre aux critères donnés en énoncé et aussi :
- La librairie JNI se nommera **libtaa-gpst-jni**
 - Le fichier contenant les fonctions JNI s'appellera **taa-gpst-jni.c**
 - Celle-ci implémentera les **fonctions suivantes** de la **libtaa-gpst** :
 - *void init()*
 - *byte[2] coord get_coord(int xy)*
 - *byte[2] coord set_coord(int xy, byte[2] new)*
 - *void reset_coord(int xy)*
 - Le nom de package à obtenir pour une extension de framework est **us.taa.gpst.jni** avec la classe **TaaGPS**. Les noms de méthodes sont libres.

(a) Complétez le fichier source : (1,5 points)

```
-- taa-gpst-jni.c --
...
#include _____
...

//Prototype function init
_____
_____

...

//Prototype function get_coord
_____
_____

...

//Prototype function set_coord
_____
_____

...

//Prototype function reset_coord
_____
_____

...

-- taa-gpst-jni.c --
```

(b) Complétez le contenu du Makefile :(1,5 points)

```
-- Makefile --
_____
_____
_____ := \
    taa-gpst-jni.c
_____ \
    _____ \
    _____ \
    _____ \
    _____ libcutils \
    _____ libutils \
    _____ libtaa-gpst-jni
LOCAL_PRELINK_MODULE:= false
_____
-- Makefile --
```


- (b) Ce service doit être ajouté au script d'init qu'un collègue va vous fournir pour intégration dans la définition de votre produit. Complétez les blancs suivants dans le fichier d'init :(1 point)

```
-- Fichier d'init --  
  
service _____  
    _____ taa-user-vid  
    _____ taa-class-lib  
    _____ TAA-VID-CODEX x264  
    _____ //service essentiel au système  
  
-- Fichier d'init --
```