

# LO52 A2017 - Examen final

## Consignes :

- Support de cours fourni par l'enseignant autorisée uniquement
- 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. Quelle est la première commande à exécuter lors de la production d'un système Android ?
  - (a) man aosp
  - (b) build/envsetup.sh
  - (c) lunch
  - (d) make
2. Quel(s) bus de messages est utilisé principalement dans Android ?
  - (a) Binder
  - (b) D-Bus
  - (c) ActiveMQ
  - (d) <troll>Aucun, les bus de messages c'est mal</troll>
3. Lequel de ces types de build n'en est pas un :
  - (a) Cross-compilation
  - (b) La compilation cross-native
  - (c) La compilation cross-canadienne
  - (d) La compilation canadienne
4. Quelle est la signification de l'acronyme CDD pour Android ?
  - (a) Contrat à Durée Déterminé pour les employés de Google France
  - (b) Compatibility Definition Document
  - (c) Compatibility Document Definition
  - (d) Compatibility Definition for Dummies
5. De quelle(s) manière(s) peut-on passer des arguments au Kernel ?
  - (a) A travers le bootloader
  - (b) En utilisant l'option CONFIG\_CMDLINE
  - (c) A travers un fichier de la partition boot
  - (d) Aucune des réponses ne convient

6. Laquelle de ces commandes **n'existe pas** ?
- (a) mm
  - (b) m&ms
  - (c) m
  - (d) mmm
7. En **quelle année** Android a été rachetée par Google ?
- (a) 2000
  - (b) 2004
  - (c) 2005
  - (d) 2006
8. Quelle(s) **commande(s) fastboot** permet de **tester le Kernel** passé en paramètre ?
- (a) fastboot flash boot boot.img
  - (b) fastboot ktest bzImage
  - (c) fastboot boot bzImage
  - (d) fastboot flash :raw boot bzImage
9. Laquelle de ces réponses ne correspond pas à un **composant** d'une application Android ?
- (a) Des Services
  - (b) Des Activités
  - (c) Des Content Providers
  - (d) Aucune, toutes les réponses précédentes sont justes
10. Quel est le **nom d'une extension du framework** sous Android ?
- (a) Framework Extension
  - (b) Framework Library
  - (c) Platform Library
  - (d) Platform Extension
11. Quel **répertoire** contient le code relatif au bootloader sous l'AOSP ?
- (a) boot
  - (b) bootloader
  - (c) u-boot
  - (d) bootable
12. Laquelle(s) de ces affirmations est(sont) vraie(s) ?
- (a) WebKit est un composant de l'AOSP
  - (b) Un exécutable est un composant de l'AOSP
  - (c) L'application Galerie est un composant de l'AOSP
  - (d) Bionic est un composant de l'AOSP

13. Quel(s) est(sont) le(s) types de services pour une application Android
- (a) Les Bounded Services
  - (b) Les Bonded Services
  - (c) Les Started Services
  - (d) Les Bounded Started Services
14. Laquelle(s) de ces commande(s) n'existe(nt) pas ?
- (a) make kernel install
  - (b) make install
  - (c) make modules install
  - (d) make all install
15. Quelle est la signification de AOSP ?
- (a) Android Open Source Project
  - (b) Adroid Operating System for Profit
  - (c) Android Open Source for Profit
  - (d) Android Operating System Project
16. Laquelle ou lesquelles de ces variables d'environnement sont obligatoires pour la compilation croisée du Kernel ?
- (a) ARCH
  - (b) GCC
  - (c) LD\_LIBRARY\_PATH
  - (d) CROSS\_COMPILE
17. Quelle(s) est(sont) parmi les réponses suivantes, les solutions pour stocker les données d'une application Android ?
- (a) Les Shared Preferences
  - (b) Les Internal Storage
  - (c) Les External Storages
  - (d) Les Databases
18. Laquelle de ces commandes logcat permet de n'afficher que les logs type Error de l'activité LO52 ?
- (a) logcat LO52 :\*
  - (b) logcat LO52 :5 \* :0
  - (c) logcat LO52 :E \* :S
  - (d) logcat -s \* LO52 :E
19. Comment peut-on intégrer un fichier ou binaire ou librairie pré-compilée dans l'AOSP ?
- (a) En créant un Android.mk avec la directive BUILD\_PREBUILT
  - (b) En faisant une demande auprès de Google
  - (c) En utilisant la directive PRODUCT\_COPY\_FILES

(d) En invoquant un shaman vaudou car vous ne vous rappelez plus du cours

20. Quel fichier d'un produit décrit les directives liées au matériel ?

- (a) BoardConfig.mk
- (b) Hardware.mk
- (c) BoardConfigCommon.mk
- (d) HAL2001SpaceOdyssey.mk

## 2 Partie 2 : Exercice "Android dans la nouvelle montre Samsung Product "Gear Fit" (15 points)

Vous voilà aujourd'hui en ST50 au sein du fabricant "**Samsung**" qui souhaite mettre en place un système Android sur leur nouvelle collection de montres connectées destinées aux personnes âgées.

Ce système va permettre d'ajouter des fonctionnalités de santé, notamment à travers de nouveaux capteurs afin de surveiller l'état de santé des personnes et alerter les autorités compétentes en cas de danger pour elle.

La carte à partir de laquelle vous allez travailler est une réalisation spécifique au projet "**Gear-FIT**" dont le nom de référence est "**gearfit**" mais dont l'essentiel des composants électroniques et des caractéristiques sont les mêmes que la **Samsung Gear Live (device sprat)**.

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

- *vendorsetup.sh*
- *AndroidProducts.mk*
- *BoardConfigSprat.mk*
- *BoardConfig.mk*
- *sprat\_common.mk*
- *full\_sprat.mk*
- *CleanSpec.mk*

### Consignes :

- Vous définirez votre nouveau produit sous l'arborescence **device/samsung/gearfit**
- Celui-ci héritera de la **Samsung Gear Live** dans chaque fichier où cela sera possible
- On souhaite **deux** types de cibles une cible dite **engineering**, une et une cible **user**
- Tous les composants externes iront directement dans le répertoire **external** sous la forme :
  - **libgfit-<nom>** pour la librairie
  - **gfit-<nom>-bin** pour les binaires/services
  - **libgfit-<nom>-jni** pour les interfaces JNI
  - et **gfit-<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 += 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 :

- *default\_wallpaper.jpg*
- *main.xml*
- *strings.xml*

Sachant que l'on travaille pour une montre connectée qui sera destinée pour le marché américain et que ces fichiers 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-es-nodpi/**,
- **drawable-normal-es-nodpi/**,
- **drawable-large-es-nodpi/**,
- **drawable-xlarge-es-nodpi/**
- et **drawable-xxlarge-es-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 **gearfit.mk** : (1 point)

```
-- gearfit.mk --
```

```
-- gearfit.mk --
```

6. Le prestataire de services a fourni un firmware pour le capteur cardiaque **cardio\_fw.bin** comme vous lui avez demandé dans les spécifications externes. Vous avez décidé de copier celle-ci sous **system/xbin/** sur votre système Android durant l'étape de production de votre système.

(a) Où allez vous stocker ce firmware dans les sources ? (0,5 points)

```
-- Réponse --

-- Réponse --
```

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

```
-- gearfit.mk --

_____ \
_____ : _____
-- gearfit.mk --
```

7. Celui-ci souhaite aussi que vous personnalisiez entre autres les propriétés **hw.bpm\_min**, **hw.bpm\_max** avec les valeurs recommandées par l'OMS "**80 BPM pour la valeur minimale et 140 pour la valeur maximale**". Quelle(s) ligne(s) allez vous ajouter à votre **Makefile** de produit ? (0,5 points)

```
-- gearfit.mk --

_____ \
_____ = _____ \
_____ = _____
-- gearfit.mk --
```

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

- Les fichiers sources à compiler sont *libgfit-diab/main/config.c* *libgfit-diab/src.c* et *libgfit-diab/external/external.c*
- Les en-têtes se situent dans les répertoires *libgfit-diab/inc* *libgfit-diab/ext/inc* ainsi que le fichier *libgfit-diab/main/config.h*
- Il faut compiler la librairie avec le **CFLAG** *EXTERNAL\_LIBC\_SUPPORT*

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

```
-- Réponse --

-- Réponse --
```



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

```
-- gfit-diab-jni.c --
...
#include _____
...

//Prototype function init
_____
_____

...

//Prototype function get_status
_____
_____

...

//Prototype function set_status
_____
_____

...

//Prototype function get_sensor_value
_____
_____

...

-- gfit-diab-jni.c --
```

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

```
-- Makefile --
_____
_____
_____ := \
  gfit-diab-jni.c
_____ \
  _____ \
  external/libgfit-diab/inc \
  external/libgfit-diab/ext \
  external/libgfit-diab/main
_____ libcutils \
  libutils \
  _____
_____ libgfit-diab-jni
_____
LOCAL_PRELINK_MODULE:= false
_____
-- Makefile --
```

10. Vous allez maintenant mettre en place une extension de framework s'appuyant sur les fonctions de la JNI implémentées à la question précédente (en terme de noms de classe, de méthodes et de package).

(a) Voici le Makefile Android pour cette extension de framework (partie Java). A vous de remplir les trous : (1,5 points)

```

-- Makefile --
_____
_____
_____ := \
  _____, java)
_____ gfit-diab-fw-ext
_____
_____
-- Makefile --

```

(b) Voici la classe Java associée à cette extension de Framework. Complétez les trous : (1 point)

```

-- Classe Java --
package _____;

public class _____ {
    static {
        _____;
    }
    _____
    _____
    _____
    ...
}
-- Classe Java --

```

(c) Voici maintenant le XML de permissions nécessaire que vous devez compléter (0,5 points) :

```

-- gfit-diab-fw-ext.xml --

<?xml version="1.0" encoding="utf-8"?>
<permissions>
  <library name="_____"
    file="_____" />
</permissions>

-- gfit-diab-fw-ext.xml --

```



- (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 _____  
    _____ gfit-user  
    _____ gfit-class  
    _____ GFIT_BLOOD_PRESSURE 0  
    _____ //service essentiel au système  
  
-- Fichier d'init --
```