request_firmware API

You would typically load firmware and then load it into your device somehow. The typical firmware work flow is reflected below:

```
if(request_firmware(&fw_entry, $FIRMWARE, device) == 0)
       copy_fw_to_device(fw_entry->data, fw_entry->size);
release_firmware(fw_entry);
```

Synchronous firmware requests

Synchronous firmware requests will wait until the firmware is found or until an error is returned.

request_firmware

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\firmware\(linux-master)(Documentation)(driver-api)(firmware)request_firmware.rst`, line 20)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/base/firmware_loader/main.c
   :functions: request_firmware
```

firmware_request_nowarn

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\firmware\(linux-master)(Documentation)(driver-api)(firmware)request_firmware.rst`, line 25)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/base/firmware_loader/main.c
   :functions: firmware_request_nowarn
```

firmware_request_platform

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\firmware\(linux-master)(Documentation)(driver-api)(firmware)request_firmware.rst`, line 30)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/base/firmware_loader/main.c
   :functions: firmware_request_platform
```

request_firmware_direct

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\firmware\(linux-master)(Documentation)(driver-api)(firmware)request_firmware.rst`, line 35)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/base/firmware_loader/main.c
   :functions: request_firmware_direct
```

request_firmware_into_buf

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\firmware\(linux-master)(Documentation)(driver-api)(firmware)request_firmware.rst`, line 40)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/base/firmware_loader/main.c
   :functions: request_firmware_into_buf
```

Asynchronous firmware requests

Asynchronous firmware requests allow driver code to not have to wait until the firmware or an error is returned. Function callbacks are provided so that when the firmware or an error is found the driver is informed through the callback. request_firmware_nowait() cannot be called in atomic contexts.

request_firmware_nowait

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\firmware\(linux-master) (Documentation) (driver-api) (firmware) request_firmware.rst`, line 54)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/base/firmware_loader/main.c
   :functions: request_firmware_nowait
```

Special optimizations on reboot

Some devices have an optimization in place to enable the firmware to be retained during system reboot. When such optimizations are used the driver author must ensure the firmware is still available on resume from suspend, this can be done with firmware_request_cache() instead of requesting for the firmware to be loaded.

firmware_request_cache()

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\firmware\(linux-master) (Documentation) (driver-api) (firmware) request_firmware.rst`, line 68)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/base/firmware_loader/main.c
   :functions: firmware_request_cache
```

request firmware API expected driver use

Once an API call returns you process the firmware and then release the firmware. For example if you used request_firmware() and it returns, the driver has the firmware image accessible in fw_entry->{data,size}. If something went wrong request_firmware() returns non-zero and fw_entry is set to NULL. Once your driver is done with processing the firmware it can call release_firmware(fw_entry) to release the firmware image and any related resource.