

# Linuxカーネル開発者予備軍 のためのGit Howto

Japan Technical Jamboree 2009/05/12

(株)ルネサス ソリューションズ  
システムビジネス本部 システム推進部 第三グループ  
岩松 信洋

# 自己紹介

- 岩松 信洋 (IWAMATSU Nobuhiro)
- Linux カーネル開発をしています(Renesas SH)
- U-Boot のSHアーキテクチャメンテナしています
- Debian の開発に参加しています
  - Debian Maintainer / Debian JP Project Leader

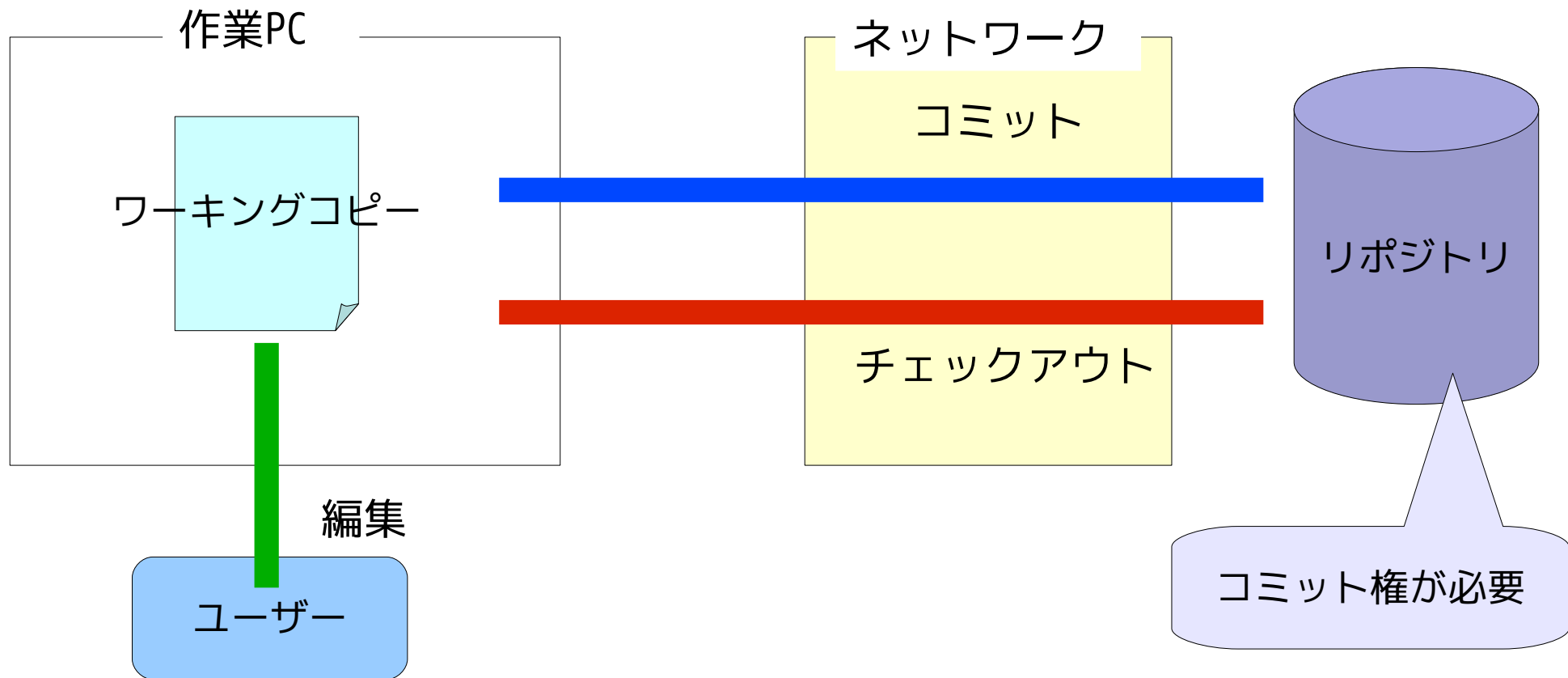
# 今回のお題

## Linuxカーネル開発者予軍 のためのGit Howto

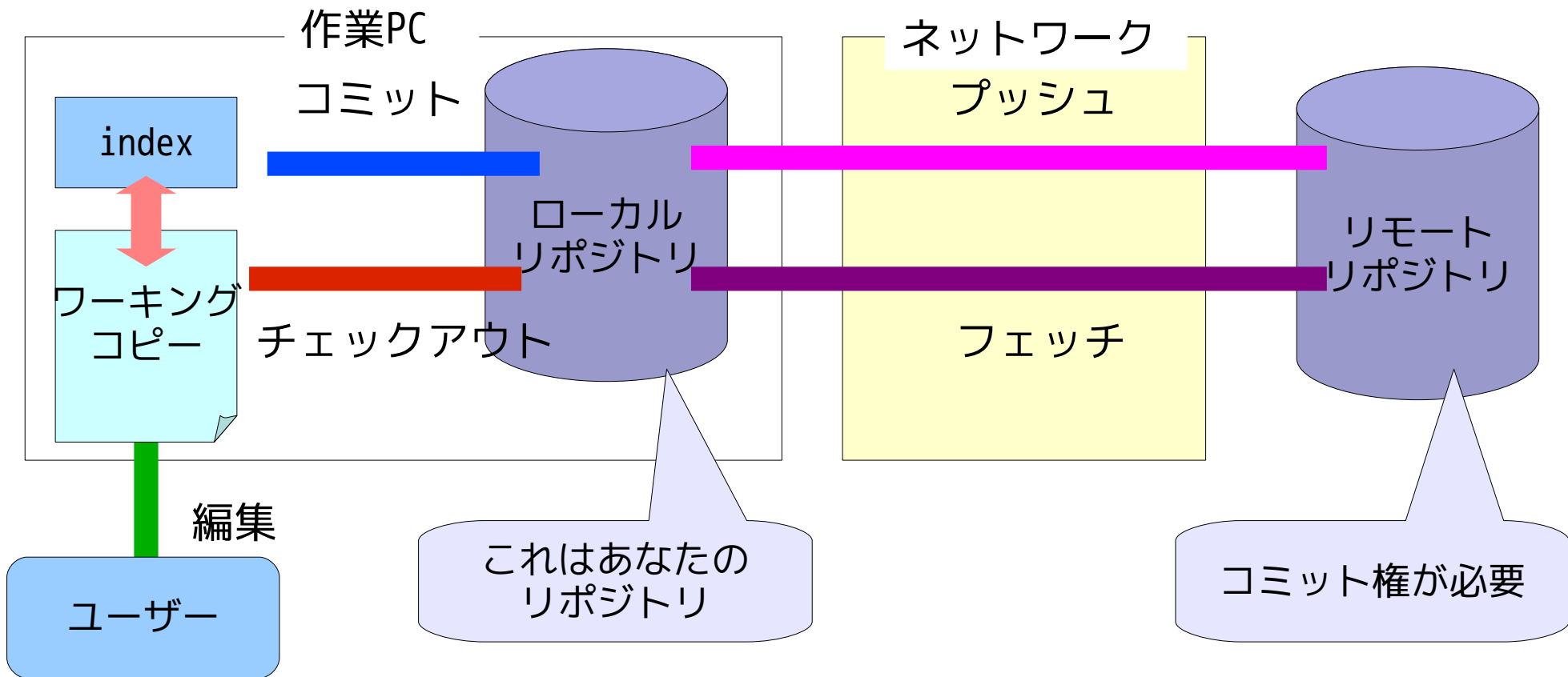
1. Gitの基本的なコマンド
2. Gitを使った開発の流れ
  1. パッチを送るまで
  2. 最新機能を試す
3. 質疑応答

# 1. Gitの基本的なコマンド

# Subversion



# Git



# Git の基本的なコマンド

- git clone (リポジトリのコピー)
- git add/rm/mv (ファイルの追加、削除、リネーム)
- git commit (ローカルリポジトリへコミット)
- git diff (差分の表示)
- git log (コミットログの確認)
- git format-patch (パッチの作成)
- git remote (リモートリポジトリの管理)
- git rebase/merge (ブランチ間のリベース、マージ)
- git pull (リモートリポジトリからコピー、マージ)
- git branch / git checkout (ブランチの操作)

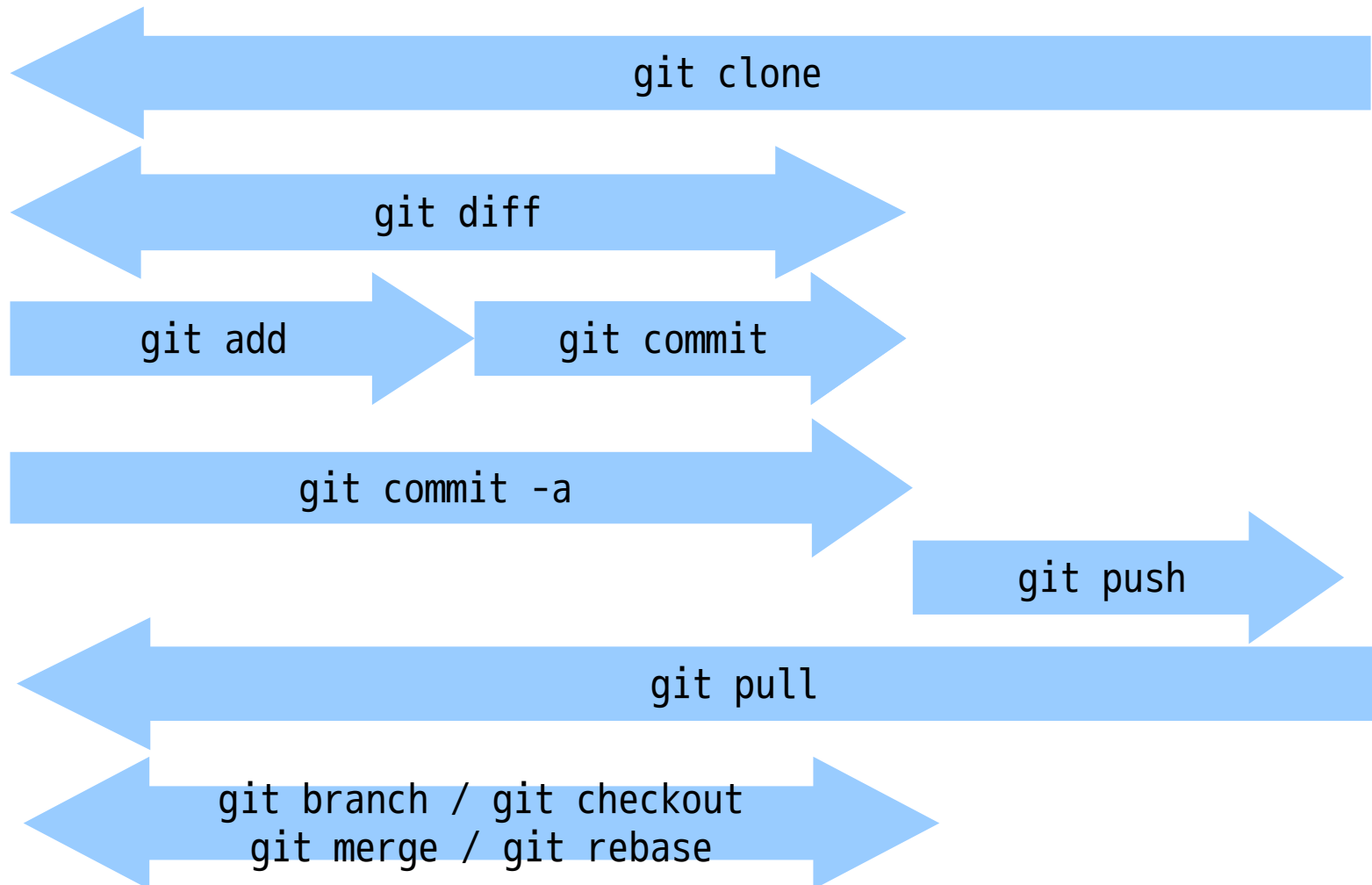
# Git の基本的なコマンド

ワーキングコピー

index

ローカルリポジトリ

リモートリポジトリ



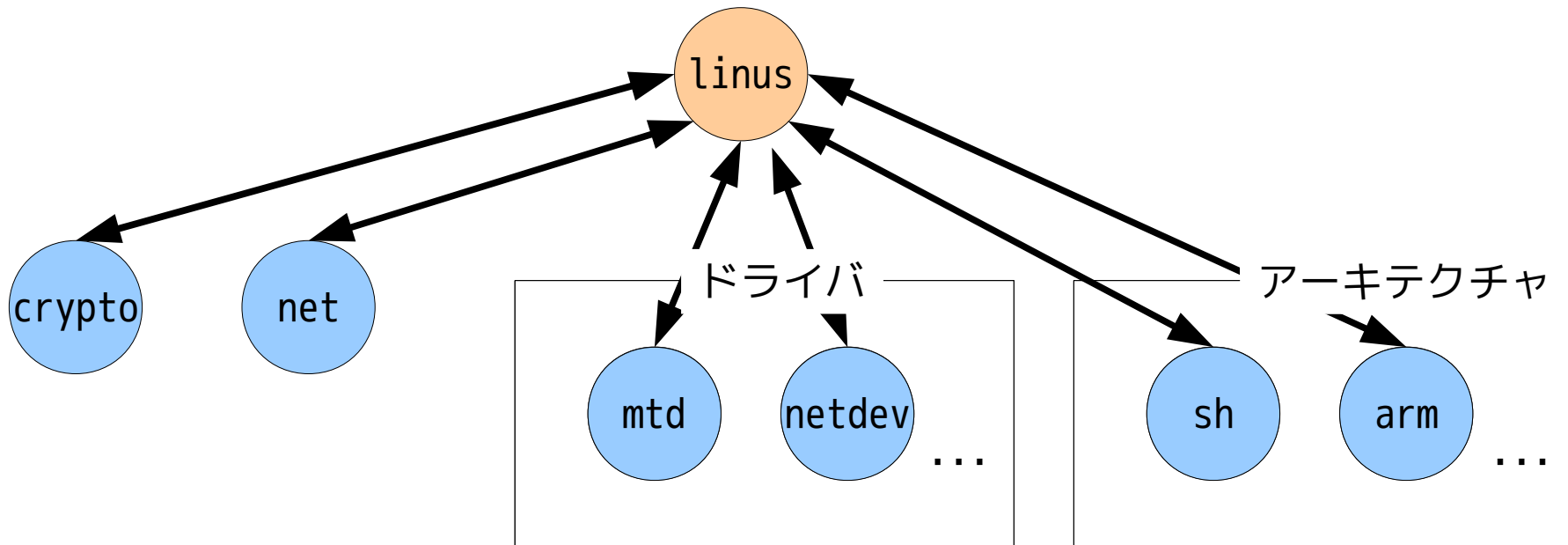


## 2. Gitを使った開発の流れ

## 2.1. パッチを送るまで

- 基本的な考え方
- パッチは開発者の最新のコミットに対して修正を送る
- カーネルでは、各機能毎にメンテナがいるので、メンテナがメンテナンスしているリポジトリに対して修正を行う
- パッチもメンテナおよび機能毎に用意されているメーリングリストに送る

## 2.1. パッチを送るまで



例) SHアーキテクチャに関する  
パッチを送る場合

## 0. 環境の設定

Linux カーネル開発/Git では、名前とメールアドレスを利用する

Gitで使う名前とメールアドレスを設定する

```
$ git config --global \  
    user.name "Nobuhiro Iwamatsu"  
$ git config --global user.email \  
    "iwamatsu.nobuhiro@renesas.com"
```

# 1. リポジトリをコピーする

SHアーキテクチャのGitリポジトリはMAINTAINERS  
ファイルに書いてある

SUPERH

P: Paul Mundt

M: lethal@linux-sh.org

L: linux-sh@vger.kernel.org

W: <http://www.linux-sh.org>

T: git

<git://git.kernel.org/pub/scm/linux/kernel/git/lethal/sh-2.6.git>

# 1. リポジトリをコピーする

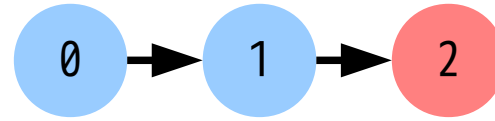
git clone コマンドでリポジトリをコピー

```
$ git clone \  
git://git.kernel.org/pub/scm/linux/kernel/gi  
t/lethal/sh-2.6.git
```

# 1. リポジトリをコピーする

sh-2.6リモート  
リポジトリ

master  
ブランチ

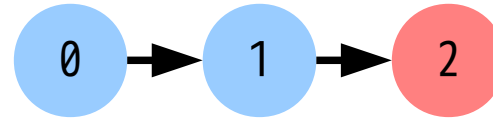




# 1. リポジトリをコピーする

sh-2.6リモート  
リポジトリ

master  
ブランチ

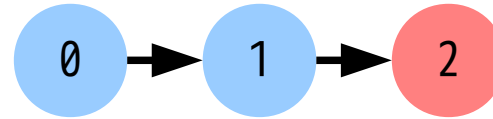


リポジトリをコピー

# 1. リポジトリをコピーする

sh-2.6リモート  
リポジトリ

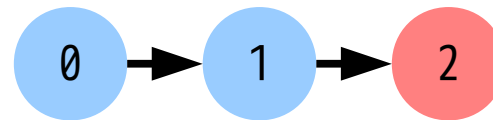
master  
ブランチ



リポジトリをコピー

ローカル  
リポジトリ

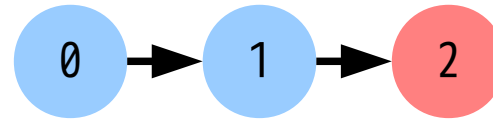
origin/master  
ブランチ



# 1. リポジトリをコピーする

sh-2.6リモート  
リポジトリ

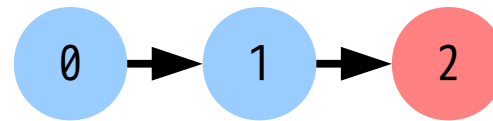
master  
ブランチ



リポジトリをコピー

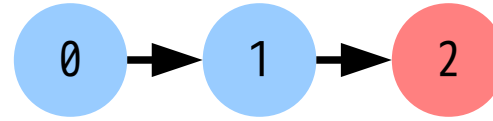
ローカル  
リポジトリ

origin/master  
ブランチ



リモートリポジトリのmasterブランチを  
ローカルブランチとして登録する

heads/master  
ブランチ



2. git clone を実行すると、sh-2.6 ディレクトリができていますので、cd する

```
$ ls  
sh-2.6  
$ cd sh-2.6
```

### 3. 不具合の修正をする

```
$ vi arch/sh/Kconfig
..... edit .....

$ git diff
diff --git a/arch/sh/Kconfig b/arch/sh/Kconfig
index 8d50d52..19566c8 100644
--- a/arch/sh/Kconfig
+++ b/arch/sh/Kconfig
@@ -530,7 +530,7 @@ source kernel/Kconfig.hz

config KEXEC
    bool "kexec system call (EXPERIMENTAL)"
-    depends on SUPERH32 && EXPERIMENTAL
+    depends on SUPERH32 && EXPERIMENTAL && MMU
    help
        kexec is a system call that implements the ability
.....
```

## 4. ドライバの修正ができれば、ソースのチェック、コンパイル、テストを行う

```
$ make ARCH=sh menuconfig  
$ make ARCH=sh CROSS_COMPILE=sh4-linux-gnu-\  
-j8  
$ 実機でテスト
```

## 4. 問題がなければ、変更をコミットする

```
$ git commit arch/sh/Kconfig -s
```

- -s は Signed-off をつけるという意味
- git commit実行すると、エディタが立ち上がる

## 4. 問題がなければ、変更をコミットする

```
sh: Add MMU dependency for kexec
```

```
Kexec is not likely to work on NON-MMU CPU  
because this added  
depend on MMU for kexec.
```

```
Signed-off-by: Nobuhiro Iwamatsu \  
<iwamatsu.nobuhiro@renesas.com>
```

- 1行目はメールのサブジェクトになる  
(後で説明)
- 空行を入れて、次の行からメールの本文になる



## 4. 問題がなければ、変更をコミットする

```
$ git log
commit f36b59d1a4fa1e29be606d0513b5f7fa6e720f79
Author: Nobuhiro Iwamatsu <iwamatsu.nobuhiro@renesas.com>
Date: Thu Mar 26 08:32:37 2009 +0000

sh: Add MMU dependency for kexec

Kexec is not likely to work on NON-MMU CPU because this added
depend on MMU for kexec.

Signed-off-by: Nobuhiro Iwamatsu \
                <iwamatsu.nobuhiro@renesas.com>

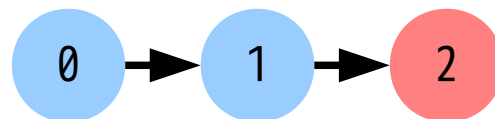
diff --git a/arch/sh/Kconfig b/arch/sh/Kconfig
index 8d50d52..19566c8 100644
--- a/arch/sh/Kconfig
+++ b/arch/sh/Kconfig
@@ -530,7 +530,7 @@ source kernel/Kconfig.hz

config KEXEC
```

# 4. 問題がなければ、変更をコミットする

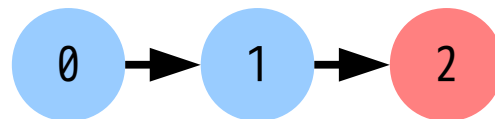
sh-2.6リモート  
リポジトリ

master  
ブランチ

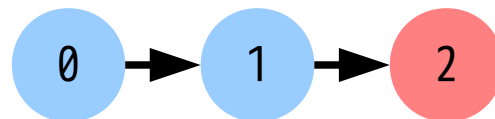


ローカル  
リポジトリ

origin/master  
ブランチ



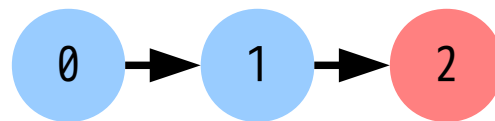
heads/master  
ブランチ



# 4. 問題がなければ、変更をコミットする

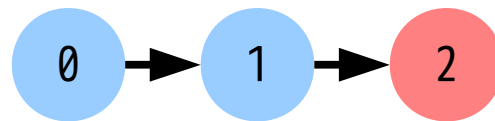
sh-2.6リモート  
リポジトリ

master  
ブランチ

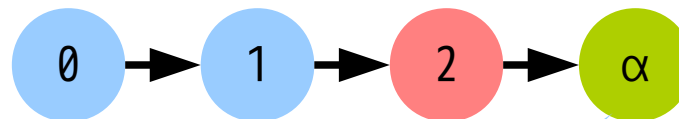


ローカル  
リポジトリ

origin/master  
ブランチ



heads/master  
ブランチ



heads/master  
にコミットされる

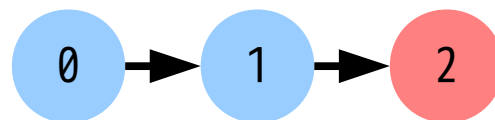
5. 修正している間にリモートリポジトリが更新されている可能性があるので、リモートリポジトリの状態をアップデートする

```
$ git remote update
```

# 5. 修正している間にリモートリポジトリが更新されている可能性があるので、リモートリポジトリの状態をアップデートする

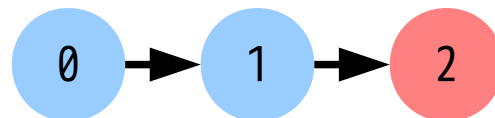
sh-2.6 リモート  
リポジトリ

master  
ブランチ

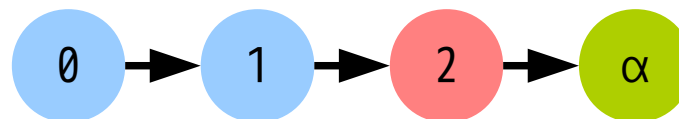


ローカル  
リポジトリ

origin/master  
ブランチ



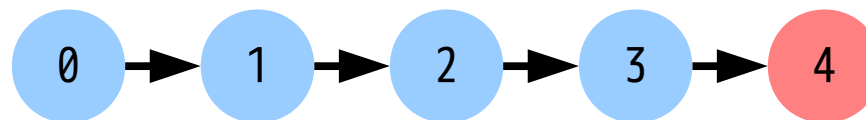
heads/master  
ブランチ



# 5. 修正している間にリモートリポジトリが更新されている可能性があるので、リモートリポジトリの状態をアップデートする

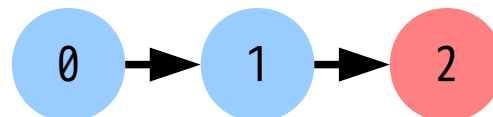
sh-2.6 リモート  
リポジトリ

master  
ブランチ

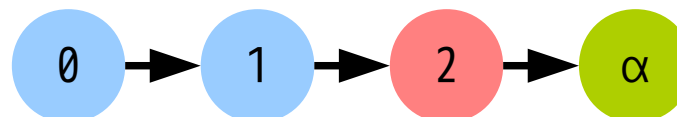


ローカル  
リポジトリ

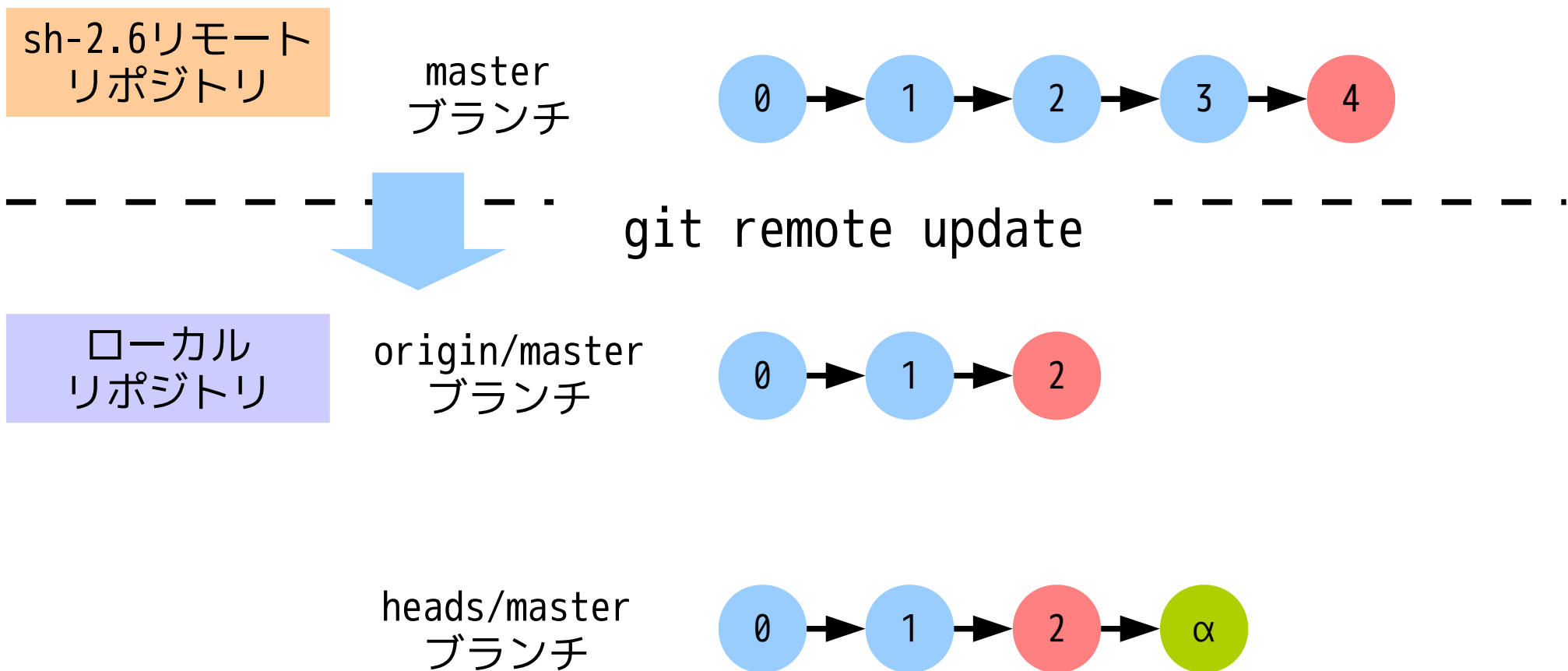
origin/master  
ブランチ



heads/master  
ブランチ



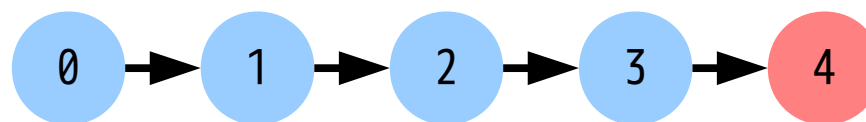
# 5. 修正している間にリモートリポジトリが更新されている可能性があるので、リモートリポジトリの状態をアップデートする



# 5. 修正している間にリモートリポジトリが更新されている可能性があるので、リモートリポジトリの状態をアップデートする

sh-2.6 リモート  
リポジトリ

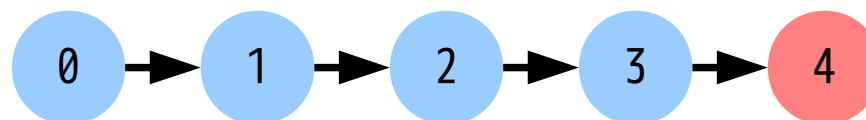
master  
ブランチ



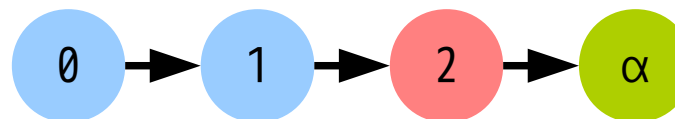
git remote update

ローカル  
リポジトリ

origin/master  
ブランチ



heads/master  
ブランチ





## 6. リモートリポジトリの状態に対してrebaseを行う

```
$ git rebase origin
```

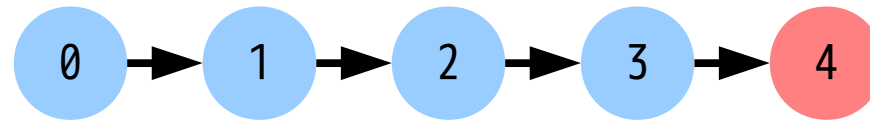
```
※git pull --rebase でも可能
```

- ソースコードのマージには2種類ある
- git rebase は指定したブランチをベースに自分の修正を追加するコマンド
- Git merge は指定したブランチの修正を自分のブランチに追加するコマンド

# 6. リモートリポジトリの状態に対してrebaseを行う

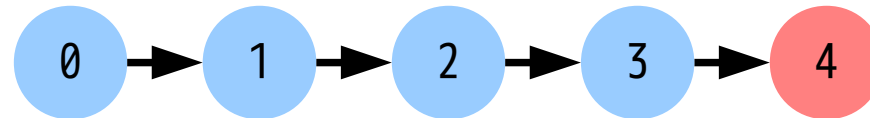
sh-2.6 リモート  
リポジトリ

master  
ブランチ

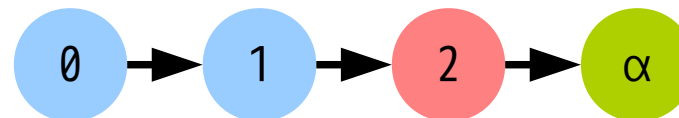


ローカル  
リポジトリ

origin/master  
ブランチ



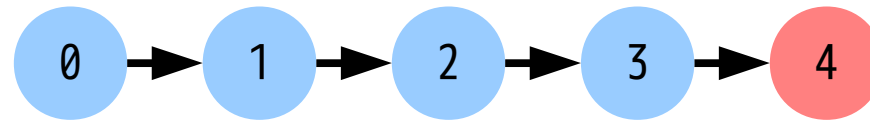
heads/master  
ブランチ



# 6. リモートリポジトリの状態に対してrebaseを行う

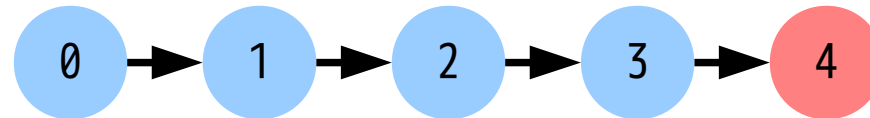
sh-2.6 リモート  
リポジトリ

master  
ブランチ



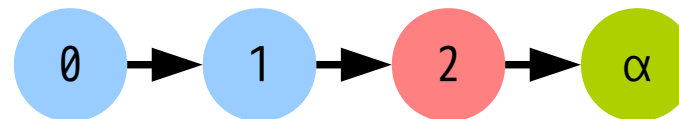
ローカル  
リポジトリ

origin/master  
ブランチ



git rebase origin

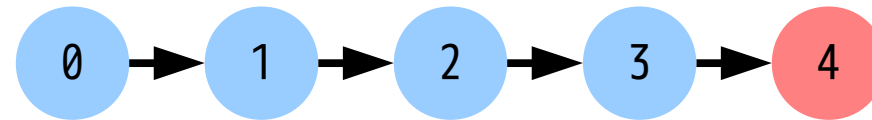
heads/master  
ブランチ



# 6. リモートリポジトリの状態に対してrebaseを行う

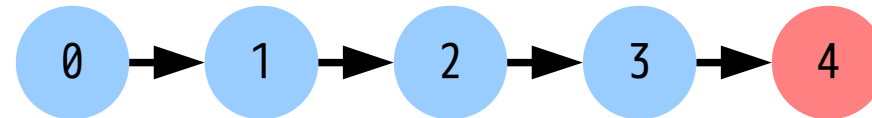
sh-2.6 リモート  
リポジトリ

master  
ブランチ



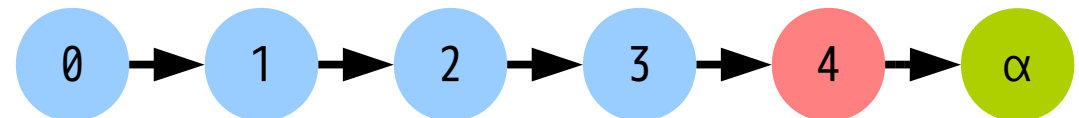
ローカル  
リポジトリ

origin/master  
ブランチ



git rebase origine

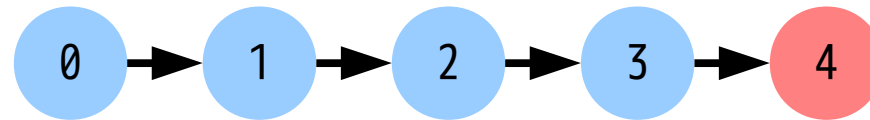
heads/master  
ブランチ



# 6. リモートリポジトリの状態に対してrebaseを行う

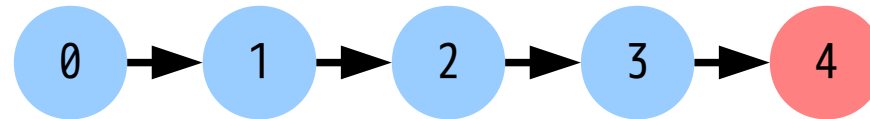
sh-2.6 リモート  
リポジトリ

master  
ブランチ



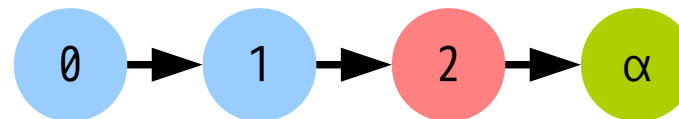
ローカル  
リポジトリ

origin/master  
ブランチ



git merge origin/master

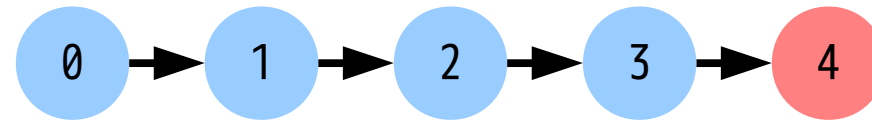
heads/master  
ブランチ



# 6. リモートリポジトリの状態に対してrebaseを行う

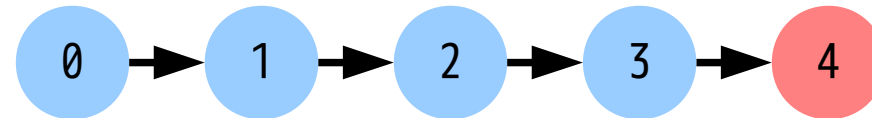
sh-2.6 リモート  
リポジトリ

master  
ブランチ



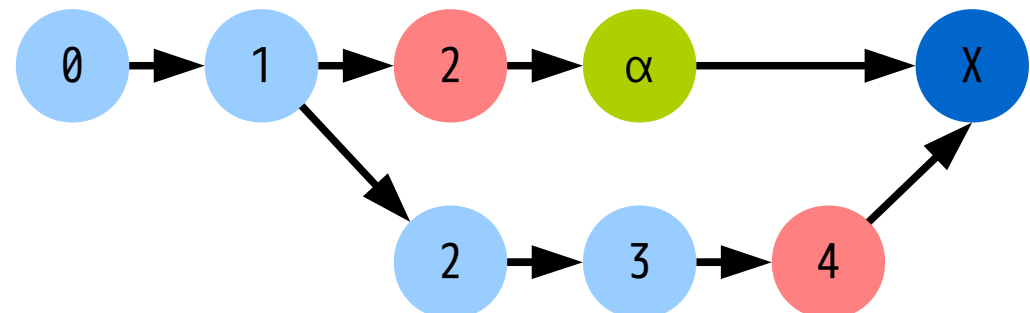
ローカル  
リポジトリ

origin/master  
ブランチ



git merge origin/master

heads/master  
ブランチ



## 7. 再度テスト

```
$ make ARCH=sh CROSS_COMPILE=sh4-linux-gnu-\  
-j8  
$ 実機でテスト
```

## 8. git format-patchコマンドでコミットからパッチメールを作成

```
$ git format-patch -o ../ origin  
../0001-sh-Add-MMU-dependency-for-kexec.patch
```



## 8. git format-patchコマンドでコミットからパッチメールを作成

```
$ cat \  
../0001-sh-Add-MMU-dependency-for-kexec.patch  
From f36b59d1a4fa1e29be606d0513b5f7fa6e720f79 Mon Sep 17 00:00:00 2001  
From: Nobuhiro Iwamatsu <iwamatsu.nobuhiro@renesas.com>  
Date: Thu, 26 Mar 2009 08:32:37 +0000  
Subject: [PATCH] sh: Add MMU dependency for kexec
```

```
Kexec is not likely to work on NON-MMU CPU because this added  
depend on MMU for kexec.
```

```
Signed-off-by: Nobuhiro Iwamatsu <iwamatsu.nobuhiro@renesas.com>
```

```
---
```

```
arch/sh/Kconfig |    2 +-  
1 files changed, 1 insertions(+), 1 deletions(-)
```

```
diff --git a/arch/sh/Kconfig b/arch/sh/Kconfig
```

```
index 8d50d52..19566c8 100644
```

```
--- a/arch/sh/Kconfig
```

```
+++ b/arch/sh/Kconfig
```

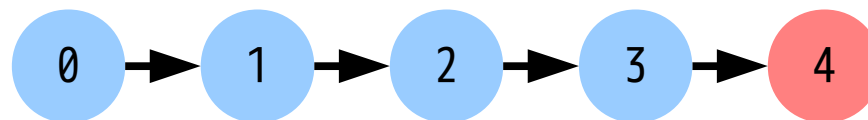
```
@@ -530,7 +530,7 @@ source kernel/Kconfig.hz
```

```
.....
```

# 8. git format-patchコマンドでコミットからパッチメールを作成

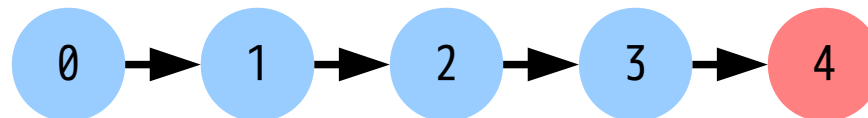
sh-2.6リモート  
リポジトリ

master  
ブランチ

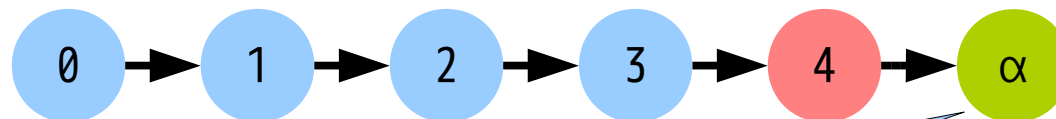


ローカル  
リポジトリ

origin/master  
ブランチ



heads/master  
ブランチ



現在のブランチとoriginとの差分を出力

## 9. 作成したパッチをメーリングリストに投稿する

```
$ git send-email \  
--to linux-sh@vger.kernel.org \  
--cc lethal@linux-sh.org \  
../0001-sh-Add-MMU-dependency-for-kexec.patch
```

## 2.2. 最新機能を試す

- 最新のドライバとか
- 最新の機能とか
- 試したい事があるというか、試す必要がある場合がある
- どのようにするか

例) sh-2.6のリポジトリをベースにしたブランチにbluetoothドライバのコードをマージしてテストをする

# 1. リモートリポジトリを追加する

BluetoothのGitリポジトリは

MAINTAINERS ファイルに書いてある

```
BLUETOOTH SUBSYSTEM
```

```
P: Marcel Holtmann
```

```
M: marcel@holtmann.org
```

```
L: linux-bluetooth@vger.kernel.org
```

```
W: http://www.bluez.org/
```

```
T: git
```

```
git://git.kernel.org/pub/scm/linux/kernel/git/holtmann/bluetooth-2.6.git
```

# 1. リモートリポジトリを追加する

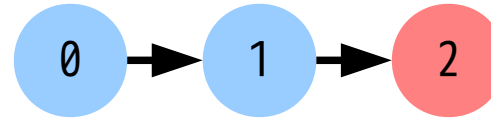
git remote コマンドでリモートリポジトリを追加

```
$ git remote add bluetooth \  
git://git.kernel.org/pub/scm/linux/kernel/gi  
t/holtmann/bluetooth-2.6.git
```

# 1. リモートリポジトリを追加する

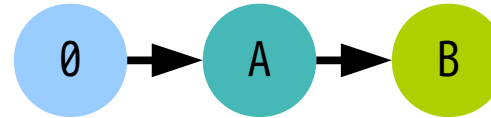
sh-2.6リモート  
リポジトリ

master  
ブランチ



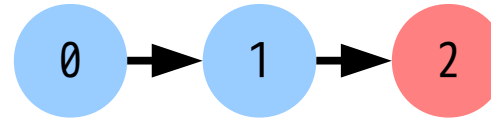
Bluetooth  
リモート  
リポジトリ

master  
ブランチ

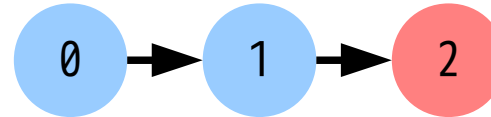


ローカル  
リポジトリ

origin/master  
ブランチ



heads/master  
ブランチ

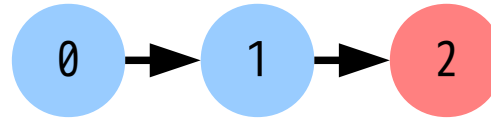




# 1. リモートリポジトリを追加する

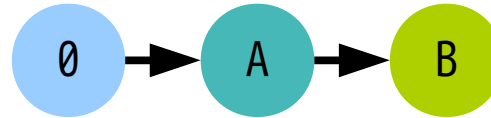
sh-2.6リモート  
リポジトリ

master  
ブランチ



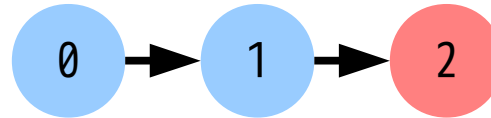
Bluetooth  
リモート  
リポジトリ

master  
ブランチ

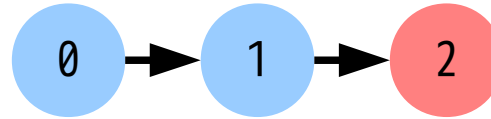


ローカル  
リポジトリ

origin/master  
ブランチ



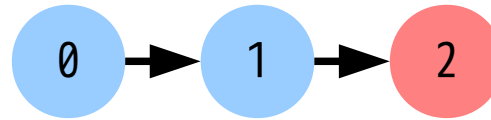
heads/master  
ブランチ



# 1. リモートリポジトリを追加する

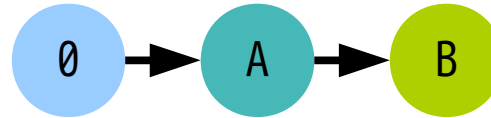
sh-2.6 リモート  
リポジトリ

master  
ブランチ



Bluetooth  
リモート  
リポジトリ

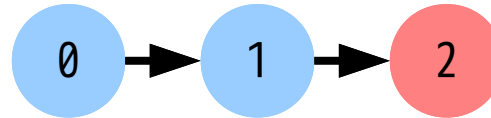
master  
ブランチ



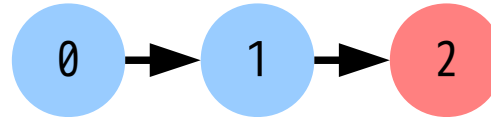
git remote add

ローカル  
リポジトリ

origin/master  
ブランチ



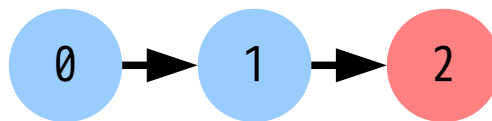
heads/master  
ブランチ



# 1. リモートリポジトリを追加する

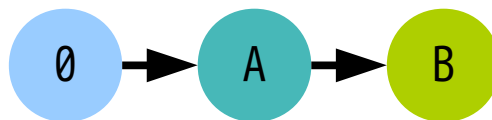
sh-2.6 リモート  
リポジトリ

master  
ブランチ



Bluetooth  
リモート  
リポジトリ

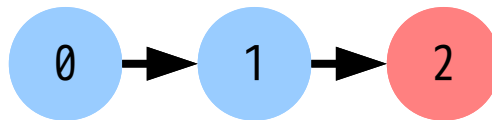
master  
ブランチ



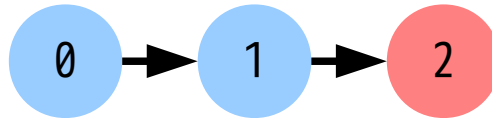
git remote add

ローカル  
リポジトリ

origin/master  
ブランチ



heads/master  
ブランチ



bluetooth

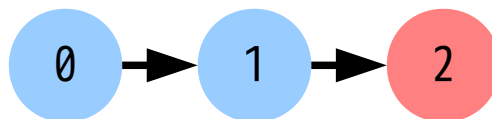
## 2. リモートリポジトリの状態をアップデートする

```
$ git remote update bluetooth
Updating bluetooth
From
git://git.kernel.org/pub/scm/linux/kernel/gi
t/holtmann/bluetooth-2.6.git
* [new branch]      master      -> \
                    bluetooth/master
```

## 2. リモートリポジトリの状態をアップデートする

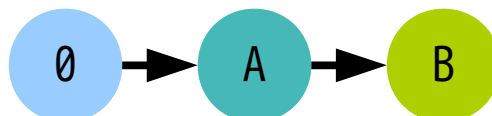
sh-2.6 リモート  
リポジトリ

master  
ブランチ



Bluetooth  
リモート  
リポジトリ

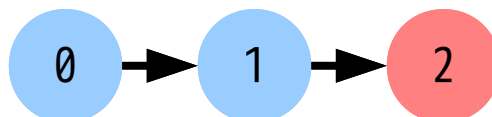
master  
ブランチ



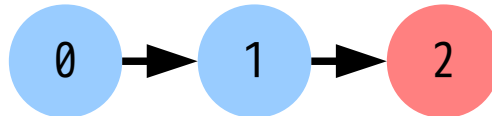
git remote update bluetooth

ローカル  
リポジトリ

origin/master  
ブランチ



heads/master  
ブランチ

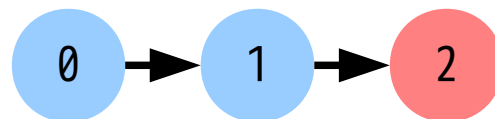


bluetooth

## 2. リモートリポジトリの状態をアップデートする

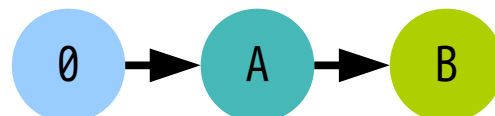
sh-2.6 リモート  
リポジトリ

master  
ブランチ



Bluetooth  
リモート  
リポジトリ

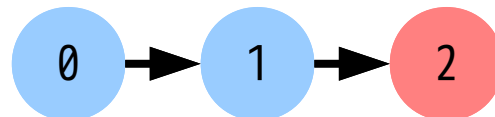
master  
ブランチ



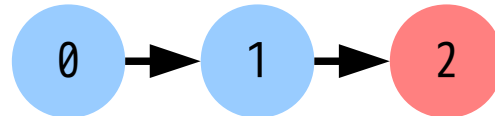
git remote update bluetooth

ローカル  
リポジトリ

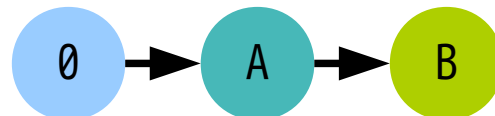
origin/master  
ブランチ



heads/master  
ブランチ



bluetooth/  
master ブランチ



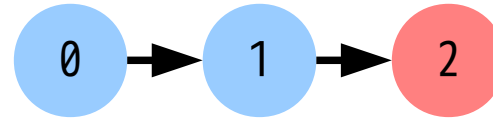
### 3. bluetoothの変更をmasterブランチに取り込む

```
$ git branch
* master
$ git merge bluetooth/master
Merge made by recursive.
Counting objects: 976200, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (147281/147281),
done.
Writing objects: 100% (976200/976200), done.
Total 976200 (delta 846940), reused 947801
(delta 820079)
.....
```

# 3. bluetoothの変更をmasterブランチに取り込む

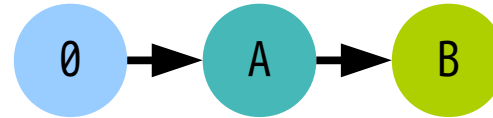
sh-2.6リモート  
リポジトリ

master  
ブランチ



Bluetooth  
リモート  
リポジトリ

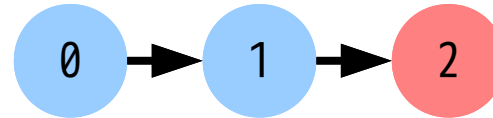
master  
ブランチ



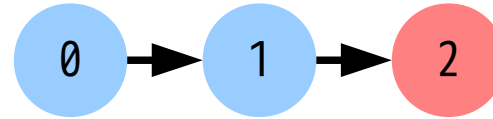
----- git merge bluetooth/master -----

ローカル  
リポジトリ

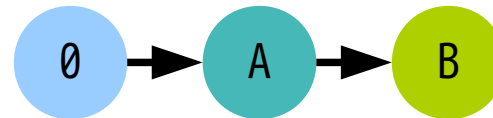
origin/master  
ブランチ



heads/master  
ブランチ



bluetooth/  
masterブランチ

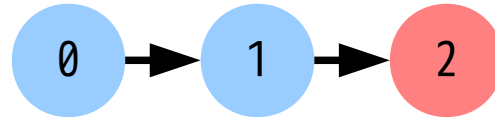




# 3. bluetoothの変更をmasterブランチに取り込む

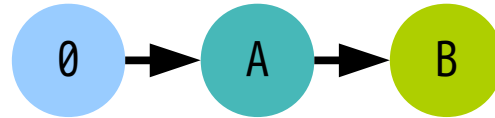
sh-2.6リモート  
リポジトリ

master  
ブランチ



Bluetooth  
リモート  
リポジトリ

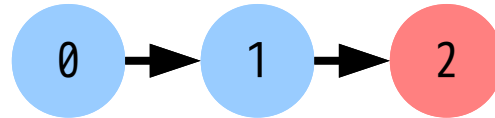
master  
ブランチ



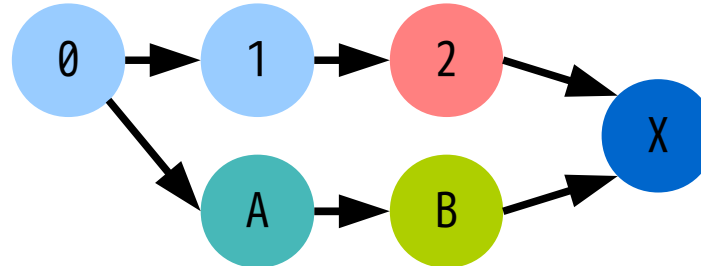
----- git merge bluetooth/master -----

ローカル  
リポジトリ

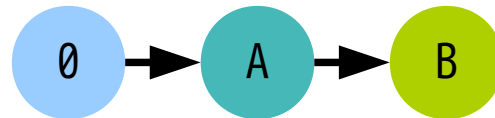
origin/master  
ブランチ



heads/master  
ブランチ



bluetooth/  
masterブランチ



### 3. bluetoothの変更をmasterブランチに取り込む

```
$ git show HEAD
commit
52b4ab9e6629884bf2a3f34dae5367a7ef5e9b4f
Merge: 138f025 4c71318
Author: Nobuhiro Iwamatsu
<iwamatsu.nobuhiro@renesas.com>
Date: Thu Jun 11 16:23:38 2009 +0900

Merge commit 'bluetooth/master' into
master
```

### 3. bluetoothの変更をmasterブランチに取り込む

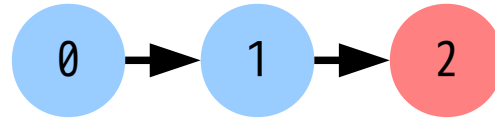
```
$ git show HEAD
commit
52b4ab9e6629884bf2a3f34dae5367a7ef5e9b4f
Merge: 138f025 4c71318
Author: Nobuhiro Iwamatsu
<iwamatsu.nobuhiro@renesas.com>
Date: Thu Jun 11 16:23:38 2009 +0900

Merge commit 'bluetooth/master' into
master
```

# 3. bluetoothの変更をmasterブランチに取り込む

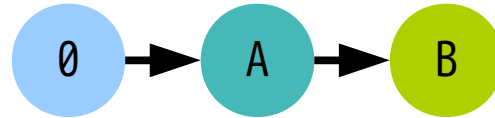
sh-2.6リモート  
リポジトリ

master  
ブランチ



Bluetooth  
リモート  
リポジトリ

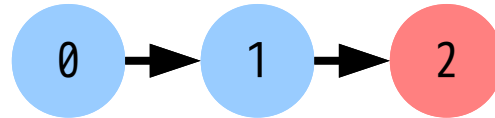
master  
ブランチ



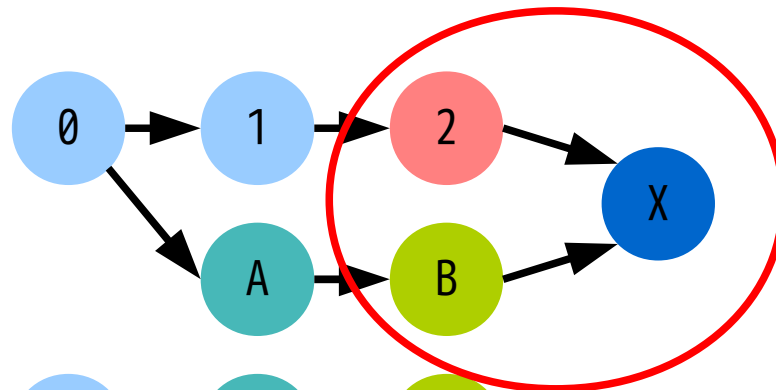
----- git merge bluetooth/master -----

ローカル  
リポジトリ

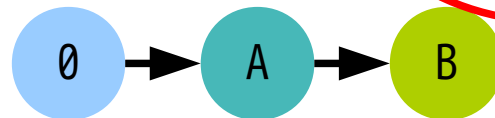
origin/master  
ブランチ



heads/master  
ブランチ



bluetooth/  
masterブランチ



## 4. テストを行う

```
$ make ARCH=sh CROSS_COMPILE=sh4-linux-gnu-\  
-j8  
$ 実機でテスト
```

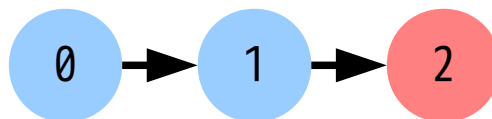
## 5. masterブランチを元に戻す

```
$ git reset --hard origin/master
```

# 5. masterブランチを元に戻す

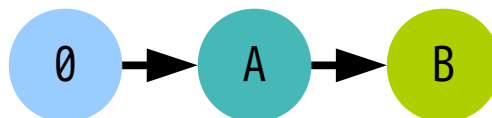
sh-2.6リモート  
リポジトリ

master  
ブランチ



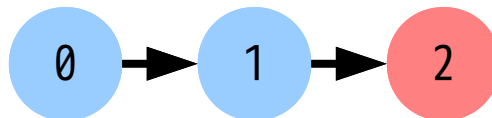
Bluetooth  
リモート  
リポジトリ

master  
ブランチ

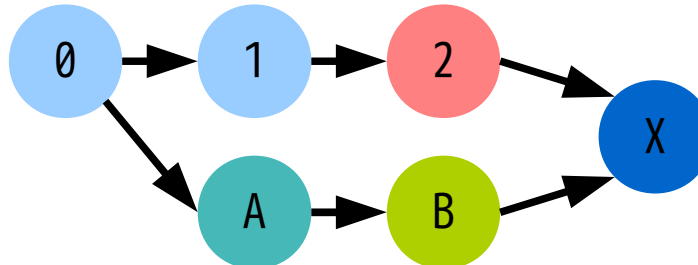


ローカル  
リポジトリ

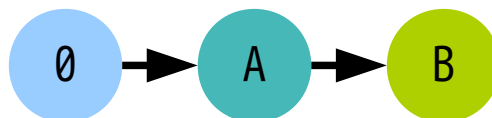
origin/master  
ブランチ



heads/master  
ブランチ



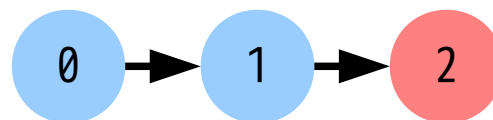
bluetooth/  
masterブランチ



# 5. masterブランチを元に戻す

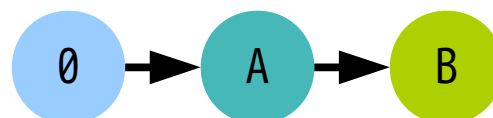
sh-2.6リモート  
リポジトリ

master  
ブランチ



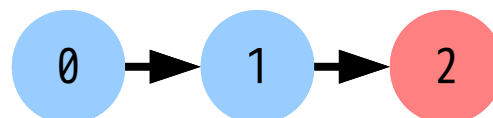
Bluetooth  
リモート  
リポジトリ

master  
ブランチ



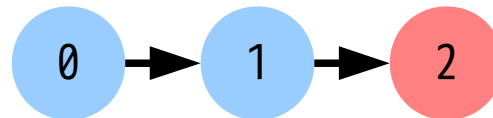
ローカル  
リポジトリ

origin/master  
ブランチ

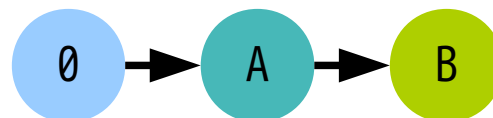


リセットする

heads/master  
ブランチ



bluetooth/  
masterブランチ





# まとめ

- リモートリポジトリとローカルリポジトリを理解しましょう
- パッチを送るときは rebaseして最新のコミットに対してのパッチを送るようにしましょう
- リモートリポジトリは git remote を使うと楽です
- merge と rebase は使い分けましょう

# その他知っておくとよい事

- Git オブジェクトの仕組み
- index の動き
- コンフリクト時の復旧方法
- git bisect/ blame の使い方

# 質疑応答

ご清聴ありがとうございました。