MA-CSEL1 (construction systèmes embarqués sous Linux)

Objectifs de l'examen oral

Conditions d'examen :

a. Après le tirage au sort d'une enveloppe contenant deux questions, l'examen oral se déroulera en deux phases :
   - une phase de préparation de 20 minutes et
   - une phase de présentation de 20 minutes.

b. Durant la phase de préparation, l'étudiant-e préparera les réponses aux questions sur transparents. Pour sa préparation, l'étudiant-e aura le droit à un résumé de 5 pages maximum sous forme papier. Les portables, tablettes ou smartphones ne sont pas autorisés.

c. Durant la phase présentation, l'étudiant-e exposera ses solutions au collège de professeurs au rétroprojecteur et répondra aux questions subsidiaires. Ces questions permettront au collège de professeur d'établir le niveau de connaissance et de compétence de l'étudiant-e. Durant la présentation, l'étudiant ne pourra utiliser que les transparents rédigés précédemment dans la phase de préparation.

d. A la fin de l'examen, les questions et les transparents seront récupérés et conservés par le collège de professeurs.

e. La liste d'objectifs ci-dessous n'est pas exhaustive. Il est possible que lors de l'examen des aspects supplémentaires relatifs à la matière examinée soient ajoutés.

Les étudiant-e-s devront être capables :

Environnement et infrastructure

Environnement Linux embarqué

1.1 De décrire et de mettre en œuvre l'environnement de développement (installation, logiciels et outils) sur la machine hôte

1.2 De décrire la structure d'un système embarqué sous Linux en mode développement et production

1.3 De décrire et de mettre en œuvre les différentes techniques proposées pour le débogage d'applications

1.4 De décrire la procédure de mise en production des applications (lancement des applications, génération du rootfs) et de déployer un système embarqué complet sous Linux

1.5 De décrire l'architecture générale de systèmes embarqués sous Linux.

MA-CSEL1 (construction systèmes embarqués sous Linux)

Objectifs de l'examen oral

Bootloader et systèmes de fichiers

2.1 De décrire le rôle d'un bootloader, de citer les principaux et de citer les facteurs à considérer lors de son choix
2.2 De décrire les caractéristiques principales de l'U-Boot
2.3 De décrire le démarrage de l'U-Boot sur la cible NanoPi NEO Plus2
2.4 De décrire le démarrage de Linux depuis la mémoire flash et depuis le réseau
2.5 De décrire les différents types de systèmes de fichiers pour les systèmes embarqués et de les mettre en œuvre
2.6 De concevoir une organisation de la mémoire flash pour une distribution Linux adaptée aux besoins spécifiques du système à déployer

Programmation noyau

Modules noyaux

3.1 De décrire les modes d'opération et l'organisation de la mémoire sous Linux
3.2 De décrire le concept de module noyau (avantages/limitations, structure/squelette, débogage, génération, installation/désinstallation)
3.3 De concevoir un module noyau complexe (paramètres noyaux, allocations dynamiques, accès aux entrées/sorties, threads, accès concurrents, mise en sommeil, gestion des interruptions)

Pilotes de périphériques

4.1 De décrire les différents types de pilotes de périphériques sous Linux et les mécanismes permettant de les accéder
4.2 De décrire et de concevoir un pilote orienté caractère (étapes d'implémentation, opérations, échange de données, mécanisme de réservation du numéro de pilote, fichier d'accès, mécanisme d'enregistrement/libération, accès bloquants)
4.3 De décrire et de concevoir un pilote orienté mémoire dans le noyau et dans l'espace utilisateur (avantages/inconvénients, opérations)
4.4 De décrire et de concevoir une interface de configuration basée sur SYSFS

Programmation système

Système de fichiers

5.1 De décrire le traitement d'erreurs sous Linux
5.2 De décrire la philosophie des systèmes d'exploitation Unix et Linux
5.3 De décrire le système de fichiers virtuels sous Linux (fichiers ordinaires et spéciaux)
5.4 De décrire les appels système pour gérer des fichiers

MA-CSEL1 (construction systèmes embarqués sous Linux)

Objectifs de l'examen oral

5.5 De décrire l'architecture et les différences entre les bibliothèques File-IO et Standard-IO
5.6 De contrôler des périphériques et des ressources au travers de fichiers spéciaux
5.7 De gérer ders répertoires et leur annuaire sous Linux
5.8 De décrire le principe de surveillance de changements sur des fichiers avec inotify
5.9 De décrire le principe de multiplexage des entrées/sorties
5.10 De décrire les accès bloquants et non bloquants
5.11 De traiter le multiplexage des entrées/sorties avec « select » et/ou « epoll»
5.12 De décrire le principe de fonctionnement des signaux (levée, capture, traitement, …)

Multiprocessing

6.1 De décrire la problématique de la programmation multitâches (processus – thread)
6.2 De citer les avantages/désavantages des processus par rapport aux threads
6.3 De décrire le modèle et les états d'un processus
6.4 De décrire le principe de clonage de processus et de création de threads
6.5 De décrire le principe de daemon
6.6 De décrire la problématique d'accès concurrent et les mécanismes d'exclusion mutuelle
6.7 De proposer des solutions pour éviter les accès concurrents
6.8 De décrire la communication interprocessus et les mécanismes IPC (pipes, fifo, etc.)
6.9 De décrire le principe de fonctionnement des 2 ordonnanceurs (CFS et RTS)
6.10 De décrire la priorisation des processus
6.11 De décrire les stratégies SCHED_NORMAL, SCHED_RR et SCHED_FIFO
6.12 De proposer des critères pour la conception d'une application normale ou temps réel
6.13 De décrire les mécanismes permettant d'attribuer un ou plusieurs CPU à un processus
6.14 De décrire la gestion des ressources des µP avec les groupes de contrôle (cgroup)

Optimisation

Performances

7.1 De décrire différentes méthodes pour mesurer le temps d'exécution
7.2 De décrire les méthodes de mesure de performances
7.3 De citer les principaux outils disponibles sous Linux pour mesurer les performances
7.4 De décrire les outils de profiling et de mesure de la couverture de code
7.5 De décrire les caractéristiques principales de l'outil perf
7.6 De décrire quelques méthodes d'optimisation de performances