

Plan de cours Linux embarqué

Durée : 20 heures (dépend de la maîtrise des pré-requis)

Niveau : intermédiaire (en fin d'étude de cycle ingénieur)

Pré requis : connaître le monde *NIX (des connaissances en Linux Desktop sont très appréciables), la programmation C, la problématique Temps Réel et embarquée

Contenu :

Introduction

1. GNU/Linux dans l'embarqué et le temps réel
2. Pourquoi le libre ?
3. Les licences
4. Considérations juridique
5. Obligations relatives à la GPL
6. « Closed source Linux kernel modules are illegal »
7. Sur la LKML
8. Le faux problème sous Linux
9. Ils ont joué et perdu
10. La GPLv3
11. Relations des entreprises avec l'aspect juridique

Linux

1. GNU/Linux, les distributions
2. Le noyau Linux
3. Les points fort de Linux
4. Historique
5. Modèle de développement
6. Dans l'embarqué
7. Le choix
8. L'existant
9. Les cibles
10. Linux (really) everywhere
11. Kernel vanilla pour l'embarqué
12. La solution « facile »
13. La difficile construction d'une distribution
14. Le packaging et la gestion à long terme
15. Collaboration commercial/communautaire

Présentation de projets

1. uClinux
2. Buildroot
3. OpenEmbedded
4. Les environnements pré-intégrés
5. Montavista
6. RTAI/Xenomai
7. Xenomai

8. Projets commerciaux « clefs en main »

Au coeur du système

1. Organisation générale
2. À la racine
3. L'espace mémoire
4. Le kernel
5. Organisation générale du noyau
6. Mémoire virtuelle : processus
7. Mémoire virtuelle : pagination
8. Comment choisir ses options/modules/supports
9. Ordonnanceur
10. Les formules magiques
11. Arguments au boot
12. Don't panic
13. Les drivers
14. Gérer les modules externes
15. Les devices
16. Dans /proc
17. Dans /sys
18. Inspecter le kernel
19. Boot du système
20. Dans /etc
21. Exploration de /etc/fstab
22. Dans /sys
23. Sécurité
24. Iptables for dummies
25. Soyez parano

Outils

1. Busybox
2. Serveurs
3. Java
4. Chiffrement
5. Autres softs pour l'embarqué
6. Outils pour intégrer

Mise en pratique

1. Cross-compiling
2. ARM : EABI et Thumb
3. NFS
4. File systems
5. Gagner de la place
6. Partitionner, formater et mapper
7. Générer une image ou un firmware
8. Boot loader
9. Das u-boot project
10. Générer une image ou un firmware
11. Les étapes à suivre
12. RS232 ou UART
13. GDB et gsbsserver
14. Tftp
15. Dhcp/bootp
16. Flasher
17. Le travail de l'ingénieur

Bonus !

1. Gcc mastering
2. Techniques de programmation
3. Communication interprocessus
4. Flash
5. Paquets
6. Paravirtualisation
7. Émulation
8. CGI
9. Cygwin
10. Plate-forme de dev commerciales
11. Construire un SDK maison
12. Les petits rien
13. BSD
14. Conseils finaux

Conclusion

1. Avantages
2. Inconvénients
3. Futur proche et lointain

Bibliographie

1. Linux embarqué
2. Building Embedded Linux Systems
3. Embedded Linux Primer
4. Linux kernel
5. Linux device drivers
6. Programmation système en C sous Linux
7. Programming Embedded Systems
8. Architecture de l'ordinateur
9. Linux Administration
10. Sur le web
11. En kiosque

Au boulot !

1. Stages
2. Emplois
3. Droit à la formation