Marc-Henri Kasses
M2 MIC

Compte rendu TP3

Temps estimé : 1journée
Temps réel : 1journée (TP non terminé)

Je n'ai pas pu joindre les fichiers sur lesquelq j'ai travaillé, le fichier .tar est trop volumineux...

Configurer le système

• Indiquez comment vous avez procéder pour effectuer les configurations ci-dessus Donnez le « chemin » de sélection dans le menu de configuration.

- ✗ Pour générer un noyau pour 32 bits, l'option était présente à l'ouverture du menu de configuration
- ✗ Pour définir le suffixe de la version du système :
  General Setup > Local version - append to kernel release
- ✗ Pour activerel'option qui permet d'utiliser un initram :
  General Setup > Initial RAM filesystem and RAM disk (initramfs/initrd) support
- ✗ Pour printk :
  General Setup > Configure standard kernel features (expert users) > Enable support for printk
- ✗ Pour activer le support par le noyau Linux des binaires exécutables ELF et des shell scripts :
  Executable file formats / Emulations > Kernel support for ELF binaries / Kernel support for scripts starting with #!
- ✗ Pour le support de 8250/16550 : Device Drivers > Character devices > Enable TTY > Serial Drivers > 8250/16550 and compatible serial support
  Ajouter l'option Console on 8250/16550 and compatible serial port pour la console

• Où se trouve la documentation du noyau Linux dans l'arborescence que vous avez installée ?

Compilez

• Combien de temps a pris votre compilation (donnez des précisions sur l'environnement de génération)?

Marc-Henri Kasses
M2 MIC TP3

La compilation a duré environ 2 minutes. Je suis sur une machine 64 bits avec ubuntu d'installé dessus.

• Où se trouve le fichier généré, quelle taille fait-il? Le noyau Linux est généré sous différentes formes. Trouvez les et décrivez (très brièvement) leurs formats respectifs.
Le fichier créé est arch/x86/boot/bzImage de 832592 octets. Il existe aussi .bzImage.cmd qui est beaucoup plus petit (151 octets) .

• Si on avait généré le noyau Linux pour notre machine de développement, que faudrait-il faire ensuite pour « installer » ce noyau et tenter de redémarrer notre machine avec notre nouveau noyau? (Donnez les quelques commandes nécessaires).

• Quelles différences avez-vous trouvé entre la 4.7.7 et la version précédente ? Quelles sources d'information avez-vous utilisées ?

QEMU

• Quel message s'affiche à la fin de l'initialisation du système Linux ? Pourquoi ?
Le message affiché est : « not syncing : No working init found. Try passing init= option to kernel. », probablement car il n'y a aucun programme à effectuer au lancement de la machine.

Bonjour le monde!

• Quelle différence y a-t-il entre qemu-i386 et qemu-system-i386 ? Pourquoi faut-il utiliser qemu-i386 pour « hellos » et qemu-system-i386 pour le noyau Linux ?
Qemu-i386 lance une architecture i386 sur la machine et qemu-system-i386 lance l'architecture dans un autre systeme

• Listez l'arborescence obtenue (format -l)

• Quelle commande avez-vous utilisée pour générer votre programme « init »? Pourquoi ?
gcc -Wall -Wpointer-arith -m32 hello.c -o hello_32 -static

• Pourquoi faut-il faire une édition de liens statique ? • Donnez le résultat de la commande file sur votre programme init / hello.
$ file hello_32
hello_32: ELF 32-bit LSB executable, Intel 80386, version 1 (GNU/Linux), statically linked, for GNU/Linux 2.6.32, BuildID[sha1]=6bae87ed65de23173586769452e5af0fb8576413, not stripped

• Quelle est la taille de cette commande init /hello? Par quelle(s) commande(s) avez-vous trouvé cette information? • Quelles sont les tailles de ses segments de code et de données, sur disque et en mémoire? Par quelle(s) commande(s) avez-vous trouvé ces informations?

• A quelle adresse en mémoire virtuelle le segment de code sera-t-il placé? Et le segment de données? Et la pile? Par quelle(s) commande(s) avez-vous trouvé ces informations ?

Une fois le « disque » fabriqué

• Quelles erreurs éventuelles avez-vous rencontrées? Pourquoi? Comment avez-vous résolu ces problèmes?
Il ne se passait rien au lancement, donc j'ai pris votre .config, mais je n'ai pa réussi à utiliser l'option initramfs.

• A quoi sert l'argument « -append » de Qemu? Que peut-on passer comme valeurs à cet argument? Où trouver cette information?
-append cmdline
Use cmdline as kernel command line
(trouvé grace a man qemu-system-i386)

• Que se passe-t-il quand votre programme « init » se termine après avoir affiché « Hello World! »? Pourquoi?

Un peu de dynamisme!

• Quelle commande avez-vous utilisée pour générer votre programme « init »?

• Donnez le résultat de la commande file sur votre programme init / hello.
• Quelle est la taille sur disque de cette commande init /hello? Par quelle(s) commande(s) avez-vous trouvé cette information?
• Quelles sont les tailles de ses segments de code et de données, sur disque et en mémoire? Par quelle(s) commande(s) avez-vous trouvé ces informations?
$ gcc -Wall -Wpointer-arith -m32 hello.c -o hello_32_dyn

$ file hello_32_dyn
hello_32_dyn: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=d0ef32a6ada2d5b1a853e76c5e1656ac58d9d980, not stripped

$ size hello_32_dyn

text	data	bss	dec	hex	filename
1202	280	4	1486	5ce	hello_32_dyn

• Donnez le résultat de la commande ldd.
$ ldd hello_32_dyn
linux-gate.so.1 => (0xf7716000)
libc.so.6 => /lib32/libc.so.6 (0xf7542000)

/lib/ld-linux.so.2 (0x56617000)

• Indiquez quelles bibliothèques vous avez copié dans votre arborescence root.

Une petite compilation croisée

• Qu'affiche la commande file ./hell_arm ?

• Pourquoi -static à la compilation pour cette compilation croisée pour ARM ?

• Que se passe-t-il lors de la première tentative d'exécution ?

• Pourquoi la dernière exécution ./hello_arm se comporte-t-elle différemment de la première ?

• Incluez un appel à pause dans votre hello.c, recompilez et invoquez la commande ps (manuellement) quand ./hello_arm est coincé dans le pause. Donnez le résultat dans votre compterendu

---

BusyBox

• Avez-vous rencontré des problèmes? Comment les avez-vous résolus?
Je n'ai pas réussi à utiliser Busybox

• Quelle séquence de commandes avez-vous essayé sur votre machine QEMU?

• Quelle taille fait votre fichier binaire exécutable BusyBox? Quelle est la taille de sa section de code? La taille de sa section de données, sur disque, en mémoire?

• Pourriez-vous configurer busybox de manière à ce que cette commande ait une taille plus réduite que le /bin/bash de votre machine de développement?