

* Session 2 *

* بعد اول Session فتدقيقها في كشكول ال RTOS كتبها هنالك

* ال Session دي هنعمل ال linux distribution بتاعتها على ARM

Steps :-

1- اول حاجة احنا محتاجين نعمل install
emulator شان هنشغل على ARM
gemu-system-arm

`Sudo apt install gemu-system-arm`

* في عندك verstile board دي gemu
Supports و دي ال احنا هنشغل
عليها

2- احنا محتاجين نعمل ال kernel بتاعتنا
فمحتاجين cross comp ال ARM نعمله بي
ال kernel شان نطها على ال verstile

`Sudo apt install gcc-arm-linux-gnueabi`

٣- محتاج انزل ال Kernel اللى قشغل عليها ، البشوهه من حمل ال session الهمه على 6.7.6 ، انا مذكرا السبب ولفيت ال Version اللى موجود 6.7.7 فكارى حوار ال Version ده

ال Kernel قشيلك Compressed فى شكل tar ، ده زي ريز او rar بس بد linux فمحتاج تخملاها untar

tar xvf linux-6.7.7.tar.xz

٤- تعالى نعمل Compile ال Kernel ، بس رقم (١) ال Kernel فيها Features و Drivers كثير اوى ، ده كل واحد بيكمل driver يجع بيرفع ، فاحنا محتاجين نعمل الاول Configuration ال Kernel شات تختار ال Features اللى انت كايزها

لو جيت رخت جوا untared Kernel بتاعتك هتلاقى Folders كثيره زي :-

ده جوا كل ال architectures → arch *
ال Supported جوا ال Kernel زي RISC و ARM و x86

make O=. /build/ ARCH=arm
CROSS_COMPILE=arm-linux-gnueabi
versteilte defconfig

كل رة Command واحد، تعالى نترجم واحدة واحدة

big 0 not zero

O=. /build/ →

انت رلوقتى رخت جوال Folder الى فيه
ال Kernel، فانت محتاج تعمل Folder و
تخط فيه ال Files الى ال make هيطلعهم،
Folder ا build الى فيه ال o/p و
لومش عندك اكمليت ولى فيه ال o/p

ARCH=arm →

ال architecture الى ر build ليها ولى arm

CROSS_COMPILE=arm-linux-gnueabi :-

انت جانا بتحدد ال Compiler بتاعك، by default
بيكون ال gcc الى هو موجود عندك كادى كالجهاز
ليس انت عايزت Compile ل ARM، فحددت ال cross
Compiler بتاعك ل gcc-arm-linux-gnueabi

verbatile - defconfig → board

* هتلا في في hidden file 1 - Config
اتعمل بتاع 3200 طر فيه ال Configurations
بتاع ال kernel كلها

How to edit kernel Configurations:-

* مهم انت عندك حين لو عايز تعدل
في ال Configurations، يا تستخدم ال Vim و
تعدل في ال Config. وده مبدئ ووش

في بقى ال Semi-GUI tool كده
menuConfig بتقدر تعدل فيها في
ال kernel Config. بال GUI ش ال BIOS
كده اها menuConfig

```
make O=/build/ ARCH=arm CROSS_COMPILE=  
= arm-linux-gnueabi-menuConfig
```

٢- تكال بقى نكمل build لل Kernel بعد
ما عملنا ال Configurations بتاعتها

عشان احرف الوقت

```
time make O=/build/ ARCH=arm  
CROSS_COMPILE=arm-linux-gnueabi-
```

Parallel processes

اللي ب build و في

نفس الوقت

الوقت احسن build kernel
محتاج build modules
الى هي محتاج

time make -j8 O=../build/ ARCH=arm
CROSS_COMPILE=arm-linux-gnueabi-modules

v- خذ modules
install
root file system build

directory الى كانت
time make -j8 O=../build/ ARCH=arm

INSTALL_MOD_PATH = rootfs

absolute path
linux kernel
Kernel بناتق هو جود

arch build الى
arm boot
Image او Compressed
zImage

ARM boot
File less
executable zImage
little endian

١- خلاص بقى تعالى distribution run
بتأكل بس فيه مشكلات بسيطة
فتعرفوا قدام 😊

qemu-system-arm -M versatilepb -Kernel
build/arch/arm/boot/zImage -dtb build/arch/
arm/boot/dts/arm/versatile-pb.dtb -Serial
stdio -append "serial=ttymAO"

* الـ ARM عند الحاجة مش كـ dtb
x86 وهى الـ device tree وري حاجت
بتقولك الـ peripherals الـ موجودة

* المشكلات عندك هنا انك في حال Panic
لا Kernel لاني عملت في system بقى
بيد الـ system روس بقى Ctrl+Alt+C
عشان توقف qemu او اقفها من فوق
كمان محتاجين صيـ utilities ده ودا
والكلام الجميل ده فمترجـ تنزيل
busybox ده فيه كل الـ utilities الـ
مشغورة

انا زهـ قمت ارفعها خذ الـ Commands
من الـ readme احسن واقولك بتعمل ايه
وليه

47 ← عمل Configuration لـ busy box ال default configurations

48 ← راج فتح ال menuConfig و عشان كايك يركب حاجات فيها

49 ← هتفتح ال settings وتعمل ال build بتاع ال busybox static عشان كل حاجت تكون موجودة

50 ← build

51 ← install و هتلاقيك داس Folder اسم install لوسميت tree جواه هتلاقي كل ال utilities المستخدمة تقريباً cat و cd بس

* بس في حاجت غريبة هتلاقي ان كل ال processes بي حرقاً بتشاور على ال busybox ، خالي بيحصل ان انت بتستخدم او بت call ال process وليكن cd مثلاً و هو بيروح ي call ال busybox وهو جواه ويتحقق بـ if مثلاً و مش if بس مشيها و ينفذ ال code بتاع ال cd

53 ← بلو قى احنا محتاجين نعمل ال busybox و ال init process
 ال جوا ال Kernel بتاعتنا، ففترج ت Copy
 ال busybox وتخطه جوا ال linux kernel جوا
 ال rootfs الى فيه ال Kernel الى عملها build

55 ← بعد كده محتاجين اننا نعمل ال init process بتاعتنا، وده
 فيحصل عن طريق اننا هيفتح File نسميها init ونعده
 فيه جيت Commands ال sh هتنفذهم

68, 69 ← خلصت هنتقل ال Files بس بتنظيم معين انا
 مش فاهمه بهراجه 😊

72, 73 ← هنعمله Compress لان ال Kernel لازم تبقى
 Compressed

76 ← هذ run ال distribution بتاعتنا

Note:-

الستهندس عمل File جديد منظم اكتر وارقام
 النسطور اتغيرت، ودي غايوة مني بهراجه بس
 كملت نفسي مش واخذ بالي وكملت