Make your own USB gadget

Kernel and userspace

Andrzej Pietrasiewicz

Samsung R&D Institute Poland
Warsaw, Poland
andrzej.p@samsung.com

June 22-25, 2017

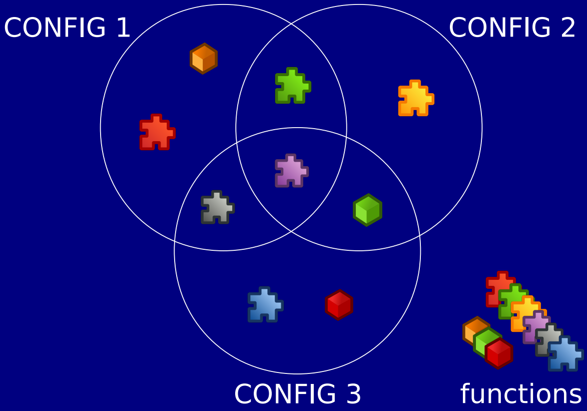
Table of Contents

1. USB gadget
   - USB and functions
   - USB gadget
   - Gadget implementation in Linux
2. Configfs
   - The idea, example
   - On implementation
3. Userspace
   - libusbgx & tools
4. Q & A

host += function

- USB: host, device
- extend the host with some function(s)

USB device composition

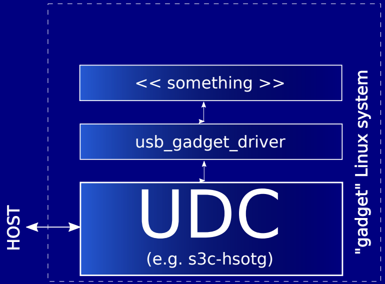


enumeration

- Device connected, presents itself
- Host decides what to do and how to talk to it

gadget = UDC + function(s)

- A piece in hardware: UDC/OTG/
- Functions: HW or SW

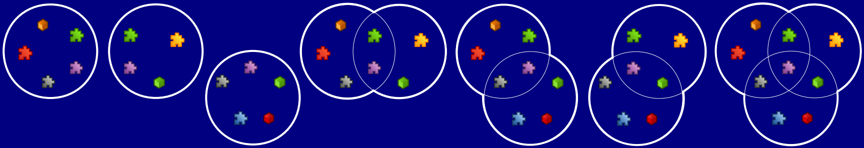


composite framework

- factor out repeated parts of code
  - drivers/usb/gadget/composite.c
- reusable functions' implementations
  - f_acm.c
  - f_serial.c
  - f_obex.c
  - f_ecm.c, f_ecm_subset.c, f_eem.c, f_ncm.c, f_rndis.c
  - f_phonet.c
  - f_mass_storage.c
  - f_uvc.c
  - f_uac1.c, f_uac2.c
  - f_midi.c
  - . . .

gadgets proper: g_xyz.c / g_xyz kernel modules

- hardcoded (!) configurations/functions/identity
- module parameters



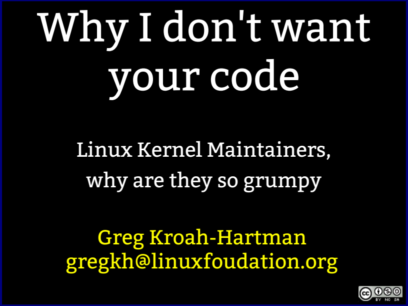
Greg

Fact

He doesn't want my code!

Fact

He wouldn't want your code, either :O



Why I don't want your code

Linux Kernel Maintainers, why are they so grumpy

Greg Kroah-Hartman
gregkh@linuxfoudation.org

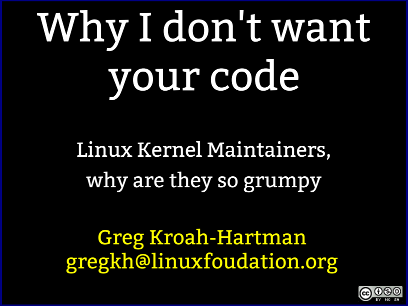
Greg

Fact

He doesn’t want my code!

Fact

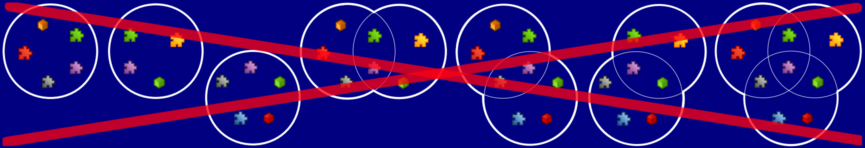
He wouldn’t want your code, either :O



Why I don't want your code

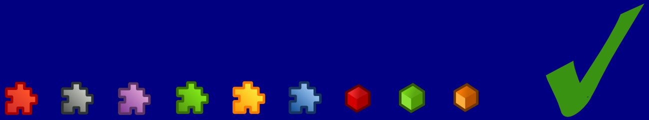
Linux Kernel Maintainers,
why are they so grumpy

Greg Kroah-Hartman
gregkh@linuxfoudation.org



Separate code from data

- decouple the information on actual gadget composition from implementation
- only provide building blocks (mechanism, not policy)



Let the user decide at runtime

action	filesystem
create	make directory
destroy	remove directory
specify value	write
get value	read (execute for directory)
group things	symlink
ungroup things	remove symlink

Command reference

```
mkdir, rmdir
echo 'something' > file, cat file, ls directory
ln -s, rm
```

Example

Example's prologue

```
$ modprobe libcomposite
```

Example

Example's prologue

```
$ modprobe libcomposite
$ mount none cfg -t configfs
```

`cfg/usb_gadget`

Example

```
drwxr-xr-x    .
drwxr-xr-x    ./strings
drwxr-xr-x    ./configs
drwxr-xr-x    ./functions
-rw-r--r--    ./UDC
-rw-r--r--    ./bcdUSB
-rw-r--r--    ./bcdDevice
-rw-r--r--    ./idProduct
-rw-r--r--    ./idVendor
-rw-r--r--    ./bMaxPacketSize0
-rw-r--r--    ./bDeviceProtocol
-rw-r--r--    ./bDeviceSubClass
-rw-r--r--    ./bDeviceClass
```

Example's prologue

```
$ modprobe libcomposite
$ mount none cfg -t configfs
$ mkdir cfg/usb_gadget/g1
$ cd cfg/usb_gadget/g1
```

Example

Example's prologue

```
$ modprobe libcomposite
$ mount none cfg -t configfs
$ mkdir cfg/usb_gadget/g1
$ cd cfg/usb_gadget/g1
$ echo "0x05e8" > idVendor
$ echo "0xa4a1" > idProduct
```

```
drwxr-xr-x    .
drwxr-xr-x    ./strings
drwxr-xr-x    ./configs
drwxr-xr-x    ./functions
-rw-r--r--    ./UDC
-rw-r--r--    ./bcdUSB
-rw-r--r--    ./bcdDevice
-rw-r--r--    ./idProduct
-rw-r--r--    ./idVendor
-rw-r--r--    ./bMaxPacketSize0
-rw-r--r--    ./bDeviceProtocol
-rw-r--r--    ./bDeviceSubClass
-rw-r--r--    ./bDeviceClass
```

Example

Example's prologue

```
$ modprobe libcomposite
$ mount none cfg -t configfs
$ mkdir cfg/usb_gadget/g1
$ cd cfg/usb_gadget/g1
$ echo "0x05e8" > idVendor
$ echo "0xa4a1" > idProduct
$ mkdir strings/0x409
$ echo "serialnumber" > strings/0x409/serialnumber
$ echo "manufacturer" > strings/0x409/manufacturer
$ echo "RNDIS Gadget" > strings/0x409/product
```

```
drwxr-xr-x   .
drwxr-xr-x   ./strings
drwxr-xr-x   ./configs
drwxr-xr-x   ./functions
-rw-r--r--   ./UDC
-rw-r--r--   ./bcdUSB
-rw-r--r--   ./bcdDevice
-rw-r--r--   ./idProduct
-rw-r--r--   ./idVendor
-rw-r--r--   ./bMaxPacketSize0
-rw-r--r--   ./bDeviceProtocol
-rw-r--r--   ./bDeviceSubClass
-rw-r--r--   ./bDeviceClass
```

One config, one function

Example

```
$ mkdir functions/rndis.usb0
```

One config, one function

Example

```
$ mkdir functions/rndis.usb0
$ mkdir configs/c.1
$ mkdir configs/c.1/strings/0x409
$ echo Conf 1 > configs/c.1/strings/0x409/configuration
$ echo 120 > configs/c.1/MaxPower
```

One config, one function

Example

```
$ mkdir functions/rndis.usb0
$ mkdir configs/c.1
$ mkdir configs/c.1/strings/0x409
$ echo Conf 1 > configs/c.1/strings/0x409/configuration
$ echo 120 > configs/c.1/MaxPower
$ ln -s functions/rndis.usb0 configs/c.1
```

One config, one function

Example

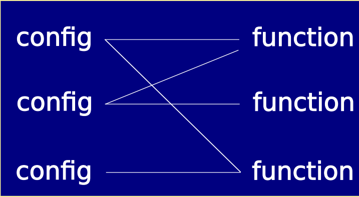
```
$ mkdir functions/rndis.usb0
$ mkdir configs/c.1
$ mkdir configs/c.1/strings/0x409
$ echo Conf 1 > configs/c.1/strings/0x409/configuration
$ echo 120 > configs/c.1/MaxPower
$ ln -s functions/rndis.usb0 configs/c.1
$ echo 12480000.hsotg > UDC
```

```
$ ls /sys/class/udc
12480000.hsotg
# formerly s3c-hsotg
```

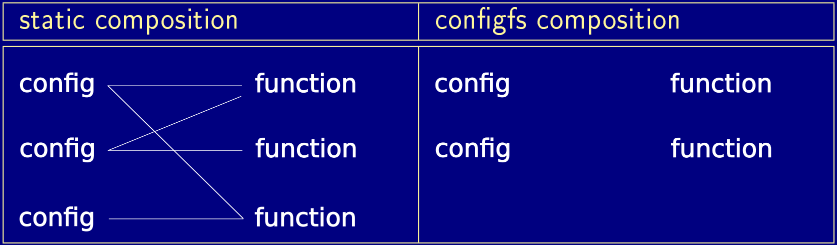
- bind!

Static composition vs composition with configfs

Traditional g_xyz.ko modules directly #included f_*.c!

static composition	configfs composition
config config config function function function	

Static composition vs composition with configfs



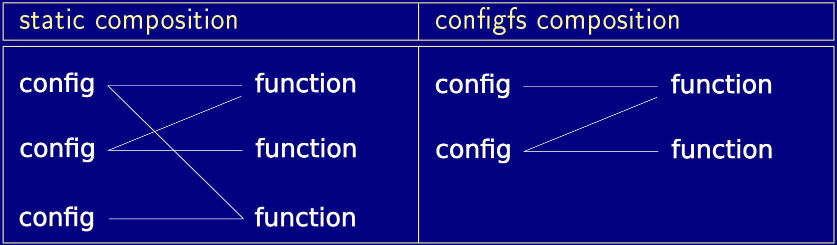
static composition

config — function
config — function
config — function

configfs composition

config — function
config — function

Static composition vs composition with configfs



static composition

config config config

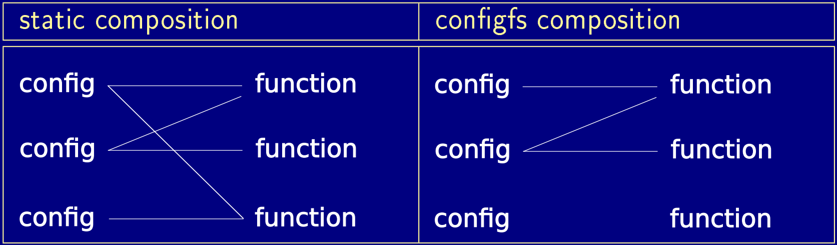
function function function

configfs composition

config config

function function

Static composition vs composition with configfs



static composition

config
config
config

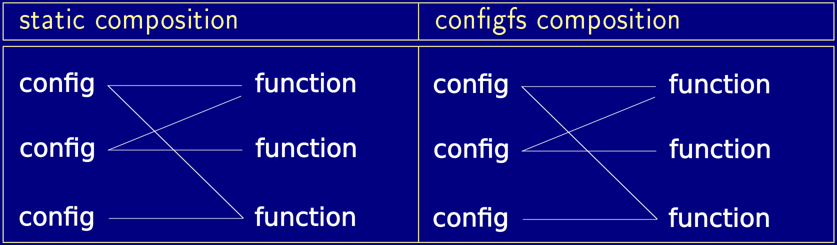
function
function
function

configfs composition

config
config
config

function
function
function

Static composition vs composition with configfs



static composition

config
config
config

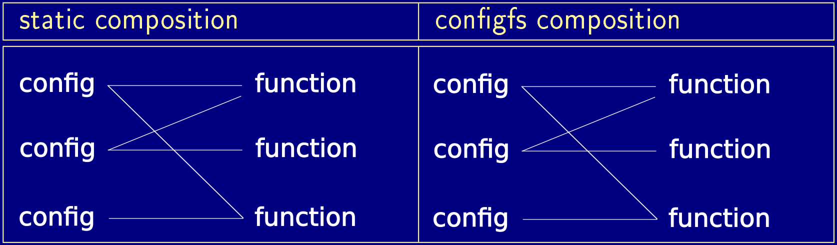
function
function
function

configfs composition

config
config
config

function
function
function

Static composition vs composition with configfs



static composition

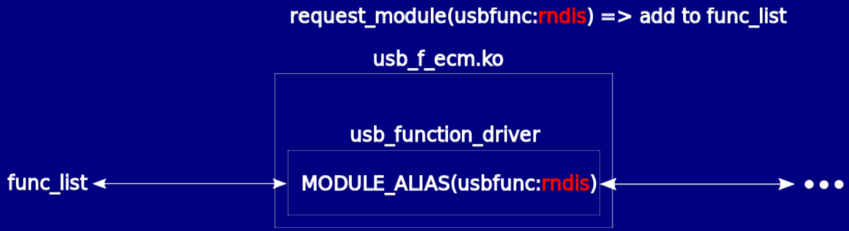
config — function
config — function
config — function

configfs composition

config — function
config — function
config — function

No g_xyz.ko modules!

Function registration framework



request_module(usbfunc:rndis) => add to func_list

usb_f_ecm.ko

usb_function_driver

MODULE_ALIAS(usbfunc:rndis)

func_list

• • •

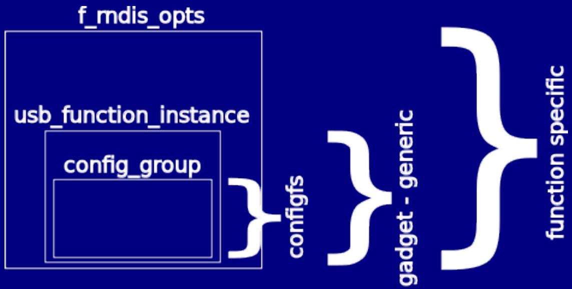
```
$ mkdir function/rndis.usb0

                 request_module()
```

- Sebastian Andrzej Siewior

usb_function_instance

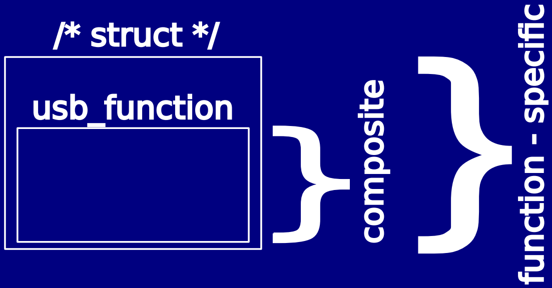
```
$ mkdir function/rndis.usb0
```



- user-accessible configuration data of this specific instance
- legacy gadgets ≈ hardcoded

usb_function

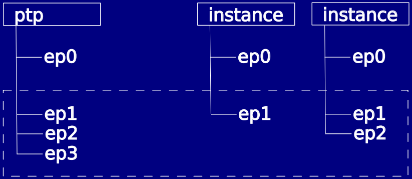
```
$ ln -s function/rndis.usb0 configs/c.1
```



- the same as in legacy gadgets
- composing a gadget (legacy/configfs) ends up providing this
- see: drivers/usb/gadget

FunctionFS vs configfs

- delegate function implementation to userspace
    - mount FunctionFS
    - write descriptors to ep0
    - read/write/poll ep[1-]



- configfs: only create FunctionFS instances
    - eg $CONFIGFS_ROOT/usb_gadget/gadget/functions/ffs.ptp

userspace

- With configfs ease of use is not concerned
- Composing is not difficult, but tedious (at the very least ≈20 shell commands)
- I want my "modprobe g_ether"!!!
  - shell script
  - dedicated userspace program

userspace

- With configfs ease of use is not concerned
- Composing is not difficult, but tedious (at the very least ≈20 shell commands)
- I want my "modprobe g_ether"!!!
    - shell script
    - dedicated userspace program

```
$ modprobe libcomposite
$ mount none cfg -t configfs
$ mkdir cfg/usb_gadget/g1
$ cd cfg/usb_gadget/g1
$ echo 0x04e8 > idVendor
$ echo 0xa4a1 > idProduct
$ echo Foo > strings/0x409/manufacturer
$ echo Bar > strings/0x409/product
$ echo 123 > strings/0x409/serialnumber
$ mkdir configs/c.1
$ mkdir strings/0x409
$ mkdir configs/c.1/strings/0x409
$ echo "Conf 1" > configs/c.1/strings/0x409/configuration
$ mkdir functions/ecm.usb0
$ ln -s functions/ecm.usb0 configs/c.1
$ echo 12480000.hsotg > UDC
```

libusbgx

- C API for all (wrap filesystem operations)
- https://github.com/libusbgx/libusbgx, maintainer: Krzysztof Opasiak
- Gadget schemes
  - gadget (composition) export to file
  - gadget (composition) import from file

libusbgx

- C API for all (wrap filesystem operations)
- https://github.com/libusbgx/libusbgx, maintainer: Krzysztof Opasiak
- Gadget schemes
    - gadget (composition) export to file
    - gadget (composition) import from file

```
attrs = { idVendor = 0x04e8; idProduct = 0xa4a1; }
strings = (
    { lang = 0x409; manufacturer = "Foo"; product = "Bar"; serialnumber = "123"; }
)
configs = (
    {
        name = "c"
        id = 1
        strings = (
            { lang = 0x409; configuration = "Conf 1"; }
        )
        functions = (
            { function = { type = "ecm"; instance = "usb0"; } }
        )
    }
)
```

Q & A

Andrzej Pietrasiewicz
andrzej.p@samsung.com

References

- http://www.spinics.net/lists/linux-usb/msg74991.html
- http://www.spinics.net/lists/linux-usb/msg76378.html
- http://www.spinics.net/lists/linux-usb/msg83460.html
- http://www.spinics.net/lists/linux-usb/msg86311.html
- http://www.spinics.net/lists/linux-usb/msg86321.html
- http://www.spinics.net/lists/linux-usb/msg86327.html
- http://www.spinics.net/lists/linux-usb/msg86561.html
- http://www.spinics.net/lists/linux-usb/msg90757.html
- http://www.spinics.net/lists/linux-usb/msg90774.html
- http://www.spinics.net/lists/linux-usb/msg90776.html
- http://www.spinics.net/lists/linux-usb/msg97006.html
- http://www.spinics.net/lists/linux-usb/msg98731.html
- http://www.spinics.net/lists/linux-usb/msg110639.html
- http://www.spinics.net/lists/linux-usb/msg110718.html
- http://www.spinics.net/lists/linux-usb/msg110962.html
- http://www.spinics.net/lists/linux-usb/msg115328.html
- http://www.spinics.net/lists/linux-usb/msg115567.html
- http://www.spinics.net/lists/linux-usb/msg122201.html
- http://www.spinics.net/lists/linux-usb/msg133963.html
- https://github.com/libusbgx/libusbgx

Images

- http://openclipart.org/detail/174619/4g-modem-and-sim-by-witcombem-174619 - slide 3
- http://openclipart.org/detail/1964/calcubot-by-johnny_automatic - slide 3
- http://openclipart.org/detail/96913/mouse-by-yves_guillou - slide 3
- http://openclipart.org/detail/27549/keyboard-keys-by-simanek - slide 3
- http://openclipart.org/detail/17924/computer-by-aj - slide 3
- http://openclipart.org/detail/176486/pen-drive-by-carloernesto-176486 - slide 3
- http://openclipart.org/detail/6633/neo1973-%28tango%29-by-ryanlerch - slide 3
- http://openclipart.org/detail/17026/icon_puzzle_blue-by-jean_victor_balin - slides 4, 8, 9, 10
- http://openclipart.org/detail/17027/icon_puzzle_green-by-jean_victor_balin - slides 4, 8, 9, 10
- http://openclipart.org/detail/17028/icon_puzzle_grey-by-jean_victor_balin - slides 4, 8, 9, 10
- http://openclipart.org/detail/17029/icon_puzzle_purple-by-jean_victor_balin - slides 4, 8, 9, 10
- http://openclipart.org/detail/17030/icon_puzzle_red-by-jean_victor_balin - slides 4, 8, 9, 10
- http://openclipart.org/detail/17031/icon_puzzle_yellow-by-jean_victor_balin - slides 4, 8, 9, 10
- http://openclipart.org/detail/17060/icon_cube_green-by-jean_victor_balin - slides 4, 8, 9, 10
- http://openclipart.org/detail/17061/icon_cube_orange-by-jean_victor_balin - slides 4, 8, 9, 10
- http://openclipart.org/detail/17062/icon_cube_red-by-jean_victor_balin - slides 4, 8, 9, 10
- http://openclipart.org/detail/122449/question-button-by-ricardomaia - slide 5
- http://openclipart.org/detail/3705/usb-plug-by-klaasvangend - slide 5

Images

- http://www.linaro.org/documents/download/304a9a3e4024a2bb70312fc81d79446d51311e50ed8f4 - slide 9
- http://openclipart.org/detail/10833/green-tick-by-ryan_taylor - slide 11
- http://openclipart.org/detail/12929/large-braces-by-anonymous-12929 - slide 16, 17