

Trabajo Práctico Nro 1 IMD

Se proponía realizar un módulo sencillo que imprima al iniciar “Hola Mundo. Estás utilizando Linux <version>” en donde la versión debe ser la del sistema en el que corre el módulo.

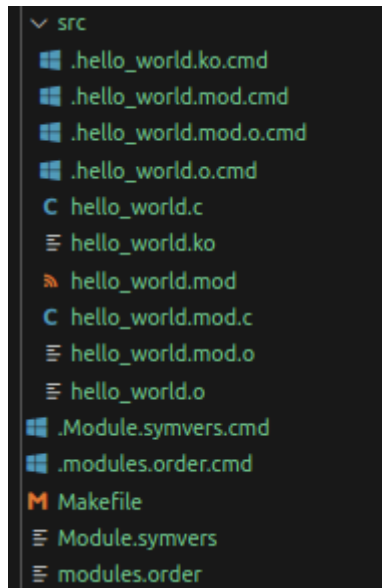
Para esto se escribió el siguiente código fuente:

```
1  #include <linux/init.h>
2  #include <linux/module.h>
3  #include <linux/kernel.h>
4  #include <linux/utsname.h>
5
6  static int __init hello_init(void) {
7      printk(KERN_INFO "Hola Mundo. Estás utilizando Linux %s\n", init_uts_ns.name.release);
8      return 0;
9  }
10
11 static void __exit hello_exit(void) {
12     printk(KERN_INFO "Adiós Mundo.\n");
13 }
14
15 module_init(hello_init);
16 module_exit(hello_exit);
17
18 MODULE_LICENSE("GPL");
19 MODULE_AUTHOR("Marcos Dominguez");
20 MODULE_DESCRIPTION("Trabajo Practico de MSE 1");
```

para compilarlo out-of-tree se utilizó el siguiente makefile:

```
Makefile u x C hello_world.c
Practicas > Practica1 > Makefile
1  obj-m += src/hello_world.o
2
3  KERNEL_DIR ?= $(HOME)/dev_rpi3_64_glibc_labs/kernel/linux-stable
4
5  all:
6      make -C $(KERNEL_DIR) ARCH=arm64 CROSS_COMPILE=aarch64-rpi3-linux-gnu- M=$(PWD) modules
7
8  clean:
9      make -C $(KERNEL_DIR) ARCH=arm64 CROSS_COMPILE=aarch64-rpi3-linux-gnu- M=$(PWD) clean
10
11 deploy:
12     cd src \
13     cp *.ko $(HOME)/dev_rpi3_64_glibc_labs/rootfs/hello_world \
14     cd ..
15
16
17
```

y al ejecutar make se obtienen los siguientes archivos de salida:



se copia el "hello_world.ko" a la sd donde esta cargado el rootfs que utiliza el kernel y se procede a cargarlo en la SBC (Raspberry 3 b+). El resultado obtenido:

```
picocom -b 115200 /dev/ttyUSB0
marcos@marcos-Lenovo-Flex-2-14:~/dev_rpi3_64_glibc_labs/kernel/linux-stable x picocom -b 115200 /dev/ttyUSB0
[ 2.317112] usb 1-1.1: new high-speed USB device number 3 using dwc2
[ 2.422311] hub 1-1.1:1.0: USB hub found
[ 2.426527] hub 1-1.1:1.0: 3 ports detected
[ 3.125108] usb 1-1.1.1: new high-speed USB device number 4 using dwc2
[ 3.496590] lan78xx 1-1.1.1.0 (unnamed net_device) (uninitialized): No External EEPROM. Setting MAC Speed
[ 4.333086] random: crng init done
- # ls
bin      etc      linuxrc  proc     sys
dev      hello_world  lost+found  sbin     usr
- # ls
bin      etc      linuxrc  proc     sys
dev      hello_world  lost+found  sbin     usr
- # insmod hello_world/hello_world.
hello_world.c  hello_world.ko
- # insmod hello_world/hello_world.
hello_world.c  hello_world.ko
- # insmod hello_world/hello_world.
hello_world.c  hello_world.ko
- # insmod hello_world/hello_world.ko
[ 191.267021] hello_world: loading out-of-tree module taints kernel.
[ 191.274092] Hola Mundo. Estás utilizando Linux 6.1.61+
- #
```

y al remover el módulo:

```
picocom -b 115200 /dev/ttyUSB0
marcos@marcos-Lenovo-Flex-2-14:~/dev_rpi3_64_glibc_labs/kernel/linux-stable x picocom -b 115200 /dev/ttyUSB0
[ 2.426527] hub 1-1.1:1.0: 3 ports detected
[ 3.125108] usb 1-1.1.1: new high-speed USB device number 4 using dwc2
[ 3.496590] lan78xx 1-1.1.1.0 (unnamed net_device) (uninitialized): No External EEPROM. Setting MAC Speed
[ 4.333086] random: crng init done
- # ls
bin      etc      linuxrc  proc     sys
dev      hello_world  lost+found  sbin     usr
- # ls
bin      etc      linuxrc  proc     sys
dev      hello_world  lost+found  sbin     usr
- # insmod hello_world/hello_world.
hello_world.c  hello_world.ko
- # insmod hello_world/hello_world.
hello_world.c  hello_world.ko
- # insmod hello_world/hello_world.
hello_world.c  hello_world.ko
- # insmod hello_world/hello_world.ko
[ 191.267021] hello_world: loading out-of-tree module taints kernel.
[ 191.274092] Hola Mundo. Estás utilizando Linux 6.1.61+
- # rmmod hello_world.ko
[ 333.613357] Adiós Mundo.
- #
```

Para compilar el module in-build se copió el código fuente en "driver/misc/"

```
03:55:45 *100 marcos ~/dev/rpi3_64_glibc_labs/kernel/linux-stable/drivers/misc g h hello * ?
ls
ad525x_dpot.c      built-in.a        eeprom            hpilo.c           lattice-ecp3-config.c  pci_endpoint_test.c  tfm_core.c
ad525x_dpot.h      c2port            enclosure.c        hpilo.h           lis3lv02d            phantom.c             ti-st
ad525x_dpot-i2c.c  cardreader        fastrpc.c          ibmasm            lkdtm                 pvpanic               tsl2550.c
ad525x_dpot-spi.c  cb710             gehc-achc.c        ibmvmc.c          Makefile              qcom-coincell.c      uacce
altera-stapl       cs5535-mfgpt.c    genwqe             ibmvmc.h          mchp_pci1xxx         sgi-gru               vcpu_stall_detector.c
apds9802als.c      cxl               habanalabs          ics932s401.c      mei                   sgi-xp                vmw_balloon.c
apds990x.c          ds1682.c           hello_world.c       isl29003.c         modules.order          sram.c                vmw_vmci
atmel-ssc.c         dummy-irq.c        hi6421v600-irq.c   isl29020.c         open-dice.c            sram-exec.c           xilinx_sdfec.c
bcm-vk              dw-xdata-pcie.c   hisi_hikey_usb.c   Kconfig            pch_phub.c             sram.h                tfm_7xx1.c
bh1770nls.c         echo              hmc6352.c           kqubits.c          nch_phub.c             tfm_7xx1.c
```

se modificó el Kconfig y Makefile como se muestra a continuación:

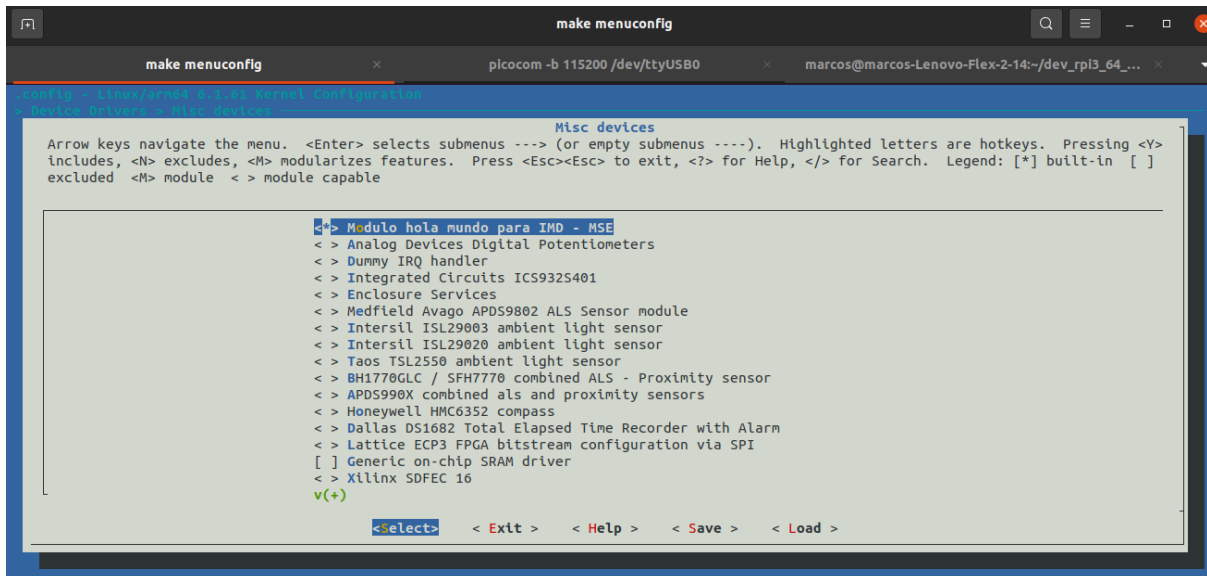
```
vi Kconfig
# SPDX-License-Identifier: GPL-2.0-only
#
# Misc strange devices
#
menu "Misc devices"

config MSE_HELLOWORLD
    tristate "Modulo hola mundo para IMD - MSE"
    default m
    help
        Utilice la opcion Y para compilar de manera built-in. Por defecto se compila como modulo in-tree
config SENSORS_LIS3LV02D
    tristate
    depends on INPUT

config AD525X_DPOT
    tristate "Analog Devices Digital Potentiometers"
    depends on (I2C || SPI) && SYSFS
    help
        If you say yes here, you get support for the Analog Devices
        AD5258, AD5259, AD5251, AD5252, AD5253, AD5254, AD5255
        AD5160, AD5161, AD5162, AD5165, AD5200, AD5201, AD5203,
        AD5204, AD5206, AD5207, AD5231, AD5232, AD5233, AD5235,
        AD5260, AD5262, AD5263, AD5290, AD5291, AD5292, AD5293,
        AD7376, AD8400, AD8402, AD8403, ADN2850, AD5241, AD5242,
        AD5243, AD5245, AD5246, AD5247, AD5248, AD5280, AD5282,
        ADN2860, AD5273, AD5171, AD5170, AD5172, AD5173, AD5270,
        AD5271, AD5272, AD5274
```

```
vi Makefile
obj-y += cb710/
obj-$(CONFIG_VMWARE_BALLOON) += vmw_balloon.o
obj-$(CONFIG_PCH_PHUB) += pch_phub.o
obj-y += ti-st/
obj-y += lis3lv02d/
obj-$(CONFIG_ALTERA_STAPL) += altera-stapl/
obj-$(CONFIG_INTEL_MEI) += mei/
obj-$(CONFIG_VMWARE_VMCI) += vmw_vmci/
obj-$(CONFIG_LATTICE_ECP3_CONFIG) += lattice-ecp3-config.o
obj-$(CONFIG_SRAM) += sram.o
obj-$(CONFIG_SRAM_EXEC) += sram-exec.o
obj-$(CONFIG_GENWQE) += genwqe/
obj-$(CONFIG_ECHO) += echo/
obj-$(CONFIG_CXL_BASE) += cxl/
obj-$(CONFIG_DW_XDATA_PCIE) += dw-xdata-pcie.o
obj-$(CONFIG_PCI_ENDPOINT_TEST) += pci_endpoint_test.o
obj-$(CONFIG_OCXL) += ocxl/
obj-$(CONFIG_BCM_VK) += bcm-vk/
obj-y += cardreader/
obj-$(CONFIG_PVPANIC) += pvpanic/
obj-$(CONFIG_HABANA_AI) += habanalabs/
obj-$(CONFIG_UACCE) += uacce/
obj-$(CONFIG_XILINX_SDFEC) += xilinx_sdfec.o
obj-$(CONFIG_HISI_HIKEY_USB) += hisi_hikey_usb.o
obj-$(CONFIG_HI6421V600_IRQ) += hi6421v600-irq.o
obj-$(CONFIG_OPEN_DICE) += open-dice.o
obj-$(CONFIG_GP_PCI1XXX) += mchp_pci1xxx/
obj-$(CONFIG_VCPU_STALL_DETECTOR) += vcpu_stall_detector.o
obj-$(CONFIG_MSE_HELLOWORLD) += hello_world.o
```

luego utilizando make menuconfig se selecciona el módulo para ser compilado built-in:



y finalmente se compila con el comando "make modules"

