Introduction

* linux howa il kernel
* GNU mye7ebouch yesta3mlouh fil embarqué
* Problème howa les carte (raspberry) sont hétérogène it is hard to found system that be compatible with. It must be compatible with source for example arm Risc 5
* objectif is to minimize the time and to optimize the functionality
* we need calculator and he is not created we must build it
* compilator natif : ‌compilator that create Executable and run in the some machine
* ‌cross compilator : compilator create file .bin that work with embedded system (arm) but it use source of pc to build
* ‌the first thing that we must create it is the bootloader (intermediate between the resources and the kernel ) for embedded U-Boot and he make the initiation for the memory , charge the kennel of os
* user space collection of library that make the system utilisable illi howa il GNU FIL pc ama a7na bech ne5demo bil buzy box
* ‌the best tools that facilate the tasks is Yocto
* ‌linux 93d fil license GPL version 2 bech y93d yesta3mlouh fil embedded w android bech ynjmou les entreprises ybedlo fil kernel
* ‌fi wireless router yst3mlo fih linux pour avoir un firewall pour filtrer les packet
* ‌processeur support MMU et NO-MMU (memory management unit) composant qui faire la traduction de adresse des mémoires virtuelle vers des adresses physiques en quelques instructions assembleur ... Itha n7eb n5adem Linux 3ala microcontrôleur lezmni nsna3 w7di il MMU ala5tir il microcontrôleur msnou3 bech ye5adem programme we7id ili hya mhyech performante
* ‌1) spécialité BSP ( board support package développement) garantie que le bootloader et le noyau marche bien
* ‌2) spécialité il intègre des applications personnalisé géré les mise à jour
* ‌3) spécialité il développe les applications mais il utilise des compilator croisés pour produire le binaire

‌Cross compiling toolchains

* ‌ bech nsna3 il compilator it need 3 machines Host-Triplet(build , host, target )

Build : where the toolchains is build

Host : where the toolchains will be executed

Target : where the binaires created by the toolchains are executed

* ‌4 common build types are possible for toolchains

one is not need

B = H = T native build

( B = H ) # T Cross build

B # ( H = T ) Cross-native build

B # H # T Canadian cross build (rasp - risc)

( B = T ) # H

* ‌andi machine mnjem nbeddel fiha chy nejem ninstalli des programmes en locale nejem ncopilih ama masbob fiha GCC 8 ama n7eb nsta3ml GCC 9 mfemech version binaire lil architecture mte3 mekina w il GCC 9 open source Solution : ncompili GCC 9 Bi GCC 8 ne5o il code source mte3 GCC 9 W nconfigirihom 3ala GCC 8 hya performante ama njemchi nzid
* ‌Est que je peut libérer la taille de l'espace occupé par le contrôleur GCC 9 : Ay hya fil 79i9a il Compilator GCC 9 illi sna3neha fil asl mcompiliya bil GCC 8 So the solution is to compile the final version with GCC 9
* il verification ili bech ye3melha hya enno y3ayet lil rou7o bech yverfi ili ki ycompili lezm ye5raj rou7o
* ‌bech nesne3o il compilator croisés GCC 9 est que il suffit de ne5o il configuration mte3 GCC 9 w na3tihom lil un GCC 8 bech ne5rej GCC 9 résultats optimale wla NON ala5tir il GCC 9 mcompili bil GCC 8
* ‌bech nejem nesna3 compilator croisés lezmna nsne3o compilator native
* ‌Cross build lezmo native build GCC 9

( X86 -> ARM ) => GCC 9 ( X86 -> X86 ) => GCC9( X86 -> ARM )

* ‌Cross-native build lezmo native build w cross build

GCC 9 ( ARM -> ARM ) => Cross build GCC 9 ( X86 -> ARM ) => GCC 9 ( ARM -> ARM )

* ‌Canadian cross build hya cross build

GCC 9 ( ARM -> R5) => Cross build GCC 9 ( X86 -> ARM ) => GCC 9 ( ARM -> R5 )

* ‌le compilator est un des constituants d'une chaîne de constituants croisé ( gcc w7do mykafich mhyech il Compilator )
* 1 . Cpp ( c pré processor ) hello.c -> hello01.c fiha kounchi il code C example #include tt7awel lil contenu ili mwjouda fi .h

2 . CC1 helloc01.c te5arej assembly itha nativ nafso ili fil pc wla tweli lil target

3 . as hello.s -> hello.o kil3ada keno nativ wla le5ir

4 . Ld hello.s -> hello.c

* ‌Binutils howa projet a part du compilator fih des commande deriture des binaire et des commande de lecture des binaire
* ‌la différence entre l'espace noyau et l'espace utilisateur : l'espace noyau howa ili 3ndo contact direct me3a il hardware w 3ndo l'accès 3ala des instructions assembleur mhomech mwjoudin fil espace utilisateur (root) -- restriction pour E U
* ‌ili min jihet il user space hya API ( application programming interface ) w ili min jihet il kernel ili fiha registre w assembleur ttsama ABI ( application binary interface)
* ‌win mte3ayet lil fonction fiha communication avec le matériel la bibliothèque C va la traduire en appelle système
* ‌mathal Samsung 3ndha teliphonet Android w illi howa Linux ili fih des bibliothèques C ken jew y7to license LGBL v 3 m3netha ya3tiw lil utilisateur bech yflashiw ili y7ebo 3lih heka 3leh ken Linux y7awel lil GPL v3 mfemema 7ata we7id mech yesta3mlo
* ‌ken bech nsné3o bil Yocto lezm nsta3mlo musl C library

‌Bootloader

* ‌3lech nte3adew par plusieurs étapes 3lech mich il matériel ychargi il système d'exploitation illi howa bootloader
* ‌bootloader : logiciel responsable du 4 tâches fundamental pour charger le noyau

1) initialisation du matériel : uniquement les matériels principaux pour fonctionner le noyau

2) charger un binaire : noyau soit min flash disk ...kima ynej ykoun mich noyau (bootloader)

3) possibly decompress

4) exécuté le noyau

* ‌sequence de démarrage d'un pc classique 2000

+ Bios from ROM : detecter les périphériques

+ Il Bios yemchi lil awel support de stockage ye3rtho wili généralement howa il disk dur w ye5rej mino il 512 octets w yermihom fi ram w ye3taberhom program w yexecutihom. Kol 512 octets ttsama sector fil disk dur w awel 2 sector ytsamew MBR ( Master boot record ) . 512 octets très peu. En ne peut pas passer le 4 partition et blein de problème.

Heka 3leh Apple san3et EFI fi 3wth il bios plus performant : presque partions illimité ect..

+Utilisé un autre endroit program w hetha il program howa il bech ymchi ya9ra w y3ml il execution binaire Lil jme3a mte3 Linux Gub 2 howa a7senbwe7id tw presque howa système d'exploitation mnjmouch nste3mlouh fil embarqué ala5tro volumineux

* ‌for the embedded system it is the some as the classic pc .
* ‌example mte3 carte min micro-ship 3ndha ROM boot ili hya ta3ml 7aja bark thib il binaire bism bien précis t7eto mich fi ram mzelet sekta t7eto fi s-ram fi west il processeur ili taille mte3ha mtfoutech il 4 k octets il question 3leh mye7touch kolo fil ROM alat5ir bech ynjmou y3mlo mise a jour be3dika nte3adew lil UBoot ili bech ye5dem il ram w il disk w il jaw il kol
* ‌fil cas mte3 il raspberry kina5demo il carte awel 7aja te5dem normalement hya il cpu or no awel 7aja te5dem hya une petite module fil GPU mini core risc yemchi ylawej 3aala fichier ismo bootcode.build fil FAT32 ychargih fil cache L2 alat5ir il ram mzelet sekta w yexecutih il mini core risc tw saret il initialisation mte3 il ram. yemchi yejeb looader.bin bech y5dem il Cpu Arm. Cheft hethom il fichier mnejmouch n3mlo 7atta interaction bech nbedlou fihom mnjmouch nsobo UBoot donc il 7all howa fi 3awth mna3tiwah il Linux kernel n3tiwah Uboot w n9ololo heka howa il kernel mte3ik wa9tha njemo njibo il noyau linux min réseaux min disk dur

‌Kernel Linux

* ‌La manière dont le processeur Intel voie les périphériques : Ports Mapped Devices w intel fih des instructions binaire ili bech ye3mlo il read w write w il method lot5ra ili yst3mlouha fil architecture Arm hya memory mapped Devices fil les deux méthodes fama restriction fil userspace 7ata kita3ml assembly code w t7eb te3ml read w write fil les adresses hekom fil asl des adresses virtuelle w il MMU howa il ytranstihom adresse dans la ram tw wlet il intel te5dem bil memory mapped Devices witha il userspace y7rb y accedi lil périphériques lezmo ydemandi il system call
* ‌kernel linux a 3 fonctionnalités de base :

- manage all the hardware resources

- set of portable

- handle concurent access and usage

* ‌system call : we take example of function in the userspace that wanna use function from the kernel space (not possible) ... Pour appeelle une fonction on doit nécessairement un pil pour devenir très simple a faire. Heka mnin jet il impossible ala5tir me3ndhomch nfs il pile . Donc wlina nlwjo 3ala support partagé entre le user w kernel ( illi howa la seul ) registre ye5dmo yama bil compteur yama bil pointeur . Whya fil asl mtji kounchi bil les interruptions bech tinterrumpu 3ala utilisateur ili hya int80 b3d ye5dem il MMU ymchi ye5adem il code assembleur w b3d yerj3 lil userspace ykml . M3ndich il7a9 bech nbeddel fil numéro mte3 il system call.
* ‌fama folder ismo proc linux 3mlo bech ye3ml exposition des fichiers et des dossiers lil user ili howa mhowech mwjoud fil disk dur virtuel homa nwemir (pid) w njm nbeddel fih

‌TD

* ‌cross compiling toolchains outil ( cross tool ng)
* ‌Bootloader U-Boot
* ‌contruire le noyau Linux
* ‌espace utilisateur avec buzy box
* ‌quelque configuration