MITSUBISHI ELECTRIC
Changes for the Better

OSAKA NDS EMBEDDED LINUX CROSS FORUM #4
2017- 2-10
http://www.nds-osk.co.jp/forum/onlcf4.html

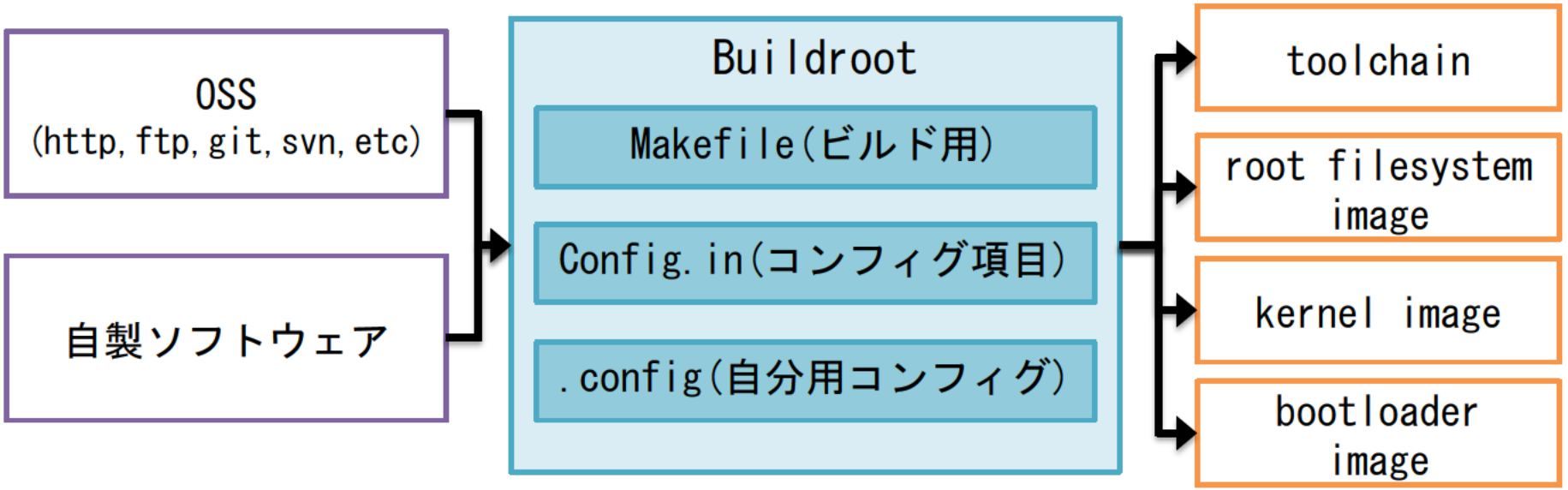
組込みLinux向けユーザランド構築ツール Buildroot概要と使い方

三菱電機(株)　情報技術総合研究所
リアルタイムプラットフォーム技術部
茂田井寛隆

MITSUBISHI ELECTRIC
Changes for the Better

Buildrootとは

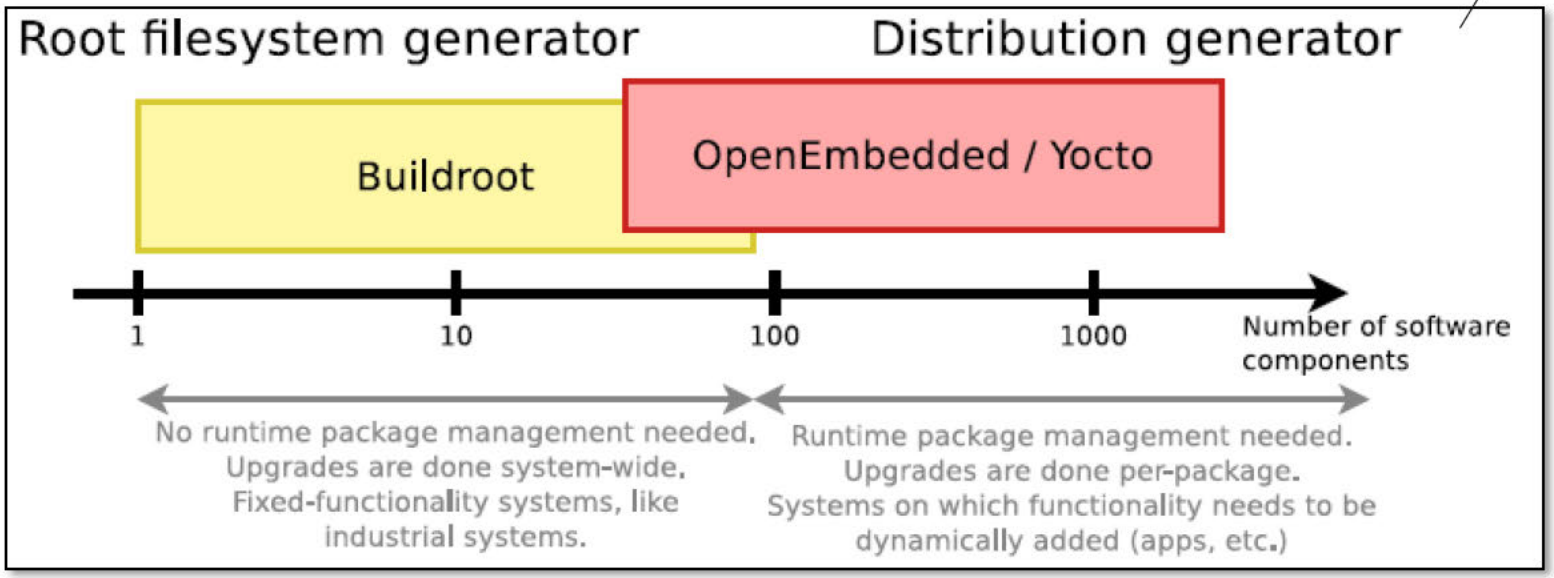
- Buildroot: Making Embedded Linux easy.
  https://git.busybox.net/buildroot/
  - ユーザランド、クロスコンパイラ、 Linuxカーネル、ブートローダを生成するツール
  - カスタマイズはメニューで変更できる
  - lego mindstorm EV3 対応済み
    - board/lego/ev3/readme.txt



Buildrootとは

- コミュニティ
  - 歴史
    - 2001：uClibc開発者によりuClibcテスト用として開発
    - 2005：開発者の増加
    - 2009：メンテナー交代と共に3ヶ月リリースが定着
  - 位置づけ

[ELC2012]Buildroot: A Nice, Simple, and Efficient Embedded Linux Build System より図を引用(http://elinux.org/images/9/9e/Buildroot2.pdf)



マニュアル　https://buildroot.org/downloads/manual/manual.html

MITSUBISHI ELECTRIC
Changes for the Better

使用例

- @Ubuntu 16.04 i386
  - # apt install which bash patch sed make gcc g++
    binutils build-essential gzip bzip2 tar
    perl cpio python unzip rsync git wget
  - $ make lego_ev3_defconfig
  - $ make
  - $ ls –lh output/images/sdcard.img
    -rw-r--r-- 1 motai 32M output/images/sdcard.img
  - $ file output/images/sdcard.img
    - DOS/MBR boot sector;
    - partition 1 : FAT32(0xc), 32768 sectors;
    - partition 2 : ext3(0x83), 23874 sectors

MITSUBISHI ELECTRIC
Changes for the Better

Buildroot - make

- 何をしているか
  1. ソースコードのダウンロード
  2. クロスコンパイラ等ツールチェーンのコンフィグ、ビルド、インストール
     - 既存ツールチェーンのインポート
  3. 選択したパッケージの configure, ビルド、インストール
  4. カーネルイメージのビルド(選択時)
  5. ブートローダのビルド(選択時)
  6. rootfsの生成

Buildroot

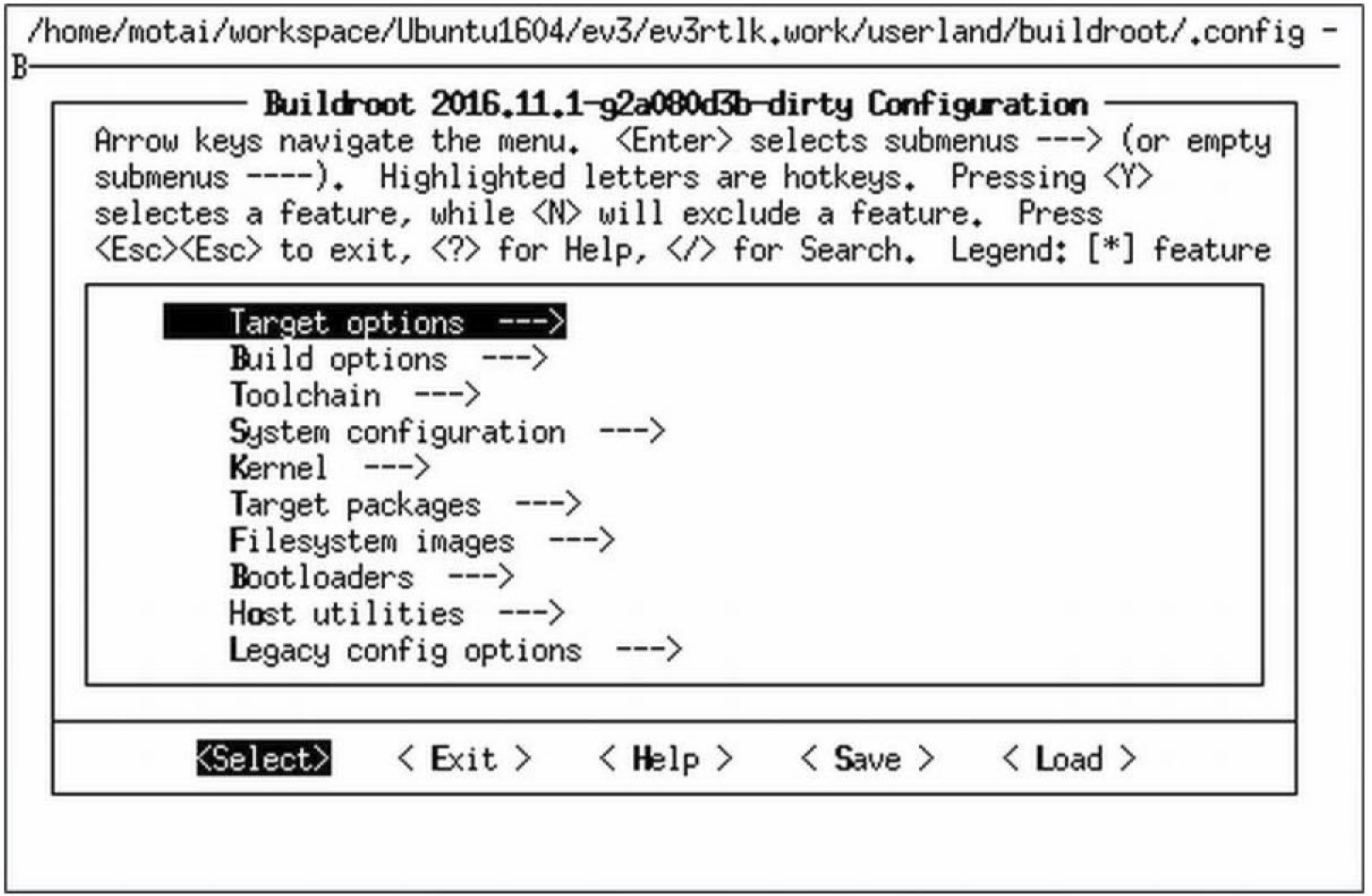
- output/

パス	中身
images/	カーネル，ブートローダ，ルートファイルシステムのイメージ
build/	コンパイル場所
staging/	コンパイル用ルートファイルシステム（ヘッダ、ライブラリ等）
target/	ターゲット用ルートファイルシステム（ライブラリ等）
host/	ホスト環境用ツールチェイン（クロスコンパイラ等）

Buildroot

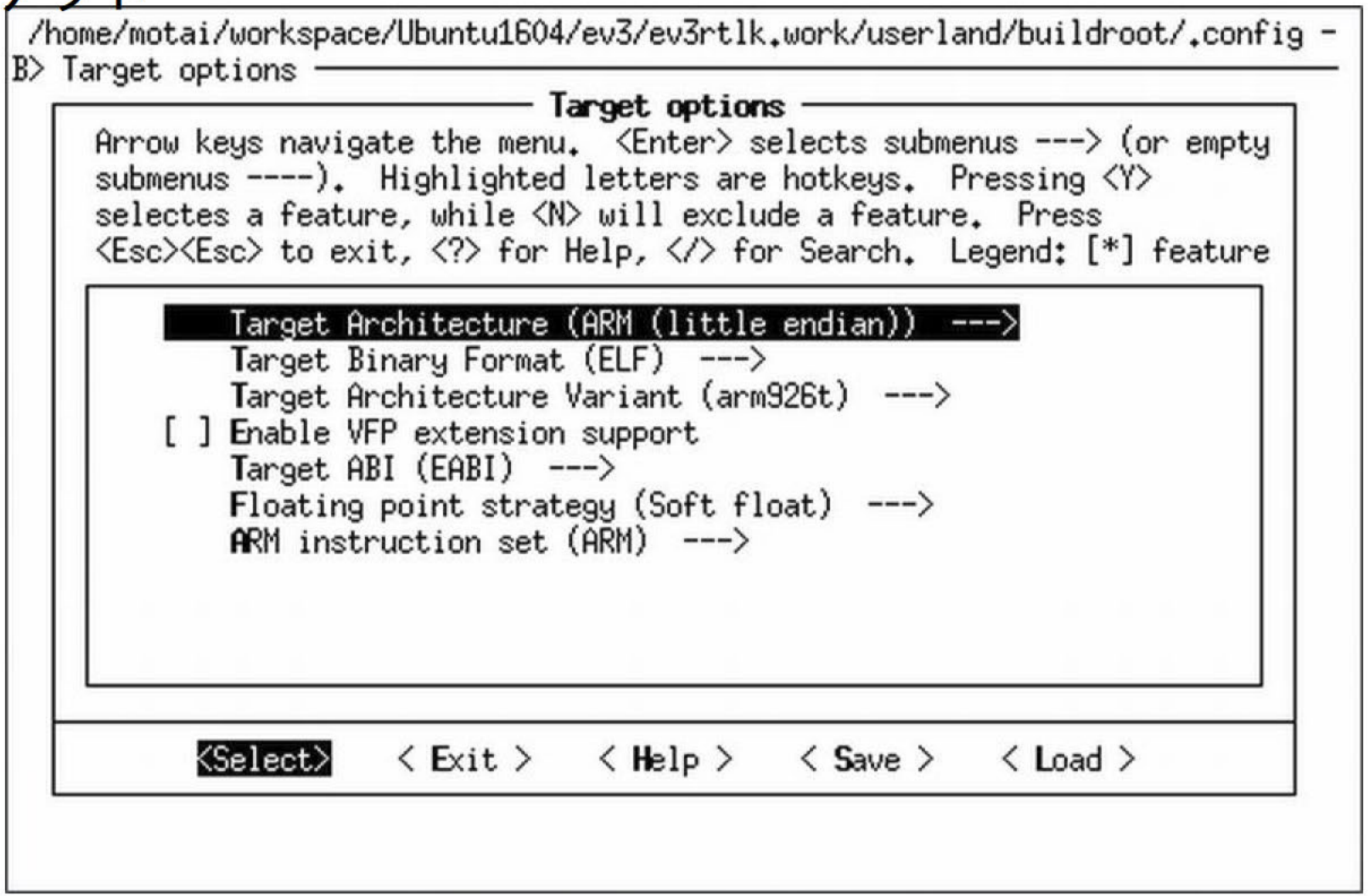
- カスタマイズ

  $ make <menuconfig | nconfig | gconfig | xconfig>



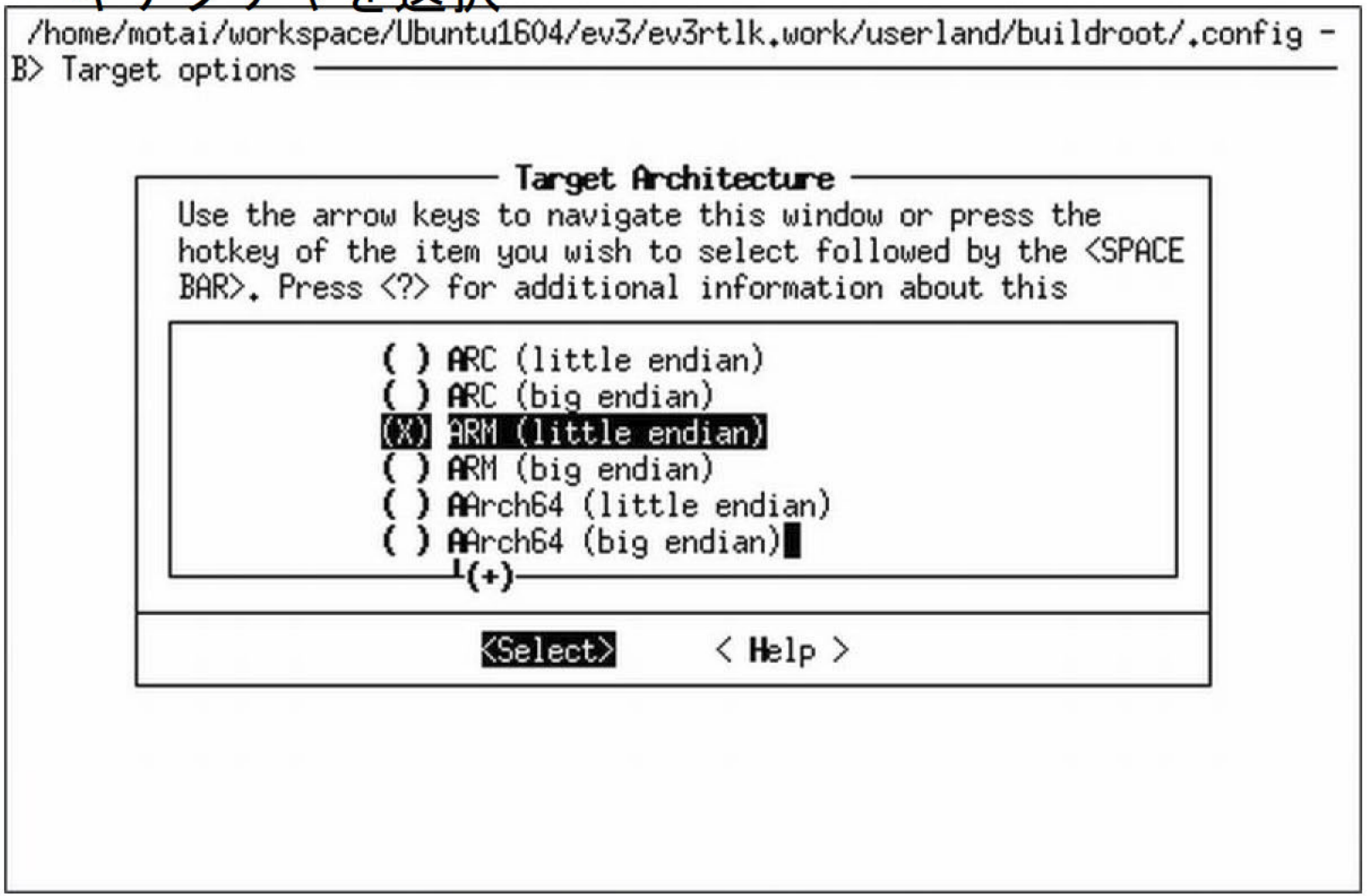
Buildroot

- ターゲット



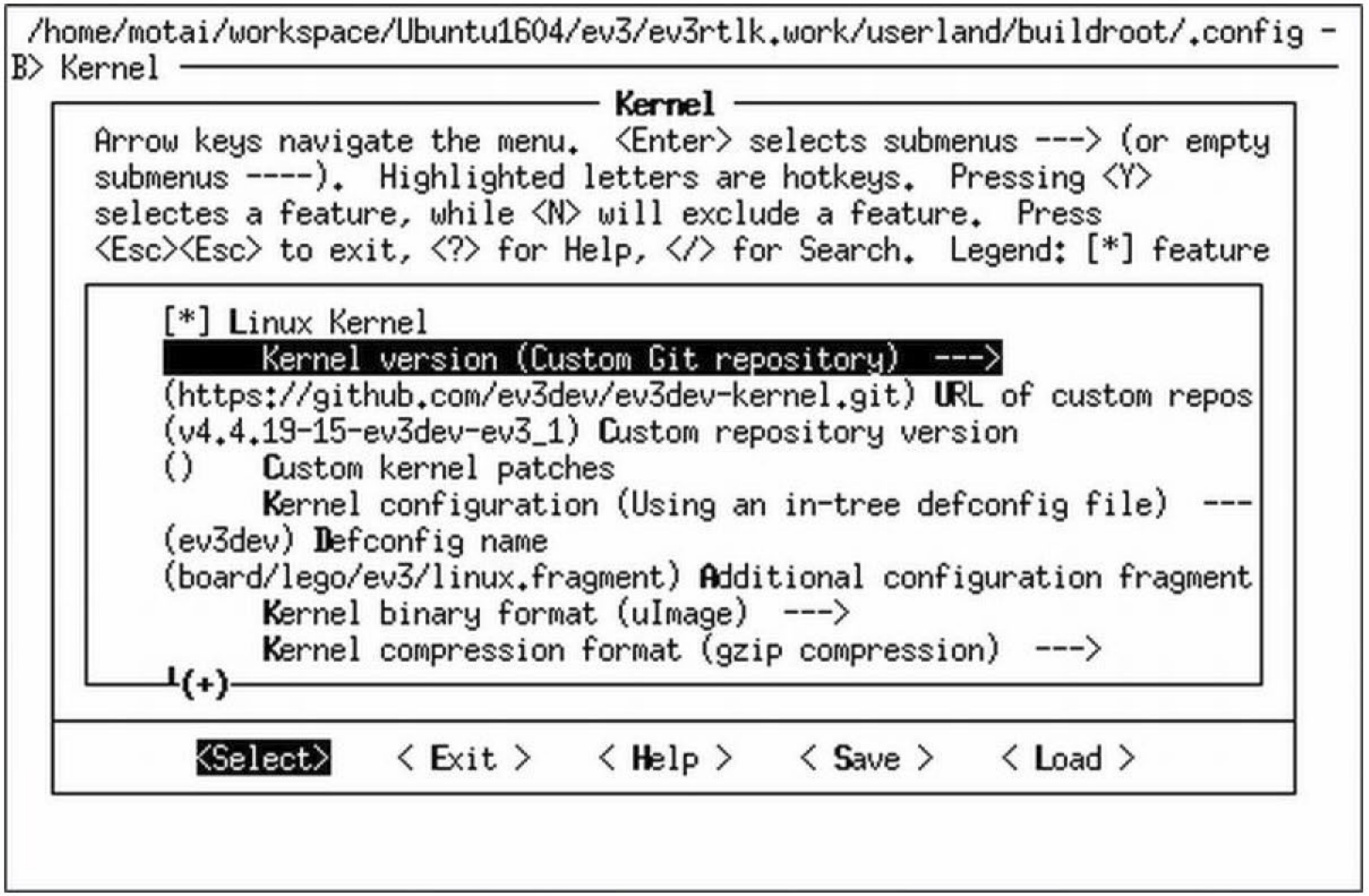
Buildroot

- CPUアーキテクチャを選択



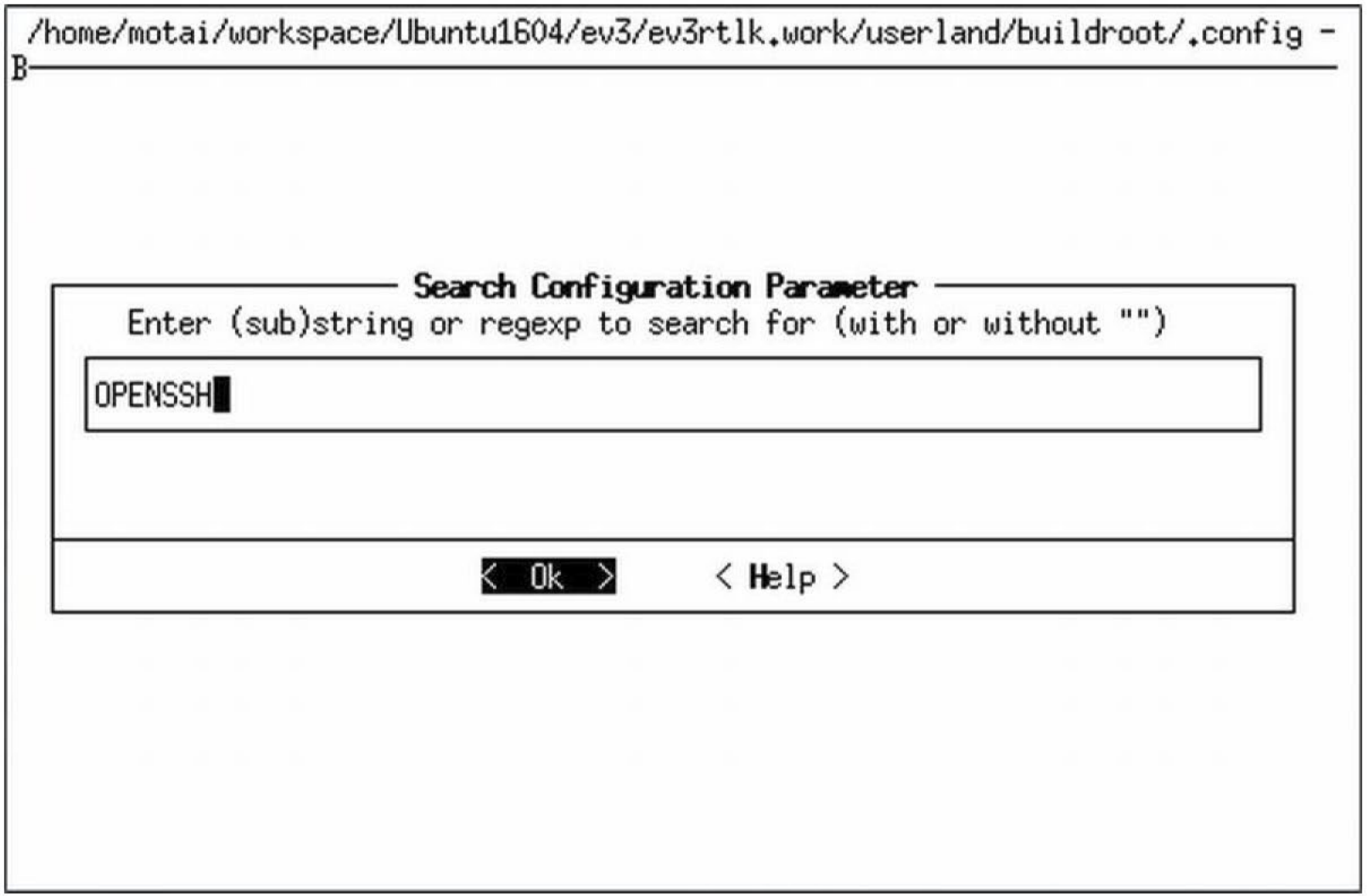
Buidroot

- カーネルをコンパイルする場合。git指定可能。



Buildroot

- 検索：/ or Ctrl+F (menuconfig, xconfigのみ)

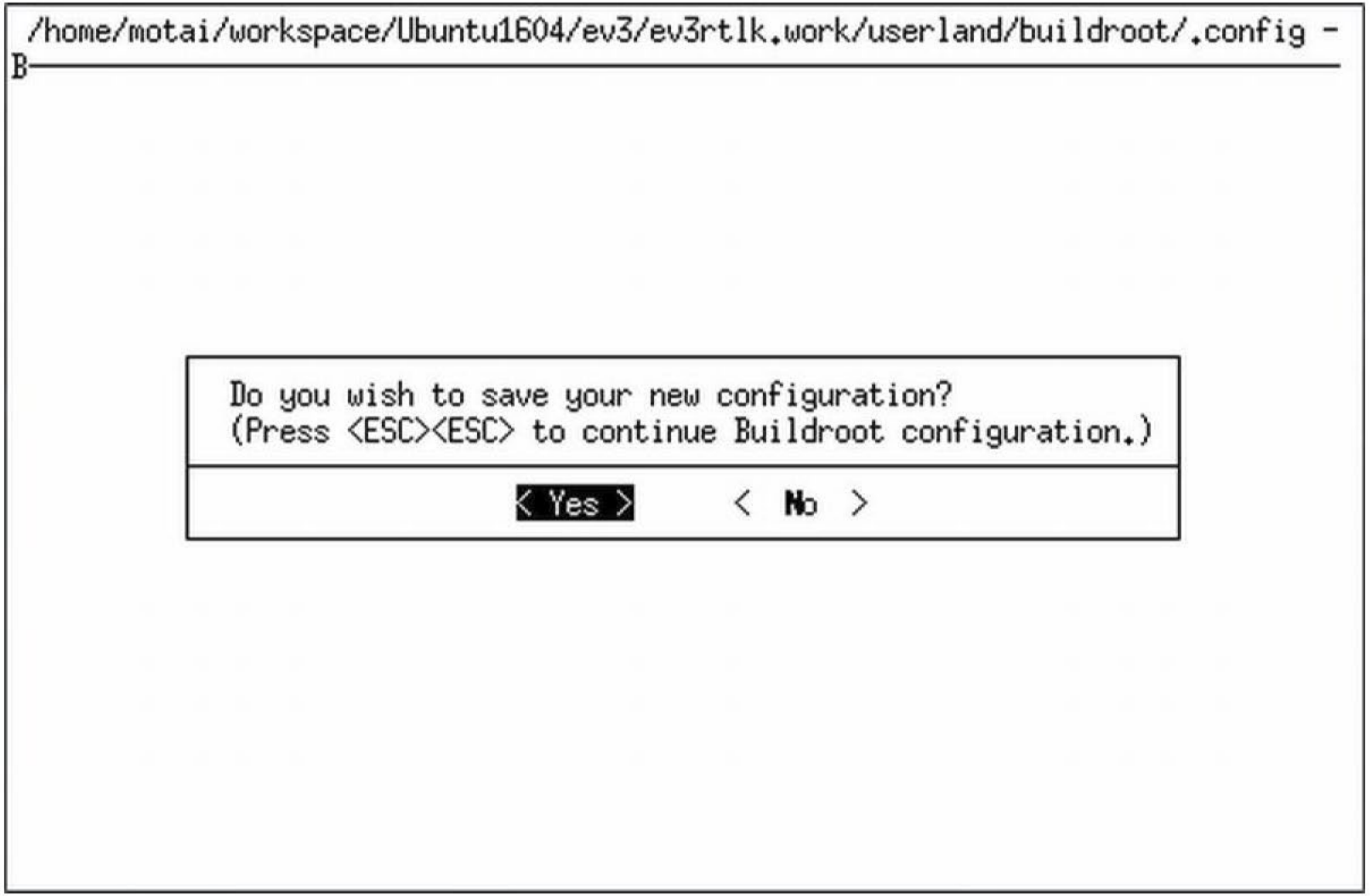


Buildroot

- 検索：/ or Ctrl+F (menuconfig, xconfigのみ)

Buildroot

- .configに保存



BUILDROOT メニュー

1. コンパイラ
2. /dev 管理
3. init system
4. 主要パッケージ一覧

Internal toolchain

- コンパイラ
  - gcc: gcc-4.x.x, gcc-5.x.x, gcc-6.x.x

- 標準Cライブラリ
  - GNU Libc ： GNU Cライブラリ
  - uClibc-ng： 組込みLinux向け小型Cライブラリ
    - Alpha, ARC, ARM, AVR32, Blackfin, CRIS, FR-V, H8/300, HPPA, i386, IA64, LM32, M68K/Coldfire, Metag, Microblaze, MIPS, MIPS64, NDS32, NIOS2, OpenRISC, PowerPC, Sparc, SuperH, X86_64 and XTENSA
  - musl ： Linuxの標準Cライブラリ
    - x86, x86_64, ARM, MIPS, Microblaze, PowerPC

MITSUBISHI ELECTRIC
Changes for the Better

External toolchain

- 既存ツールチェイン
  - 有名で試験済みのツールチェインを使いたいとき

- 確認済み：CodeSourcery, Linaro, crosstools-NG
  - sysroot 機能必須

- 未対応：OpenEmbedded, Yocto, distribution toolchain
  - コンパイル済みライブラリが含まれていて修正できない
  - x86ターゲットの場合もクロスコンパイラが必須

/dev 管理

- devノード
  - ① Static using device table
  - ② Dynamic using devtmpfs only (推奨)
    - デバイス発見次第、カーネルにより/dev配下が自動生成
    - kernel 2.6.32以降
    - CONFIG_DEVTMPFS=y && CONFIG_DEVTMPFS_MOUNT=y
  - ③ Dynamic using devtmpfs + mdev
    - BusyBox付属の軽量実装版 udev
      - /etc/mdev.confに従い、パーミッション/所有者を設定
    - FirmwareへのPushリクエスト機能あり
    - CONFIG_DEVTMPFS=y && CONFIG_DEVTMPFS_MOUNT=y

MITSUBISHI ELECTRIC
Changes for the Better

/dev 管理

- devノード（続き）
  - ④ Dynamic using devtmpfs + eudev
    - スタンドアローン版udev
      - 一般的なディストリビューションのudevと同じ（Systemdの一部）

- init systemがsystemdの場合はudevになる。

init system

① BusyBox（推奨）
- BR2_INIT_BUSYBOX=y
- /etc/inittab
  - デフォルトであれば、/etc/init.d/rcSスクリプトを実行し、gettyを立ち上げ。

② Systemd
- Cgroups、snapshot/restoreと多機能
- 並行起動、ソケット使用、D-Bus、udev

③ SystemV
- Upstart/Systemdに未対応なプログラム用
- /etc/inittab　(BusyBox形式と少し違う)

MITSUBISHI ELECTRIC
Changes for the Better

主要パッケージ一覧

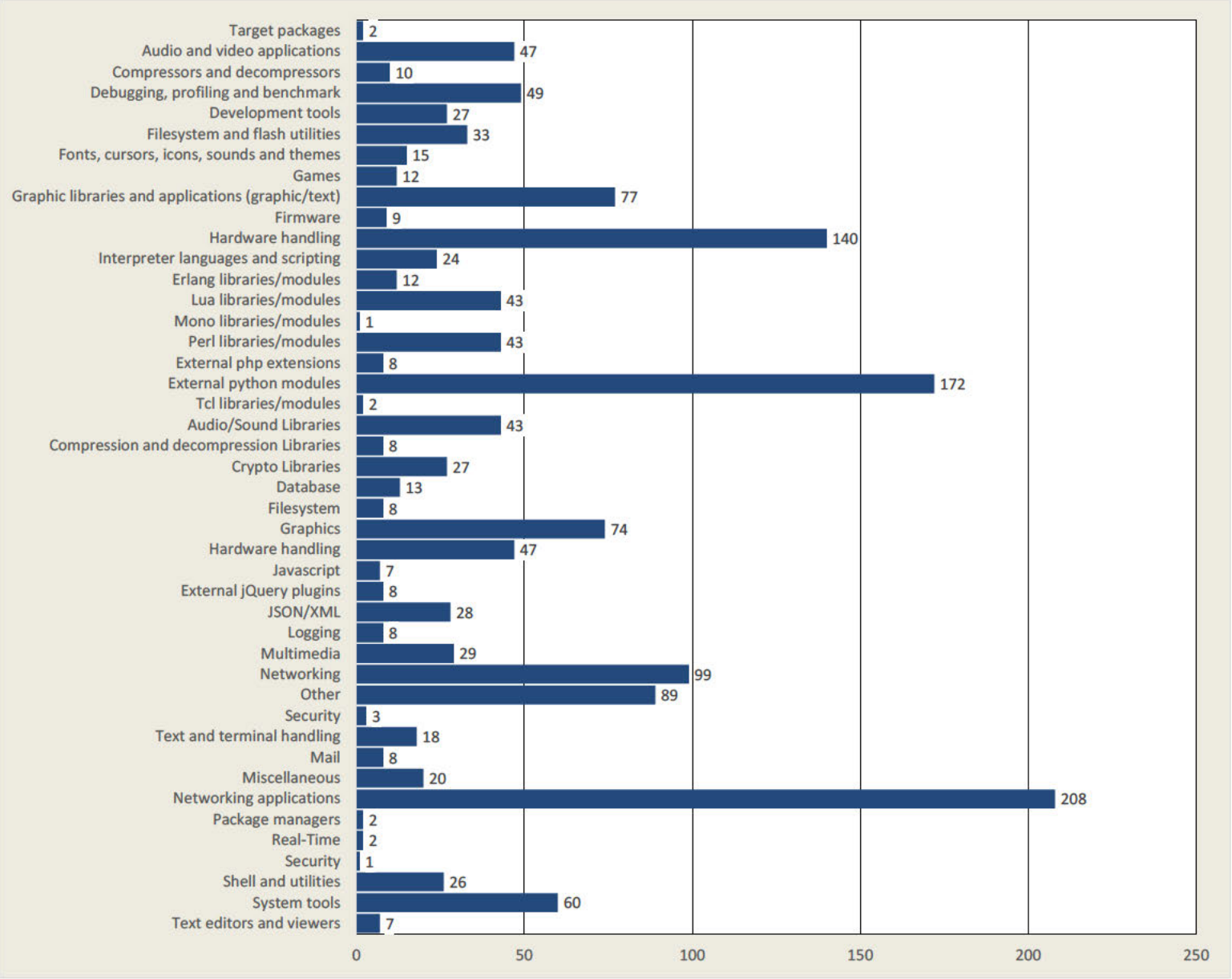
● 1569パッケージ

alsa	gstreamer	vlc	p7zip	xz	dhrystone	gdb
iozone	kexec	latencytop	lmbench	lttng	netperf	oprofile
rt-tests	ramspeed	stress	whetstone	trace-cmd	valgrind	aufs
cifs-utils	fwup	mmc-utils	ntfs-3g	sshfs	unionfs	dejavu
icon-theme	sound-theme	gnuchess	supertuxkart	fswebcam	glmark2	mesa3d-demo
qt5	rrdtool	directfb	fbterm	imagemagick	mesa3d	sdl
x11r7	docker	synergy	enlightenment	fluxbox	openbox	metacity
hdparm	i2c-tools	irda-utils	lm-sensors	minicom	powertop	erlang
lua	perl	python	php	ruby	openssl	beecrypt
mbedtls	berkeleydb	mysql	postgresql	sqlite	mongodb	gamin
freetype	wayland	pixman	libpng	jquery	json	bitstream
x264	openssh	openldap	libselinux	ncurses	exim	mutt
qemu	wine	clamav	apache	avahi	dhcpcd	iftop
iperf	ipsec-tools	lighttpd	lrzsz	nginx	nuttcp	openntpd
proftpd	rsync	samba4	tcpdump	wireshark	wireless_tools	wget
bash	zsh	xen	uemacs	vim	cgroups	lxc

詳しくはこちら　https://git.busybox.net/buildroot/tree/package

MITSUBISHI ELECTRIC
Changes for the Better

パッケージ分類



MITSUBISHI ELECTRIC
Changes for the Better

defconfig

- 132defconfig

altera_sockit	arm_juno	armadeus	at91sam9xx
atmel_sama5dxx	beaglebone	chromebook	cubieboard2
freescale_imx31	freescale_imx7	freescale_p1010	galileo
grinn_liteboard	lego_ev3	minnowboard	mx25pdk
mx6cubox	nitrogen6x	odroidc2	olimex_a20_olinuxino
openblocks_a6	orangepipc	pandaboard	pc_x86_64_bios/efi
qemu_<arch>	raspberrypi0/2/3	riotboard	roseapplepi
s6lx9_microboard	snps_axs101	stm32f4x9_disco	ts5x00
via_imx6_vab820	wandboard	warp7	zynq_microzed
…			

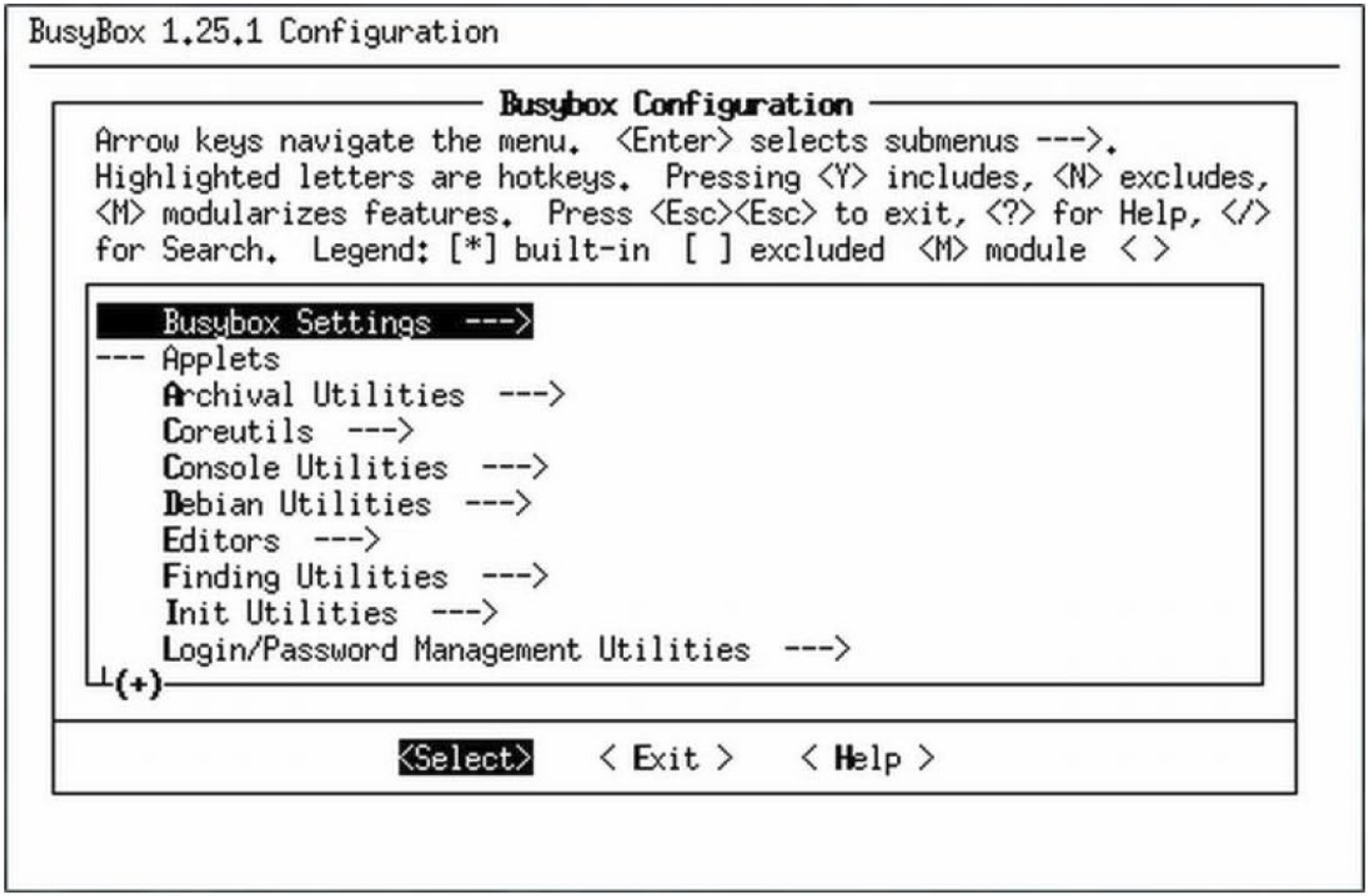
詳しくはこちら　https://git.busybox.net/buildroot/tree/configs

各パッケージのコンフィグ変更

BusyBox / uClibc / Linux Kernel

BusyBox

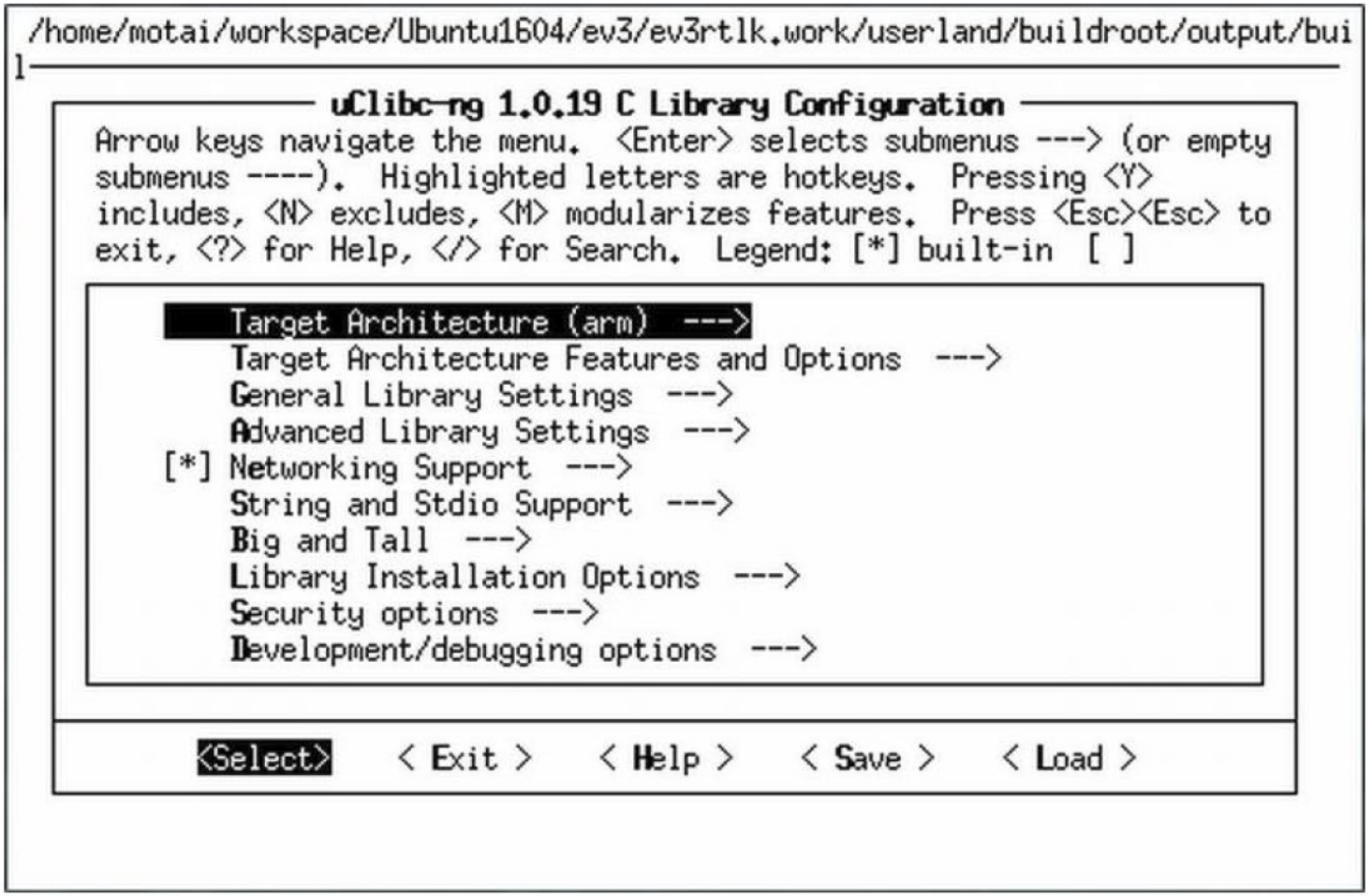
- $ make busybox-menuconfig



- BR2_PACKAGE_BUSYBOX_CONFIG="my_busybox.conf"

uClibc

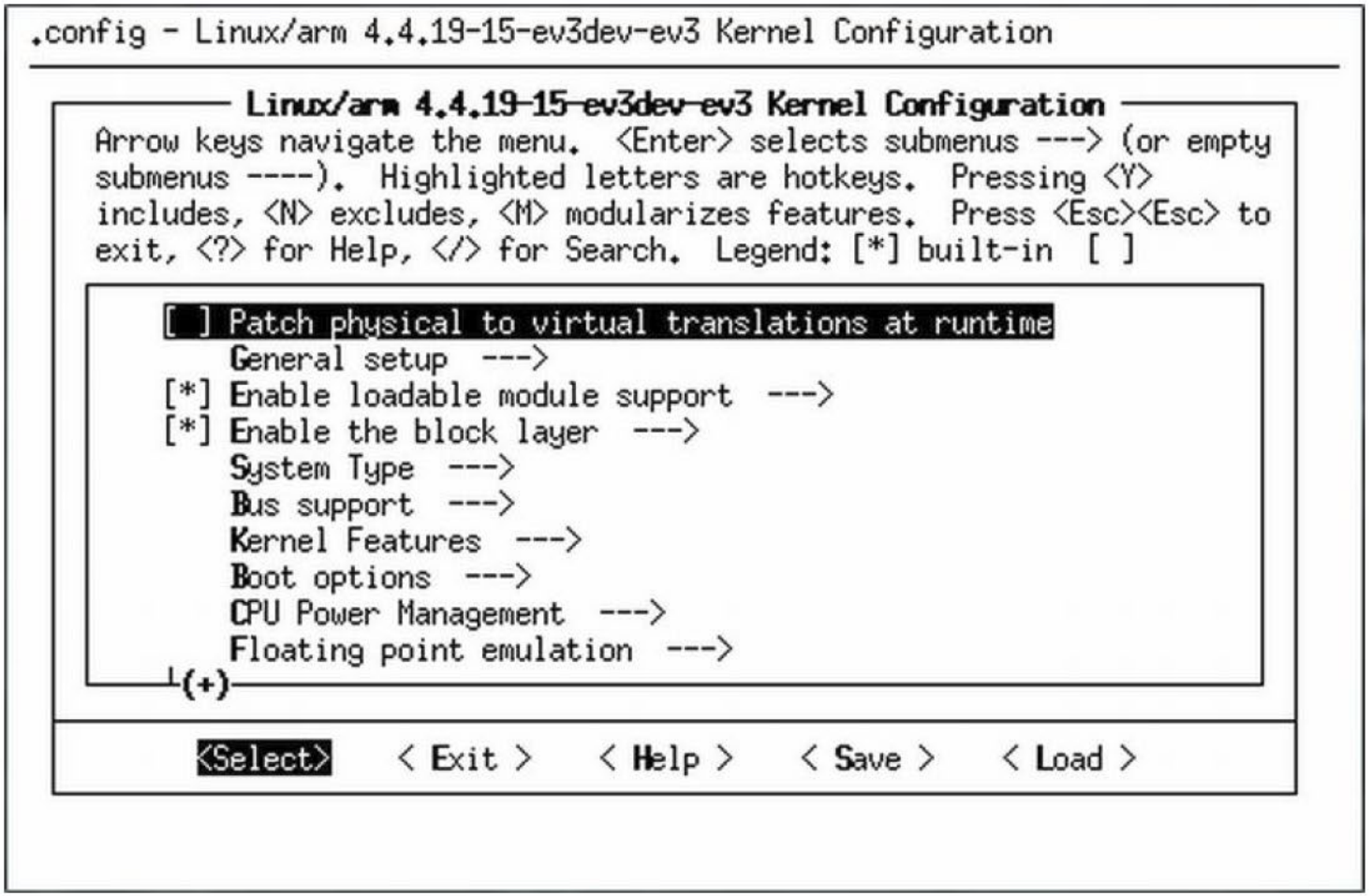
- $ make uclibc-menuconfig



- BR2_UCLIBC_CONFIG="my_uClibc.conf"

Configuration: Linux kernel

- $ make linux-menuconfig



- BR2_LINUX_KERNEL_USE_CUSTOM_CONFIG="my_dotconfig",
  BR2_LINUX_KERNEL_USE_DEFCONFIG="my_defconfig"

ビルド

ディレクトリ構成
ビルドの仕方
フルリビルドが必要な場合
各パッケージのみリビルドする場合

MITSUBISHI ELECTRIC
Changes for the Better

ディレクトリ構成

- ルートはMAKEFILE,README,COPYINGなど

	パス	内容	補足
1	toolchain/	クロスコンパイル用ツールチェイン	binutils, gcc, kernel-headers, uClibc
2	arch/	プロセッサアーキテクチャ	aarch64, arc, arm, bfin, m68k, microblaze, mips, nios2, powerpc, sh, sparc, x86, xtensa
3	package/	user-space tools and libraries	パッケージ毎にディレクトリ
4	linux/	Linuxカーネル	
5	boot/	ブートローダ	arm-trusted-firmware, barebox, grub2, uboot, xloader, など
6	system/	ターゲットファイルシステムスケルトン	
7	fs/	ファイルシステムイメージ生成用ツール	axfs, cloop, cpio, cramfs, ext2, initramfs, iso9660, jffs2, romfs, squashfs, tar, ubifs, yaffs2
8	dl/	ソースコードダウンロード先	
9	output/	生成先	

ビルドの仕方

- $ make help

```
Cleaning:
  clean           - delete all files created by build
  distclean       - delete all non-source files (including .config)

Build:
  all             - make world
  toolchain       - build toolchain
      :
    <略>
```

- ビルド　(スライド3，スライド4 もご参照ください)

```
$ make
```

- フルリビルド

```
$ make clean all
```

フルリビルドが必要な場合

- Buildrootの方針
  - パッケージの依存関係は定義できるが、開発者に対応を強いるのは困難。ユーザ側で対応する。

- ユーザに知っておいて欲しいこと
  1. ターゲットのアーキテクチャを変更：フルリビルド
  2. ツールチェインを変更：フルリビルド
  3. パッケージを追加：リビルド
     - 追加により他パッケージへ影響がある場合は除く
       例)パッケージAがあれば機能が有効になるパッケージBがある
  4. パッケージの除去：（急がないが）フルリビルド推奨

フルリビルドが必要な場合

- ユーザに知っておいて欲しいこと（続き）
  - ⑤ パッケージ内オプションを変更：フルリビルド
  - ⑥ rootfsスケルトンファイルを変更：フルリビルド
    - overlay，post-build/imageスクリプトの変更はリビルドのみ

- ビルドエラーになった場合はフルリビルド試行

MITSUBISHI ELECTRIC
Changes for the Better

各パッケージのリビルド方法

- パッケージのビルドディレクトリを消す（簡単で確実）
  - $ rm –rf output/build/<package>
  - $ make <package>
- パッケージのビルドディレクトリをクリーンする
  - $ make <package>-dirclean
- パッケージをリビルド、インストールする
  - $ make <package>-rebuild
- コンフィグからやり直す
  - $ make <package>reconfigure

MITSUBISHI ELECTRIC
Changes for the Better

ソースコードのダウンロード

- $ make source
- $ ls dl

```
DirectFB-1.7.7.tar.gz      LuaJIT-2.0.4.tar.gz      Python-2.7.13.tar.xz
autoconf-2.69.tar.xz       automake-1.15.tar.xz     bash-4.3.30.tar.gz
binutils-2.26.1.tar.bz2    bison-3.0.4.tar.xz       bonnie++-1.03e.tgz
busybox-1.25.1.tar.bz2     cairo-1.14.6.tar.xz      ...
```

- バージョン管理したい場合に使用

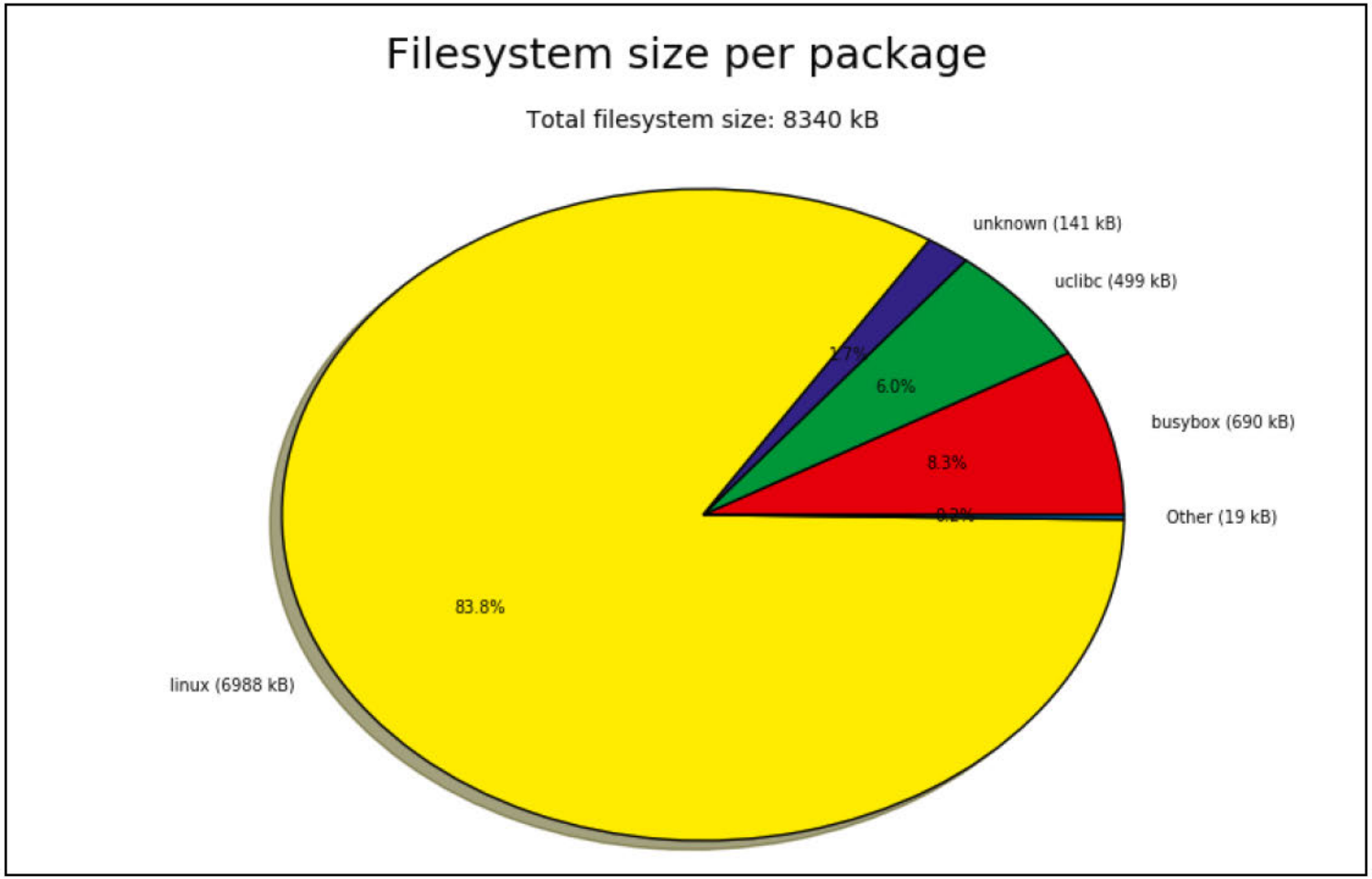
各パッケージに関するmake一覧

- $ make <package>-<target>

target	内容
source	ソースコードをダウンロード(tarball, git clone, etc)
depends	パッケージの依存関係を確認
extract	ビルドディレクトリへソースコードを展開
patch	パッチ適用
configure	コンフィグの実行
build	ビルド（コンパイル）
install-staging	ステージディレクトリへインストール
install-target	ターゲットディレクトリへインストール
install	install-staging, install-target と同じ
dirclean	ビルドディレクトリの削除
reinstall	installコマンドの再実行
rebuild	ビルドの再実行
reconfigure	コンフィグの再実行

各パッケージのrootfsに占める割合

- # apt install python-matplotlib
- $ make graph-size
  - output/graphs/graph-size.pdf



ライセンス確認 [1/2]

- 該当ファイルを検索。
  $ make legal-info
  - output/legal-info/manifest.csv

PACKAGE	VERSION	LICENSE	LICENSE FILES	SOURCE ARCHIVE	SOURCE SITE
uclibc	1.0.19	LGPLv2.1+	COPYING.LIB	uClibc-ng-1.0.19.tar.xz	http://downloads.uclibc-ng.org/releases/1.0.19
linux-headers	v4.4.19-15-ev3dev-ev3_1	GPLv2	COPYING	linux-v4.4.19-15-ev3dev-ev3_1.tar.gz	https://github.com/ev3dev/ev3dev-kernel.git
busybox	1.25.1	GPLv2	LICENSE	busybox-1.25.1.tar.bz2	http://www.busybox.net/downloads
ev3dev-linux-drivers	0e551eb25ae8600c1f178814781bfb42dc835496	GPLv2		ev3dev-linux-drivers-0e551eb25ae8600c1f178814781bfb42dc835496.tar.gz	https://github.com/ev3dev/lego-linux-drivers/archive/0e551eb25ae8600c1f178814781bfb42dc835496
uboot	2016.09.01	GPLv2+	Licenses/gpl-2.0.txt	u-boot-2016.09.01.tar.bz2	ftp://ftp.denx.de/pub/u-boot
linux	v4.4.19-15-ev3dev-ev3_1	GPLv2	COPYING	linux-v4.4.19-15-ev3dev-ev3_1.tar.gz	https://github.com/ev3dev/ev3dev-kernel.git

ライセンス確認 [2/2]

- output/legal-info/

```
|--  README
|--  buildroot.config
|--  legal-info.sha256
|--  host-manifest.csv
|--  host-licenses/      ← COPYING,LICENSE,README ファイル
|--  host-sources/       ← tarball,patchファイル
|--  manifest.csv
|--  licenses/          ← COPYING,LICENSE,README ファイル
`--  sources/           ← tarball,patchファイル
```

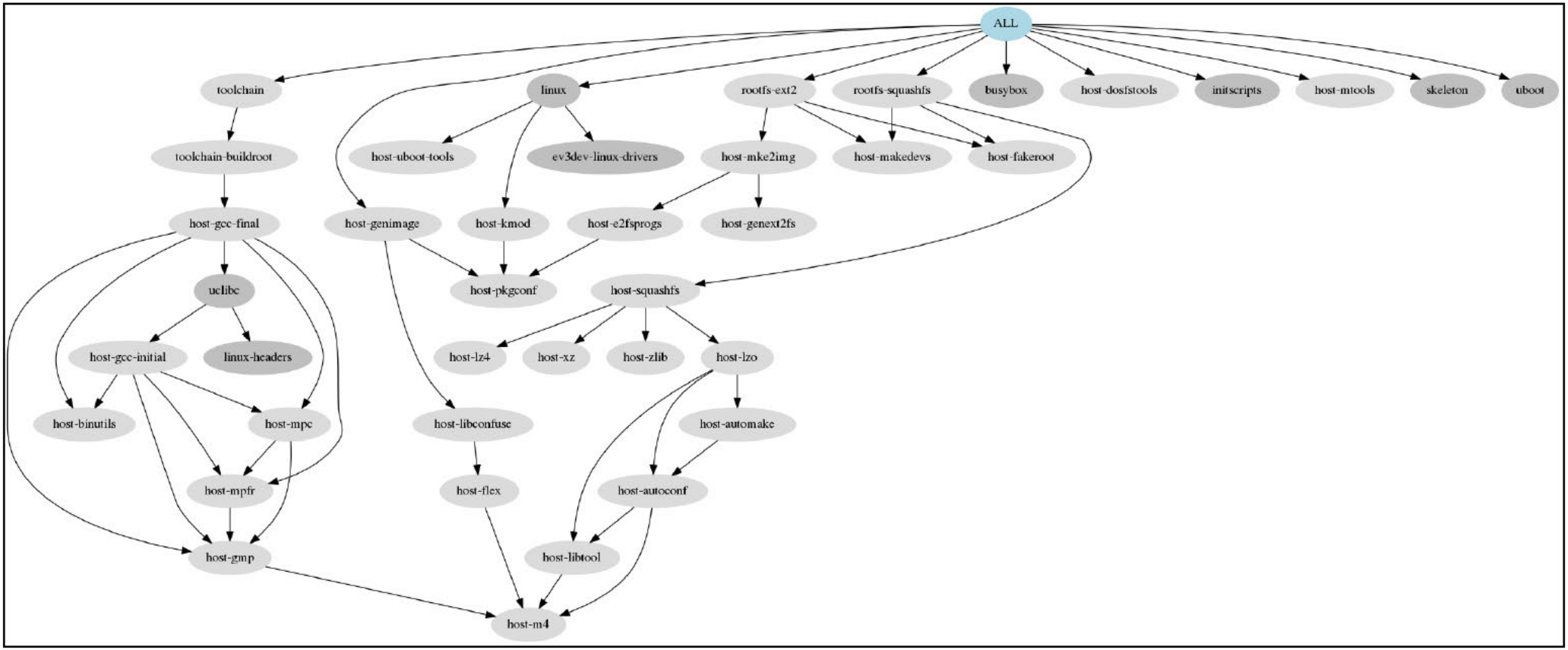
- 各ディレクトリ以下はパッケージ毎に分かれて
  該当ファイルが配置される

- Buildroot license GPLv2+ [GPLv2 or any later version]

MITSUBISHI ELECTRIC
Changes for the Better

パッケージの依存関係

- # apt install graphviz
- $ make graph-depends
  - output/graphs/graph-depends.pdf



MITSUBISHI ELECTRIC
Changes for the Better

グラフ

- 出力形式の指定
  - $ BR2_GRAPH_OUT=svg
    make graph-depends graph-build graph-size
  - $ man dot　(-Tオプション記載のフォーマット)
- 指定パッケージ起点
  - $ make busybox-graph-depends

もう少し踏み込んで開発したい

defconfigの作成
rootfsの構成変更
ユーザの追加
開発中パッケージの指定

MITSUBISHI ELECTRIC
Changes for the Better

defconfigの作成

- $ make savedefconfig
  - cp defconfig configs/<boardname>_defconfig
- $ make linux-update-defconfig
  - BR2_LINUX_KERNEL_CUSTOM_CONFIG_FILE に指定した.config からデフォルト値を取り除いたconfigを作成
- $ make busybox-update-config
  - BR2_PACKAGE_BUSYBOX_CONFIG
- $ make uclibc-update-config
  - BR2_UCLIBC_CONFIG

MITSUBISHI ELECTRIC
Changes for the Better

rootfsの構成変更

- Root filesystem overlays (BR2_ROOTFS_OVERLAY)
  - 指定したツリーをビルド後のターゲットrootfsにコピー
- Post-build scripts (BR2_ROOTFS_POST_BUILD_SCRIPT)
  - 全パッケージのインストール後に指定したスクリプトを実行
- Custom target skeleton (BR2_ROOTFS_SKELETON_CUSTOM)
  - 指定したツリーをビルド前のターゲットrootfsにコピー
- Post-fakeroot scripts (BR2_ROOTFS_POST_FAKEROOT_SCRIPT)
  - fakerootにて/devを作成した後に指定したスクリプトを実行
  - root権限にて実行される

ユーザの追加

- 特定フォーマットで記載したテキストファイルを BR2_ROOTFS_USERS_TABLESに指定する

- フォーマット（2行目までは説明のため記載）

```
USER UID  GROUP GID  Password HOME       SHELL      GROUPS      Comment
==== ====  ===== ====  ======== =========  ========== ==========  ==========
bob  -1   hoge  -1   P@ssword /home/bob  /bin/bash  alpha,bravo bob user
test 1080 wheel 1080 =        -          /bin/false  -           Test user
```

Buildrootにて自動決定

no password

明示的なhomeなし / が指定される

追加グループなし

開発中パッケージの指定

- 特定のパッケージは別のパスを指定したい場合。
  例えば、Linuxカーネルは特定のソースコードを使用したい。
  - 通常はBuildrootにより、ダウンロード、展開され、
    output/build/<package>-<version>に置かれる。

- local.mkにて<pkg>_OVERRIDE_SRCDIRを定義
  - $ cat local.mk
    LINUX_OVERRIDE_SRCDIR = /home/motai/linux/

- 注意事項
  - make clean の対象外となる。
  - ソースコードに修正を加えたら make <package>-rebuild all を
    実行する。

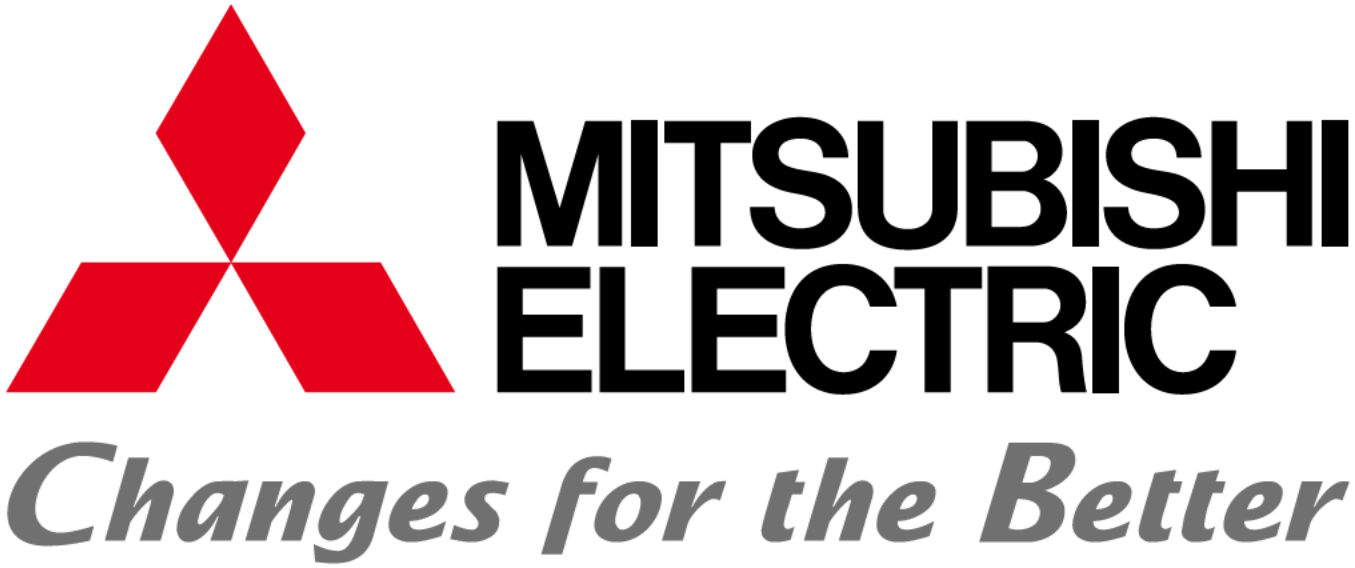
- 動作
  - rsync にて output/build/<package>-custom にコピー
  - その後、ビルド。

まとめ

- 組込みLinux向けユーザランド構築ツール『Buildroot』は
  ツールチェインからソースコードより生成する便利なツール

- 100超のリファレンスボードのdefconfigあり
  1500超のパッケージあり。
  設定変更も容易で、依存関係も図示。

- コツは必要だが、豊富なマニュアルとシンプルな作りで
  開発しやすい。


MITSUBISHI
ELECTRIC
Changes for the Better