

[Système d'exploitation pour l'embarqué](https://e-uapv2020.univ-avignon.fr/mod/bigbluebuttonbn/view.php?id=61206)

Tp: Construction d’un système linux embarqué complet

Realiser par : Rosa Hadroug

Encadré par :SILANUS Marc

**Objectif du Tp** :

L’objectif de ce TP est de réaliser une distribution complète avec mise en œuvre d'un capteur opérationnel dès sa mise sous tension.

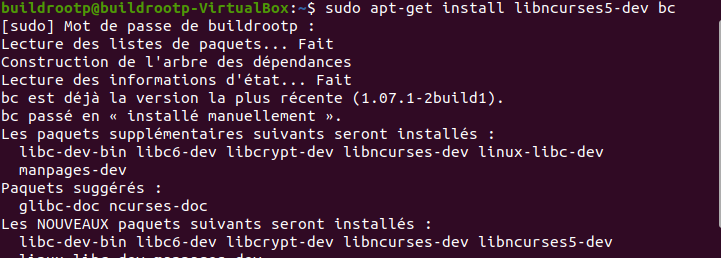
**5 Raspberry pi - le matériel**

<https://en.wikipedia.org/wiki/Raspberry_Pi>

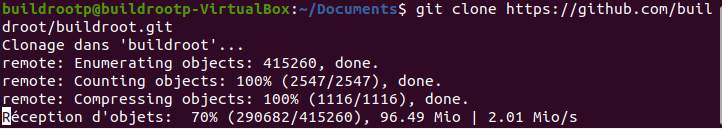
**6 Buildroot pour Raspberry Pi 2 et QEMU**

**6.1 Installation de l’environnement Buildroot**

Installation des dépendances nécessaires à l'utilisation de Buildroot :



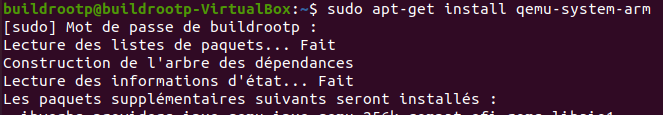
Clonage des sources de l'outil **Buildroot** dans le dossier personnel



**6.2 Construction d’un système de base**

Génération de la première image minimale de GNU/Linux à partir de **Buildroot**.

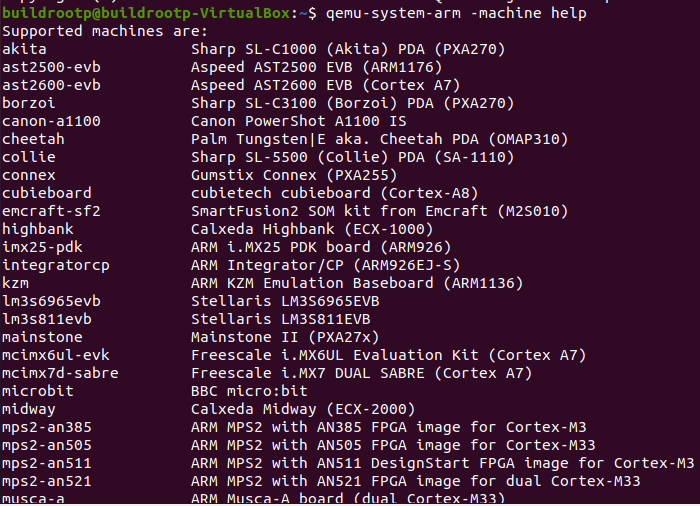
**6.2.1 Installation de QEMU**

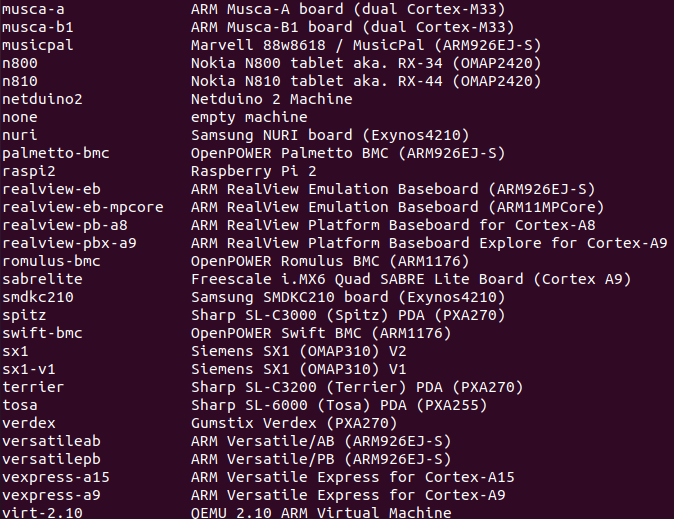


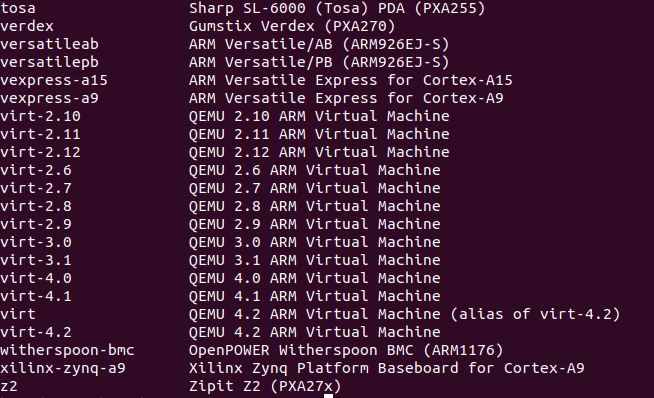
la version de qemu dont je dispose est bien supérieure à 2.4

4

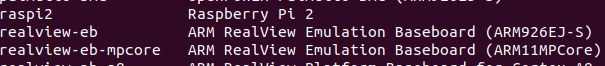
Listez l'ensemble des plateformes (machines) supportées par QEMU.





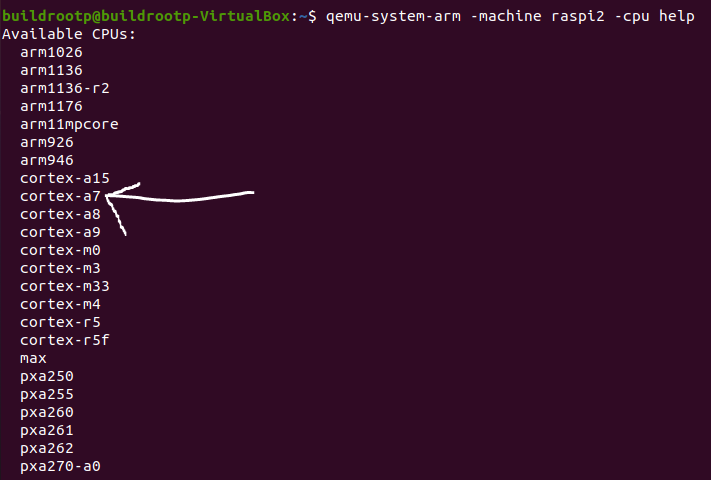


Assurez vous de la prise charge de la plateforme Raspberry pi 2

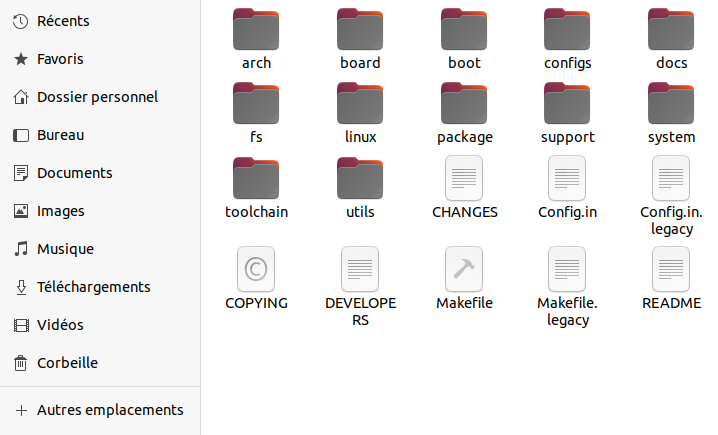


Listez les processeurs supportés par l'émulateur pour cette plateforme.

Assurez vous que le processeurs qui équipe le Raspberry pi 2B est bien supporté par l’émulateur Cortex-a7 est bien dans la liste des processeurs



**6.2.2 Génération de la première image**

****

les actions à mener pour construire un système

1) On execute la commande **'make menuconfig'** ;

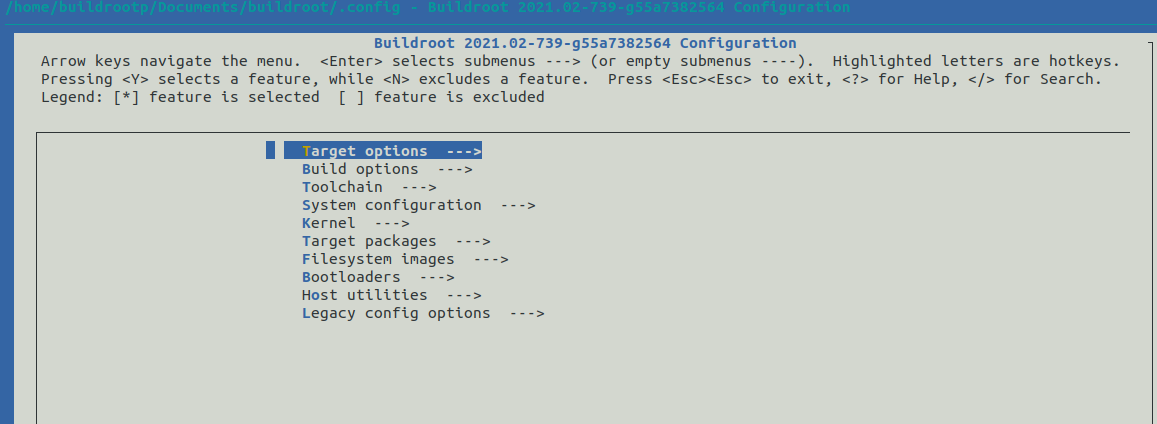
2) Sélectionnez l'architecture cible et les paquets que nous souhaitons compiler ;

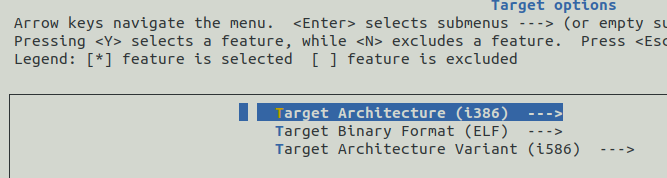
3)On execute la commande **'make'** ;

4) Attendre la fin de la compilation ;

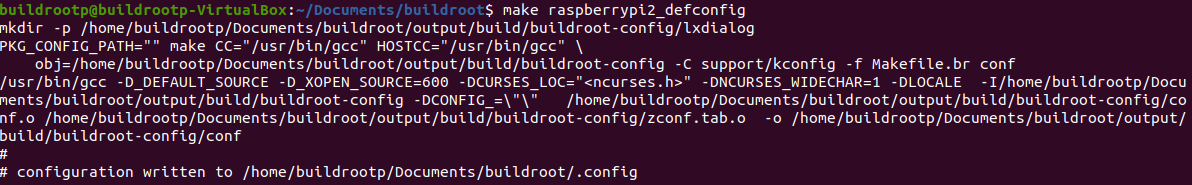
5) Trouver le noyau, le chargeur de démarrage «bootloader», le système de fichiers «filesystem»racine

Consultez le menu de configuration de buildroot par défaut, notamment **Target options**.





Appliquez le fichier de configuration standard pour le Raspberry Pi 2 :



les différentes étapes à mener pour compiler et déployez le système sur

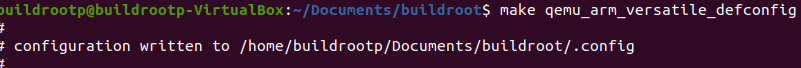
une carte micro-SD.

On configure **Buildroot**: pour notre cas raspberry pi model 2 B « **make raspberrypi2\_defconfig** »

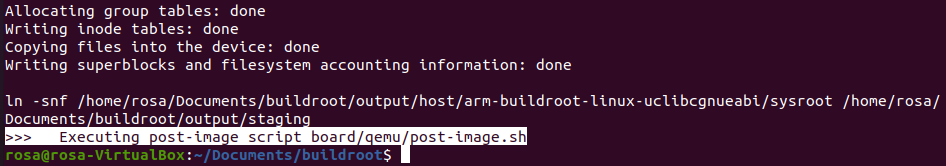
On construit le file system racine ( rootfs) « **make ».**

Il reste maintenant de déployer le système sur une carte micro-SD

Nous n’allons pas compiler un système pour Raspberry mais pour l’émulateur QEMU. Le fichier de configuration standard n’est donc pas le même :

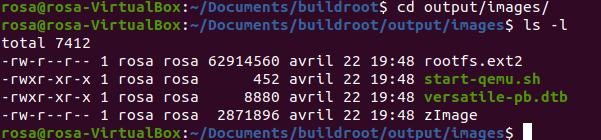


Lancez la compilation du système



À l'issue de la génération, vous obtenez 2 fichiers dans le dossier output/images/.

Le noyau correspond au fichier **zImage** et le système racine au fichier **rootfs.ext2**



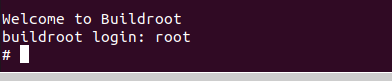
la taille du noyau est 2,8 Mo.

21

La taille du système de fichier est 4,5 Mo.

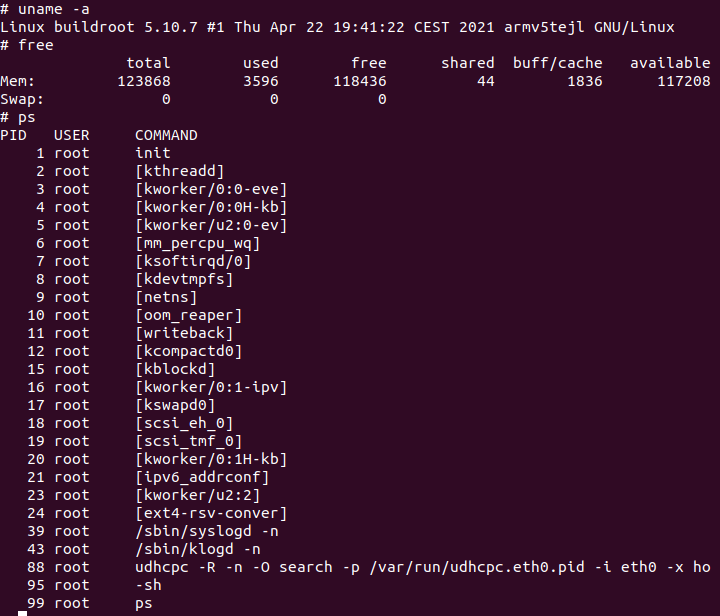
20

Démarrez le système dans **QEMU** à l’aide du script généré par **Buildroot** :

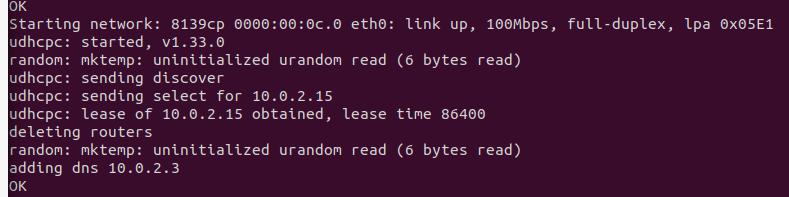


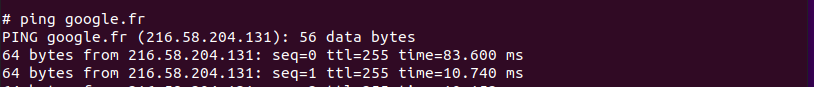
la durée de boot est de deux a trois minutes environs

la version du noyau linux

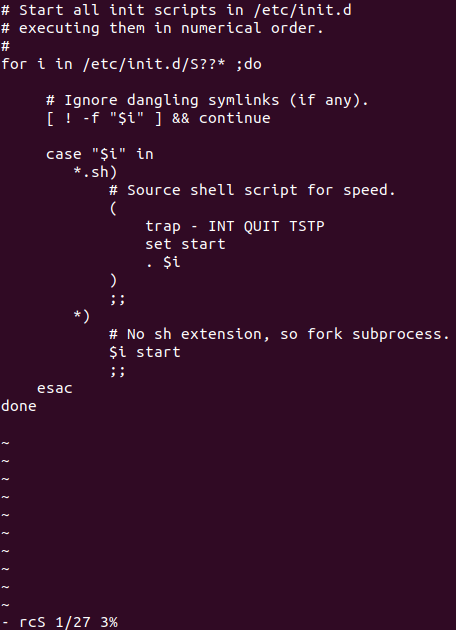


Connexion réseau,





Consultez le script /etc/init.d/rcS et expliquez la procédure de démarrage du système

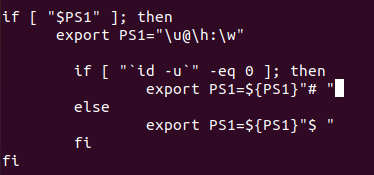


Explication

**6.3 Amélioration du système**

**6.3.1 Prompt du shell global**

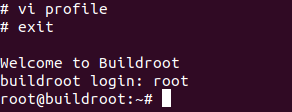
Modifiez le fichier **/etc/profile** comme ci-dessus.





Redémarrer la cible et observez l’allure du prompt quand vous changez de

dossier.



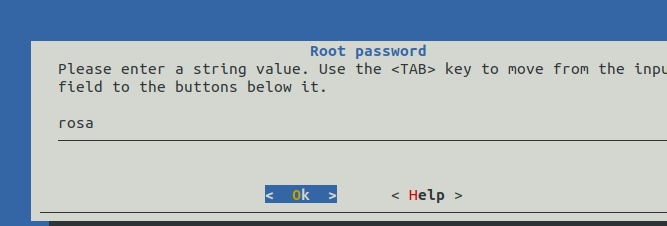
if [ "`id -u`" -eq 0 ]; then ...

**6.3.2 Accès à distance**

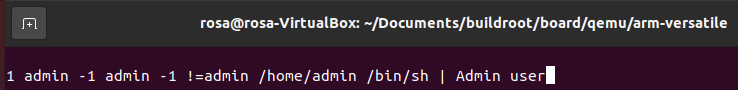
**Activer le openSSH** :

25

Sécurisez l'accès au système en attribuant un mot de passe à root :



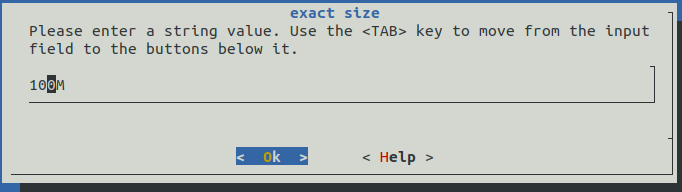
Créez dans le dossier **board/qemu/arm\_versatile** un fichier **adduser.txt**



Indiquez à Buildroot la présence de ce fichier pour créer notre utilisateur

29

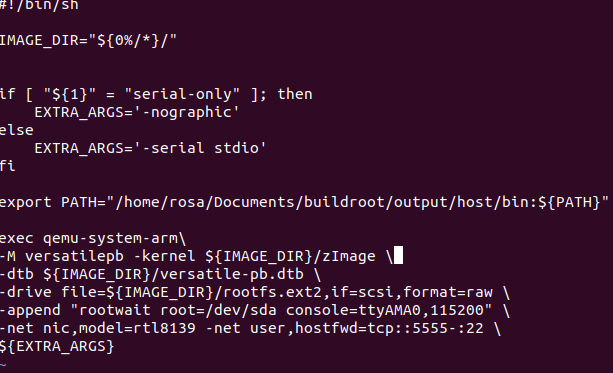
Modifiez la taille exact du système de fichier à 100Mo



Relancez le processus de construction

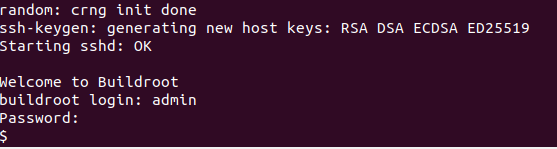
33

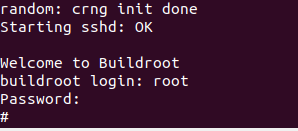
Modifiez le script start-qemu-net.sh



Observez les effets de la reconstruction sur les modifications réalisées

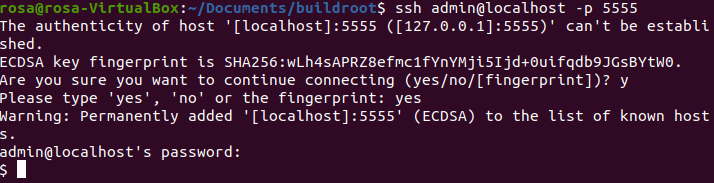
antérieurement





Temps de démarage est plus long

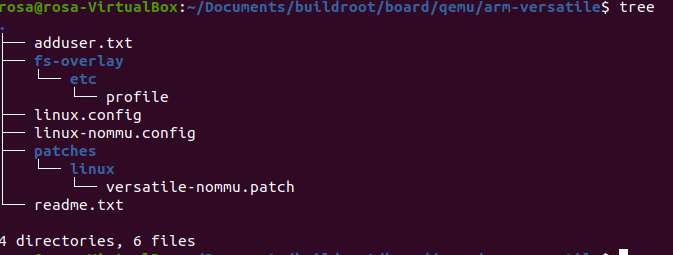
Accédez à la cible depuis votre PC hôte via une session ssh.



**6.3.3 Persistance des modifications sur rootfs**

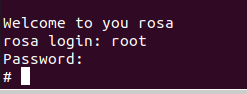
Créez le dossier de recouvrement fs-overlay et copiez-y le script

précédents assurant la bonne configuration du prompt.



Ajoutez au système un nom d'hôte et une bannière d'accueil

41

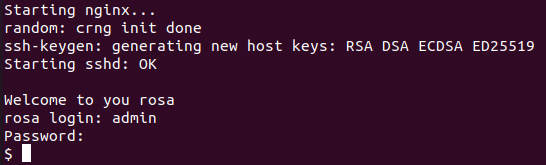


**6.3.4 Serveur web**

installer nginx:

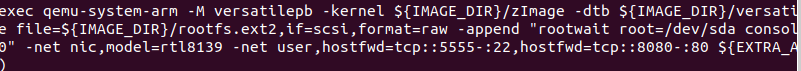
43

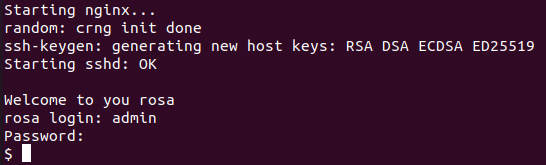
Relancez le processus de construction.



Modifiez le script de démarrage start-qemu-net.sh pour ajouter une

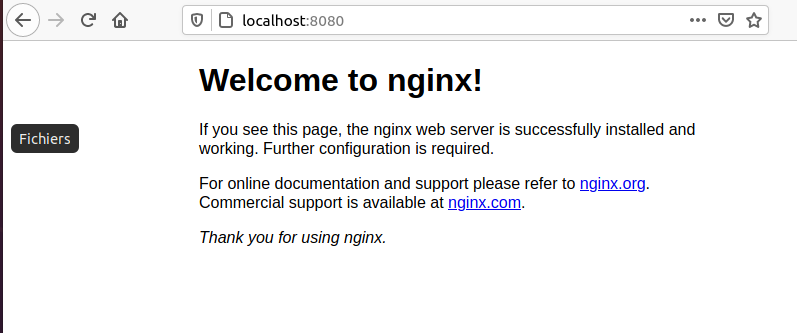
redirection du port 80 vers le port 8080



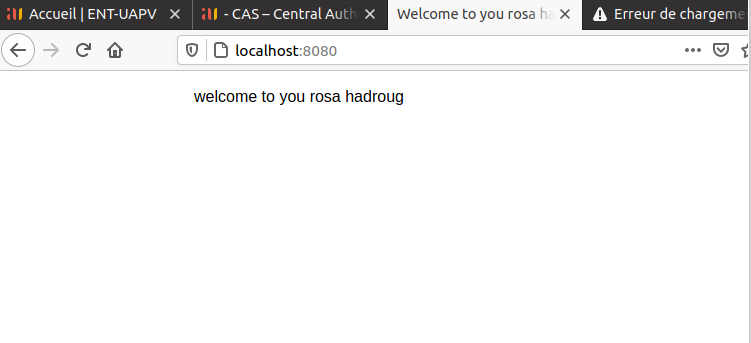
****

Redémarrez la cible accédez au service web depuis votre hôte en utilisant

l’URL **http://localhost:8080**

****

Proposez une page index.html qui contient vos nom et prénom.

****

**7 Image finale**

Compressez le dossier **images** qui doit contenir les fichiers suivants :