An implemention of support for multiple run-time architectures in a packaging system perspective

TDT4900 Master Thesis, Technology Programme

Tollef Fog Heen (tfheen@idi.ntnu.no)

June 10, 2005

Acknowledgements

I would like to thank my advisor Anders Christensen for his good help and guidance with this thesis. He has been a lot of help on making the thesis a thesis rather than just a bunch of patches for free software. I would also like to thank Magni Onsøien for a lot of beating over the head with regards to references. I am in debt to the free software community for making this work possible, there is no way this could have happened without the ability to look at, change and distribute changed code.

Last but not least, my fiancée Karianne for her love, help and encouragement.

Trondheim, June 2005
Tollef Fog Heen

Abstract

Multiarch is a mechanism for packages supporting multiple architectures to be installed at the same time on the same machine in the same operating system. This paper shows a sample implementation of one way do do this on a UNIX-like system using the dpkg package manager. It shows the needed changes to both packages and the package manager itself.

Contents

Assignment text

The assignment is given as part of the project in the tenth semester of the Master of Technology programme at the Department of Computer and Information Science at the Faculty of Information Technology, Mathematics and Electrical Engineering, NTNU.

The assignment text is as follows:

> Up to now, most computer systems have been single architecture or had a very limited support for multiple architectures. A need for symmetric multiarchitecture is emerging in the Linux and Free Software world. The Linux kernel already supports executing binaries for different architectures, but support in packaging system and policy is still missing. Multiarch facilitates migration to new architectures due to its ability to reuse legacy software and take advantage of new features (such as 64 bit support) gradually.
>
> The assignment is to design and implement prototype symmetric multiarch support in a Linux packaging system.

The assignment is given by Anders Christensen, leader of the Technical Group at Department of Computer and Information Science.

Chapter 1

Introduction

Multiarch will be used as a term describing a system which has support for running binaries with different architectures, simultaneously and in the same file system hierarchy. Architecture is here the GNU system type. Typical values are "i386-linux", "x86_64-linux" and "i386-freebsd4.8". This is because both the software (and in some cases, the version of the software) as well as the hardware is defining the ABI which software on the system has to conform to.

Being able to do this is a goal because not all software exists or works correctly on all the architectures. Current examples are OpenOffice.org not working correctly on 64 bit architectures and the limited availability of proprietary software. A nice feature you get "for free" with multiarch is the ability to migrate from one platform to another without having a single day on which the whole system is switched.

This has, in one form or another, been supported in many operating systems for a long time. NetBSD/i386 has been able to run Linux/i386 binaries, FreeBSD likewise. This is usually accomplished through having a complete system installation in a chroot and the kernel doing more or less dirty tricks to make the programs work. A problem with most of those solutions is they force the administrator to maintain two different operating system installations, often very different in how they are maintained. The integration between them is poor and some software is installed twice. Examples of the latter are text-processing tools such as sed and awk.

This multiarch implementation will hopefully solve this problem. It will provide smooth integration and administer the system as a single system rather than multiple separate installations.

1.1 History

Multiarch systems have a long history, spanning back to at least the IBM S/360 model 30 which included support for emulating the IBM 1401 and

the IBM 7000 series[1] where emulation was used not only as an intermediate stage when moving from one architecture, but used as a permanent solution[2].

Later, emulation has been used in numerous research systems and production systems ranging from the PDP-11 emulation in Nanodata QM1[3] to more modern examples such as Apple's emulation of the m68k family of CPUs when they introduced the Power Macintosh and Transmeta's[4] Crusoe familiy of CPUs.

Even though multiarch has been used for a long time, it has not caught on except for some lone efforts. It is only lately that computers have begun having enough CPU power that one is able to afford the extra cost of supporting non-native architectures symmetrically. In this context, symmetrically means that any emulated architectures are first-class citizens and run at decent speed and with good integration into the rest of the system. This as opposed to an emulator where the emulated system runs inside a virtual machine and is only accessible through the host system. Earlier efforts have been on supporting a single legacy architecture to ease the transition cost. Multiarch is an effort where the term "native architecture" is no longer as meaningful since it uses the strength of the different ABIs provided rather than a "one size fits all" approach.

Another reason why this is happening now and not ten or twenty years ago is the growth of free software which runs on different hardware architectures. Multiarch unifies this. Proprietary vendors have usually had a bundle of hardware and software they wanted to sell and therefore had little incentive to investigate cross-architecture compatibility. Their efforts was at most limited to a biarch approach, usually with a 32 bit and a 64 bit set of libraries but without any support for alternative kernels (such as Linux). Those solutions also tend to have a much bigger focus on one of the architectures than the other. An example is the 64 bit support in Solaris which is "for selected application only" rather than a full and generic 64 bit development and runtime environment.

This also means the number of references in this thesis is fairly low and mostly limited to URLs to proposals and mailing list postings. Those are the media which the free software world uses for development, rather than the academic papers and publications.

1.2 Delimitations

The implementation here is done around dpkg, the package manager used in Debian and derived distributions such as Ubuntu. Due to time constraints, only the dpkg implementation is done, but a similar implementation could be done for other package managers such as RPM.

The file system implementation is an extension of the FHS[5] and could

therefore easily be reused for other Linux distributions. The BSDs, Sun Solaris and so on would possibly want to customise it a bit, but it could at least provide inspiration for a similar solution.

The implementation is done with UNIX and UNIX-like operating systems in mind. It might be possible to do something similar for VMS or Microsoft Windows, but that would most likely require heavy changes in both the design and implementation so they are not discussed further.

Building binaries for other architectures is not covered in this thesis. It is interesting to do that, but getting support for just running binaries from other architectures is needed first.

1.3 Intentended audience

This thesis is intended as a part of the debate surrounding how to do multiarch in Debian. The intended audience is Debian Developers and other interested people. An understanding of package relationships and the Debian policy[6] is recommended.

Chapter 2

Motivation and problem definition

The motivation for multiarch might not seem obvious, but on closer inspection it supports some very interesting use-cases:

- A user is on a 64 bit platform but needs a proprietary 32 bit plugin in his web browser to access sites using that plugin. An example is the flash plugin from Macromedia.
- A new architecture is introduced. This has a different ABI than the most prevalent architecture. However, the new architecture has the ability to run older binaries at acceptable speed.
- New features in binary formats or changed ABIs leave old libraries incompatible.

The first use-case is fairly common for AMD64 users today if they are using the platform in 64 bit mode rather than as in i386 legacy mode. Besides proprietary plugins, we have other proprietary applications such as a lot of games. Even in the cases where the source code is available the porting job might be big. Examples are OpenOffice.org and partimage.

We see the second case on both AMD64 and newer PowerPCs today. Those are able to run 64 bit code and older code at an acceptable speed. For the PowerPC a full migration does not make sense since it is usually a bit slower in 64 bit mode than in 32 bit mode. AMD64 is usually faster due to the greater number of registers compared to i386 and this offsets the speed penalty from increased pointer size. Multiarch allows the user to run his legacy applications until they are available for the new ABI. Developers are also able to test their applications on the new platform without worrying about breakage in the base system and only have to worry about the libraries they use directly.

The third use case is much more common than one would expect. For more or less each new release of the C++ compiler, the ABI is changed. This means all C++ libraries and any depending applications have to be recompiled. This is obviously a big pain and something to avoid whenever possible. Multiarch will make it a lot less painful and make it possible to transition gracefully without breaking user-compiled programs.

The problem is "how do we make this work"? This consists of multiple sub-problems:

- Where should the support for running the binaries be? In kernel- or user-space?
- File systems are hierarchical which means two files cannot exist in the same place. A policy on how to name files to avoid overlaps is needed.
- The packaging system will refuse to install packages for a foreign architecture today. This needs to be changed, but in such a way that only packages which will work are possible to install. This relationship is not bidirectional, since i386 packages are useful on amd64 but not vice versa.

As the target of our studies is dpkg which usually runs on Linux systems which use ELF[7] as the native object file format, ELF and Linux is what will be discussed further.

2.1 Starting an ELF object

A small introduction on how executing an ELF object (program) helps explain how it is possible to run binaries of different architectures, even without any kernel support. When the kernel is asked to execute an ELF object, it locates and loads the interpreter noted in the ELF header of said object. The interpreter then loads the program and any objects marked as NEEDED in the ELF header. Such external objects are typically libraries, but could be more or less anything the program needed. Then `_init` for each of the loaded objects (if available) is run and any constructor functions (prime examples are C++ global objects which are constructed automatically when the library is loaded). Finally, `main` itself is called and the program runs.

2.2 Kernel- or user-space support

The use cases which are interesting in this context already has support for emulation either in the kernel (as is the case for running i386 Linux binaries on a Linux/x86_64 kernel) or in user-space (as is the case for running i386 Linux binaries on a Linux/PowerPC system with qemu[8]).

The kernel support is implemented in a variety of different ways. An example is the support in Linux/x86_64 where the kernel only switches the CPU into 32 bit mode and executes the binary. Some other emulations use a `set_fs_altroot` kernel function which changes the visible root for the file system. `/emul/i386-linux` is the usual place for the alternate root.

There are multiple ways to implement multiarch in user-space. One way is to emulate a full system with no or very limited integration between the host and the guest system. This is a virtual machine approach where one usually sees the virtual machine in a terminal or similar on the host system. This approach is taken by VMware[9] and qemu for i386. The exact emulation can be done through emulation or virtualisation. As long as you are working on the same binary formats, a pure syscall interception route is also possible. In that case, a wrapper catches all the instructions and system calls and rewrites those to the native architecture.

2.3 File hierarchy policy

Once support to execute binaries is in place, a policy for where to place "foreign" binaries is needed. This is mostly an aesthetical issue, but important to ensure acceptance. Ugly solutions have a history of not gaining wide acceptance even though they solve the technical problem at hand. There are two dinstinct ways to organise the tree with small variations on exactly how to do it.

- Separate trees/chroot
- Merge the trees

2.3.1 Separated trees or chroot

Pros:

- Simpler, both in terms of layout and to construct.
- Easy handling of multiarch binaries
- "Pure" trees available to chroot into

Cons:

- Duplication of files
- Administrative overhead

Using a total segregation of the trees is a fairly simplistic approach. It shows clearly which files belong to which architecture. It also makes it easy

to change into a tree which is pure to that architecture rather than a mix of binaries for different architectures.

On the downside, having fully separate trees means more overhead as architecture-independent data is duplicated between the trees. Depending on how the packaging system is set up and works, it might be necessary to maintain two wholly separate systems with the administrative overhead that it causes.

An example of what fully separated trees could look like:

/
/lib
/usr
/usr/lib
/usr/share
/emul
/emul/i386−linux/lib
/emul/i386−linux/usr
/emul/i386−linux/usr/share
/emul/i386−linux/usr/lib

2.3.2 Merged trees

Pros:

- Integrated feeling
- Files are only stored once

Cons:

- More complex, needs policies which resolve file conflicts/overlaps
- No handling of multiarch binaries

Merging the trees and thereby making one multiarch installation rather than two (or more) installations side-by-side is a much cleaner approach. It makes the whole file system layout look better and more integrated. Architecture independent files can be placed in only one place rather than being duplicated in different parts of the tree. A single packaging system can handle all the installed packages. This all sums up to a much better look and feel of the system where multiarch support is actually integrated rather than "bolted on" afterwards.

A merged tree approach gives a few more possible issues than the separated trees approach. A clear policy on how to name architecture-dependent parts of the tree is needed to avoid file name overlaps between packages. The

ability to change into a separate tree or chroot and use that as an independent system goes away. Some tools, such as compilers, need to be available for multiple architectures. Any such tools need to have their name include an architecture-specific part, or be placed in a directory whose name includes an architecture-specific part.

As most binaries aren't that interesting to be provided in multiarch versions, the first approach will be chosen. Programs such as compilers are often installed multiple times on systems today and already provide good internal support for transforming the program name. Other programs which are part of the toolchain usually also has this support.

A solution where a merged tree with only the libraries (and any binaries which supports being installed with an architecture-specific pre/suffix) being multiarch is the solution which will be examined further. This approach gives us the biggest gain for the least effort. It gives us the ability to select applications of different architectures without imposing the overhead of providing all applications in co-installable versions.

An alternative would be to put all the binaries in paths with an architecture component and manage preferred versions of binaries with symlink farms. This would give the advantage of being able to run binaries for multiple architectures just by changing symlinks or calling the wanted version explicitly. It would also make exporting the tree of binaries to machines of other architectures a lot simpler. An example of a system which does this is Store[10].

An example of what a merged tree could look like is:

/
/lib
/lib/i386−linux
/lib/x86_64−linux
/usr
/usr/lib
/usr/lib/i386−linux
/usr/lib/x86_64−linux
/usr/share

2.4 Packaging system support

The packaging system needs to know about the file system policy so it can implement it correctly. The level of support will vary, depending on the abstraction level of the packaging system as well as the amount of semantics attached to the files. [11] Most have a dependency concept where one can say something like

Package: bazaar
Depends: libc6, diff, patch

This means that the `bazaar` package needs to have the packages `libc6`, `diff` and `patch` installed but does not say anything about which architectures those packages need to be. Since `libc6` is a library, it needs to be of the same architecture as the binaries in the `bazaar` package. Unless it uses diff and patch in a very peculiar way (by using dlopen and calling functions in them, effectively treating them as shared libraries), `bazaar` can use `diff` and `patch` packages of any architecture. They are text-processing tools which have text-based interfaces.

Our implementation needs to be able to express the architecture of dependent packages while preserving the current set of semantics. The last part is important so the current set of packages will continue to work correctly and not give an administrator any unpleasant surprises. If a dependency from one package to another suddenly meant a package of any architecture could fulfill it one would end up with a large amount of broken systems.

2.5 Related problems

Clashes in library and regular program names are not the only problems. Many libraries and binaries have embedded search paths. They have search paths for plugins and for data files. They have search paths for caches, for icons and so on. Making sure that any paths which contain architecture dependent files point to the right places and the same for architecture independent files is important and probably something which will cause some extra work and disruption when implementing multiarch. A number of tools will certainly need some changes.

Some examples of tools which need to be changed are:

- binutils[12] and gcc[13] as they have a default search path which needs to be updated.
- pkg-config[14], to look in multiple directories for definitions on the architecture-specific data files it uses.
- libtool[15], autoconf[16], automake[17] and other tools which know the default search path of the compiler and looks for installed libraries.

Chapter 3

Analysis of existing solutions

The existing solutions have a few common shortcomings:

- Limited number of architectures
- No way to handle multiple architectures with same pointer size
- Little or no real support for multiarch, just hacked-on solutions

There are some existing solutions for handling multiarch and there has been a consensus forming on how to solve it on the file system hierarchy level, but nobody has so far been willing to take the first step and make their system multiarch-compliant. The implementations which exist are limited and usually only supports two architectures, not an unlimited number.

In most cases, support for two architectures will be enough. A very common case will be support for i386 binaries on an x86_64 machine or support for powerpc or sparc binaries on their equivalent 64 bit systems. This can be solved adequately with a solution which just provides support for two architectures, for instance the file system layout most Linux distributions have so far with `/usr/lib` and `/usr/lib64` (or `/usr/lib` and `/usr/lib32` where the "native" architecture is 64 bit).

The lib/lib64 solution doesn't work well once you want to support multiple 64 or 32 bit architectures. An example is NetBSD[18] and FreeBSD's[19] support for running Linux binaries. Another example would be the moment ia64[20] get support for running x86_64 binaries where you'd get a collision between the architectures in what should go in /usr/lib (since that is a 64 bit directory on ia64).

One can construct more contrived examples, such as running i386 and amd64 binaries for Linux, FreeBSD and NetBSD, on a NetBSD system. One would then need six different library directories. Once you have such large systems, a generalised solution is obviously needed.

The existing solutions have been more thoroughly described in [11]. This is recommended reading to understand some of the scalability problems in the software distribution systems of today.

3.1 The "File Hierarchy Standard"

The current FHS[5] only includes support for the `/lib`*qual* form with *qual* being 32 or 64 depending on the pointer size. There are proposals for adding other structures with the most common being `/lib/`*architecture* [21] [22]. Another proposal was using the already existing support for a `/usr/`*architecture* which is well-supported in the toolchain, but usually used for cross-compilation, not for binaries of a "foreign" architecture.

3.2 Current multiarch support in packaging tools

The support for multiarch in the current set of packaging tools is mostly missing with a few exceptions. One such exception is RPM, which supports installing packages for multiple architectures.

By inspection of Fedora Core 3 RPMs, it does not seems to special-case cross-architecture dependencies. Most dependencies on binary files are file-based dependencies rather than dependencies on packages. The dependencies are just on "64 bit" without any further specification, so it is prone to confusing ia64-linux and x86_64-linux, for instance. File dependencies by themselves are hard to get right since you need to make sure only one package in the distribution provides the file depended on.

Chapter 4

Implementation of a prototype

Appendix A contains a set of patches used for implemention of a prototype of the suggested solution. It is both an implementation of the file system layout as suggested by the proposals referenced in section 3.1 as well as a the needed changes to dpkg.

4.1 Toolchain changes

For packages to build correctly, the C library and the toolchain need to become aware of the new directory structure. The following changes were implemented:

- common
  - Move files into trees with the architecture name in the path.
  - Split out any arch-independent files into its own package.
- glibc
  - Add support for looking in `/lib/`*architecture* and `/usr/lib/`*architecture* to the run-time linker.

  The patches for glibc are available in Appendix A.1.
- binutils
  - Add support to look in `/lib/`*architecture* and `/usr/lib/`*architecture* by default to the linker.

  The patches for binutils are available in Appendix A.3.
- gcc

  - Add support to look in `/lib/`*architecture* and `/usr/lib/`*architecture* to the default set of linker options.
  - Add support to look in `/usr/include/`*architecture* for the C preprocessor.

  The patches for gcc (version 3.4) are available in Appendix A.2.

- pkg-config
  - Add `/usr/lib/`*architecture*`/pkgconfig` to the default search path for .pc files.

  The patches for pkg-config are available in Appendix A.5.

4.2 Per-package changes

The last package, pkg-config, does not strictly need to be changed in a minimal implementation. However, it is increasingly used as part of the build chain and therefore included as an example of an auxiliary tool which will need changes. Other such tools are autoconf, automake and libtool. We can avoid changing those for the purpose of this minimal implementation by taking care when passing parameters to the configure script.

Once the changes to the toolchain are in place, the following changes to a few packages are needed:

- Split the packages into architecture-independent and arch-dependent parts to avoid file overlap between packages. Any overlaps will cause the packages not to be coinstallable as the package manager has no way to know which file is the correct one.
- Making sure any architecture-dependent files either are in directories with the architecture as part of their name or that they include the architecture name themselves. Most notably are proper libraries, but other examples are plugins or auxiliary data.
- Packages using autoconf, automake and libtool should set the libdir variable to point to the right directory so any .la files installed have the correct contents. The .la files are used by libtool as part of the link process to ensure the correct set of libraries are linked with.

4.2.1 Example

As an example, the `libogg` source package will be changed from a regular, non-multiarched package to a multiarched one. The "common" package will be called libogg-common. The "common" development package will be called libogg-dev-common. The actual naming here is just a convention,

there is nothing in the packaging system which requires it to be named something in particular.

The `libogg0` package needs to depend on `libogg0-common` due to Debian policy constraints with regards to the shipping of a copyright.

Then all the architecture-independent files have to move to the "-common" package. Since the package uses debhelper, simply adjusting the .install file suffices.

The build system needs some minor changes too. As the package uses autoconf and automake the change needed is limited to the call to configure which has "`--libdir`" and "`--includedir`" parameters added.

The patches for this are in Appendix A.6.

4.3 Package manager changes

Dpkg itself needs fairly minimal fixes:

- Handle the "Multi-Arch" field so a package can be installed multiple times for different architectures. This requires changes to dpkg itself as well as dpkg-gencontrol to not strip out the field when building a package.
- Support installing packages for "foreign" architectures.
- Handle the case of a symlink pointing to a directory and being owned by different packages as a special case. Dpkg should allow it an reference-count the link rather than the default behaviour which is to exit with an error because of the overlapping file. (This is required by Debian's policy of all packages being required to have a `/usr/share/doc/`*packagename* which shall be a directory or a symlink to a package built by the same source package. This is because the package has to ship a license, which is usually contained in `/usr/share/doc/`*packagename*`/copyright`)

This is a preliminary prototype and in no way intended for general usage. It shows the point and demonstrates that multiarch this way is possible and works.

A filtered set of patches are in Appendix A.4. All auto-generated files such as those made by automake, autoconf and libtool are removed from the set of patches due to space and readability reasons. The dpkg patches in that set build upon work done previously by Hugo Mills as part of an earlier push for multiarch. His work in turn was based up specifications and ideas as published in [22] and [23] by this author.

While the implementation was written, the upstream author of dpkg has continued development and provided further changes and some extensions on

the idea into “Feature dependencies”[24]. This will supersede the prototype implementation.

Chapter 5

Evaluation of prototype

5.1 Testing

Testing clearly shows that the file system level implementation is solid. It works well so far without any trouble. As the implementation tested is a small-scale implementation without any programs doing complex operations such as heavy reliance on dlopen(3) or plugins. Various programs will most likely have issues when those files are relocated, but those are most likely fairly easily fixed.

The dpkg prototype implementation is not complete and has a number of bugs which would need to be fixed before being generally usable. However, those are not visible in the case that one would use it in the regular, non-multiarch fashion.

5.2 Adoption

Several of the changes have already been adopted by distributions as well as upstream.

- glibc: The changes to ld-linux.so are already in both Debian 3.1 "Sarge" and Ubuntu 5.04 "Hoary". The changes which split the package into different parts will have to wait until the previous version of the distribution has support for handling the "Multi-Arch" field in the control file due to the policy of said distributions.
- binutils: Not yet adopted, but will most likely go in post-"Sarge" and for Ubuntu 5.10 "Breezy".
- gcc: Not yet adopted, but will most likely go in post-"Sarge" and for Ubuntu 5.10 "Breezy".
- pkg-config: As the author is the current maintainer and multiarch support could be added without any additonal breakage, it was added

a couple of versions back. Debian 3.1 “Sarge” has it and Ubuntu 5.10 “Breezy” will also have it.

As the changes to dpkg are in no way done, stable or tested yet the package splitting changes have not been adopted by anybody. They will simply not make any sense until it is actually possible to have co-installable packages.

5.3 Adoption by upstream

As most of the patches are fairly new and the work has been to push them into a distribution rather than all the way to upstream where it would then trickle downstream to distributions again, there is no adoption of multiarch by upstream yet.

The prospects look good, however. There is a fair amount of interest in multiarch from various distributors as well as from standard bodies such as the Linux Standards Base and the FHS. None have moved forward and actually implemented anything yet as they are waiting for Debian to show that it is actually possible to make the system work.

Chapter 6

Design of an production-ready implementation

A design along the lines as suggested in [24] would require support for a number of new features. A feature is here loosely defined as "something the program should do with a certain amount of size and complexity". Features can be overlapping, but they should generally be orthogonal. Dpkg needs some changes to support the FeatureDependencies (from [24]) specification.

6.1 Dpkg changes

- dpkg-provided packages.
- Ability to extract features "smartly" by tools such as dpkg-shlibdeps.
- Extending the syntax of control files to be able to say "this package with this feature". Examples would be "Depends: `libfoo:i386`"
- Continued support for sharing of symlinks with common targets between packages.

6.1.1 dpkg-provided packages

Impact: small

Current packages do not have an explicit dependency on the ABI they need. It works out well due to each system having one and only one architecture and the ABI thereby being implicitly defined from the packages depended on and the architecture of the package. The architecture is an implicit and second class citizen, dependency-wise.

To solve this and make the architecture a first-class citizen, the ABI should be specified in the Depends field. Since providing the system ABI doesn't belong naturally to any package, it should be a non-existing, dpkg-provided package. The ABIs should be determined at run-time as they can change based on the installation and removal of emulator packages.

6.1.2 Smart extraction of features

Impact: medium

As part of most packages's build process, dpkg-shlibdeps is run. It looks at the programs in the package and what libraries those link to. Then it checks a file which says at what point those libraries last changed their ABI and writes appropriate dependency information into a "substvars" file. Dpkg-shlibdeps needs to be extended to include the full ABI information in the substvars file.

In addition, it must be possible to specify a list of features by hand in the regular "debian/control" file.

6.1.3 Features in Depends fields

Impact: big

Currently, the only information one can specify in a Depends field is a package with an optional versioned relation (less than, less than or equal, equal, equal than or bigger or bigger than). This must be changed to a syntax which allows specification of features.

It is here important to distinguish between the syntax of debian/control in the source package and DEBIAN/control in the binary package. (See [25] and [26] for more information on the difference between those two files.) The former must be easy to both read and write for humans. The latter must be parseable for computers (but still be possible to read for humans). The proposal here lists two different syntaxes for specifying features depending on which file is the relevant one

6.1.3.1 Source package

The syntax of the human-readable debian/control should be:

Depends: **package**:<feature1 feature2>

Feature can (for instance) be "gnome2" or "i386-linux".

6.1.3.2 Binary package

The syntax in the binary package's DEBIAN/control should be:

Depends: **package**:feature1, **package**:feature2

This is because generation of extra parts of the Depends line is a lot easier when split up rather than inside a set of brackets.

An example is the source debian/control file

Package: bazaar
Depends: libc6, diff:<any>, patch:<any>

which would become the binary DEBIAN/control file

Package: bazaar
Depends: system:i386, libc6:i386, diff, patch

The parser in dpkg needs to be adjusted to interpret this syntax correctly, both in the source and binary packages.

6.1.4 Sharing of symlinks with common target

Impact: small

At the moment, any overlap in files between two packages will cause dpkg to error out on installation with an error message. Debian policy requires the copyright information for a package to be available as `/usr/share/doc/`*package*`/copyright`. This will, naturally, cause trouble with multiarch and dpkg will therefore be changed so a symlink with a common name and target is reference-counted and can be shared between different packages. This is similar to how directories are handled.

The code for this support can be copied off the prototype, but it needs a bit more work. It currently just removes the symlink on remove, which is obviously wrong and it should rather have some reference-counting code in there. The already existing conflict handling should kick in if a new package with a symlink pointing to somewhere else is installed.

6.2 Other affected systems

Dpkg isn't the only tool which interprets and uses package relationships. As both the syntax and the semantics of the control file fields change all of those tools might need updating. No exhaustive list exists, but the most prominent ones are:

- dak, also known as katie. Those are the scripts used for managing the Debian archive.
- linda, lintian. Those packages are tools which are used to "lint" packages for common errors and mistakes.

- apt, dselect, aptitude, synaptic. All package managers handle dependencies between different packages. They need to show the features provided by different packages and which of those causes conflicts.

Chapter 7

Conclusion

As this thesis shows, multiarch is a very real possibility. Most of the building blocks are already in place, but they still need to be assembled and tested as a whole. One of the significant missing pieces is support in the packaging system itself. A proof-of-concept prototype has been developed and shown to work.

The toolchain changes are already in some Linux distributions such as Debian and Ubuntu, and in the pipeline for others where it will be part of the next stable release. If the implementation is shown to be beneficial, it will most likely be adopted by other distributions and possibly other operating system vendors as well.

The changes in dpkg are still immature and would need a fair amount of work to become more stable and mature. The dpkg maintainer (who is also the author of the "FeatureDepends" specification) has expressed interest in how to best make dpkg multiarch-capable.

Multiarch enables users to run the software they want, in the version they want on their machines. It enables software developers to more easily develop and test for other platforms.

Chapter 8

Further work

Further work in this case constists of two parts: Going deeper into the work and finding bugs and fixing applications to work and understand multiarch correctly. The other part is alternative approaches.

8.1 Fixing bugs and applications

Some of the applications which need to be fixed are

- libtool. Needs to know about the new paths and in many cases it would be useful if it searched for libraries properly instead of relying on hard-coded paths. Without this fix, moving a library another library needs to link means applications using the latter library will fail to build.
- pango (and other libraries using configuration files). Pango is a library implementing Unicode support for GTK+ applications. It has support for plugins implemented as shared objects and loaded at run-time. Those are found using a configuration file which lists all the plugins and their properties. The problem is that the configuration file actually lists the full path to the plugins. As the configuration file has no architecture component, this will have to be solved by not having the full path in the configuration file and using a search path, or through some other mean.

Pango is included here not because it is unique, but rather because it shows an example of behaviour which is problematic when going from a single architecture to multiple architectures.

8.2 Alternative implementations

Changing the file system layout is only one way to solve the multiarch problem. One other possibility is handling it all in kernel-space so Linux/x86_-

64 applications accessing `/usr/lib` will see a different directory from Linux/i386 applications. This is certainly possible, but requires changes in the kernel code and can be confusing for the system administrator. An interesting challenge would be to decide how the packaging system would make sure to access the correct library directory.

References

[1] S. G. Tucker. Emulation of large systems. *Commun. ACM*, 8(12):753–761, 1965.

[2] Mark Smotherman. A brief history of microprogramming. http://www.cs.clemson.edu/ mark/uprog.html, 1999.

[3] J. C. Demco and T. A. Marsland. An insight into PDP-11 emulation. In *MICRO 9: Proceedings of the 9th annual workshop on Microprogramming*, pages 20–26, New York, NY, USA, 1976. ACM Press.

[4] Transmeta site. http://www.transmeta.com/.

[5] Filesystem hierarchy standard. http://www.pathname.com/fhs/pub/fhs-2.3.html, 01 2004.

[6] The debian-policy mailing list. Debian policy manual. http://www.debian.org/doc/debian-policy/.

[7] Tool Interface Standard (TIS) Executable and Linking Format (ELF) Specification. http://www.x86.org/ftp/manuals/tools/elf.pdf, 05 1995.

[8] Qemu web site. http://fabrice.bellard.free.fr/qemu/.

[9] Vmware web site. http://www.vmware.com/.

[10] Anders Christensen and Tor Egge. Store - a system for handling third-party applications in a heterogeneous computer environment. In Jacky Estublier, editor, *SCM*, volume 1005 of *Lecture Notes in Computer Science*, pages 263–276. Springer, 1995.

[11] Tollef Fog Heen. Scalable software distribution systems. Technical report, Norwegian University of Science and Technology, Department of Computer and Information Science, 2004. http://err.no/ntnu/project/report.pdf.

[12] binutils web site.
http://www.gnu.org/software/binutils/.

[13] GCC web site.
http://gcc.gnu.org/.

[14] pkgconfig web site.
http://pkgconfig.freedesktop.org.

[15] libtool web site.
http://www.gnu.org/software/libtool/libtool.

[16] autoconf web site.
http://www.gnu.org/software/autoconf/.

[17] automake web site.
http://www.gnu.org/software/automake/.

[18] NetBSD. http://www.netbsd.org/ports/.

[19] FreeBSD web site. http://www.freebsd.org/.

[20] Itanium/ia64 web site at Intel.
http://developer.intel.com/design/itanium/family/.

[21] Matt Taggart. Multiple architecture problem and proposed solution. http://www.linuxbase.org/ taggart/multiarch.html, 06 2004.

[22] Tollef Fog Heen. How to handle multiarch on x86-64 (and other platforms). http://err.no/debian/amd64-multiarch-3, 05 2004.

[23] Tollef Fog Heen. How to handle multiarch on x86-64 (and other platforms). http://err.no/debian/amd64-multiarch-2, 02 2004.

[24] Scott James Remnant. Dpkg web site: Feature dependencies.
http://www.dpkg.org/FeatureDependencies.

[25] The debian-policy mailing list. Debian policy manual; binary package control files – debian/control.
http://www.debian.org/doc/debian-policy/ch-controlfields.html#s-binarycontrolfiles.

[26] The debian-policy mailing list. Debian policy manual; source package control files – debian/control.
http://www.debian.org/doc/debian-policy/ch-controlfields.html#s-sourcecontrolfiles.

Appendix A

Patches for prototype implementation

Those are the patches for the prototype implementation. The patches have been cleaned of auto-generated files due to size concerns (with all auto-generated files, the patches would have been a total of more than a thousand pages). The patch logs from the revision control system and any commit messages have also been removed. They can all be browsed online at `http://arch.err.no`.

A.1 Glibc

```
--- orig/debian/changelog
+++ mod/debian/changelog
@@ -1,3 +1,35 @@
+glibc (2.3.2.ds1-20multiarch5) multiarch; urgency=low
+
+  * Actually make the symlinks in the -dev package, ↩
   not the normal one.
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Wed, 23 Feb ↩
   2005 14:52:40 +0100
+
+glibc (2.3.2.ds1-20multiarch4) multiarch; urgency=low
+
+  * If DEB_BUILD_GNU_TYPE and $(GCC_ARCH) differ, make↩
    symlinks in
+    /usr/include as well.
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Thu, 10 Feb ↩
   2005 14:33:02 +0100
+
+glibc (2.3.2.ds1-20multiarch3) multiarch; urgency=low
```

```
+
+  * Make compat symlinks for crt1.o and friends.
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Wed,  9 Feb ↩
    2005 10:04:00 +0100
+
+glibc (2.3.2.ds1-20multiarch2) multiarch; urgency=low
+
+  * Bump version number and upload .orig.tar.gz
+  * Remove ccache from debian/rules to facilitate ↩
    autobuilding.
+
+ -- Tollef Fog Heen <tfheen@debian.org>  Fri,  4 Feb ↩
    2005 16:03:05 +0100
+
+glibc (2.3.2.ds1-20multiarch0) multiarch; urgency=low
+
+  * Multiarch build
+
+ -- Tollef Fog Heen <tfheen@debian.org>  Mon, 24 Jan ↩
    2005 12:47:49 +0100
+
 glibc (2.3.2.ds1-20) unstable; urgency=high

   * GOTO Masanori <gotom@debian.org>

--- orig/debian/control.in/libc
+++ mod/debian/control.in/libc
@@ -3,11 +3,12 @@
 Section: base
 Priority: required
 Provides: ${locale:Depends}
+Multi-Arch: yes
+Depends: @libc@-common (= ${Source-Version})
 Description: GNU C Library: Shared libraries and ↩
    Timezone data
  Contains the standard libraries that are used by ↩
     nearly all programs on
  the system. This package includes shared versions of ↩
     the standard C library
  and the standard math library, as well as many others↩
     .
- Timezone data is also included.

 Package: @libc@-dev
 Architecture: @archs@
@@ -15,10 +16,21 @@
 Priority: standard
 Depends: @libc@ (= ${Source-Version})
```

```
 Recommends: gcc | c-compiler
+Multi-Arch: yes
 Description: GNU C Library: Development Libraries and ↩
    Header Files
  Contains the symlinks, headers, and object files ↩
     needed to compile
  and link programs which use the standard C library.

+Package: @libc@-dev-common
+Architecture: @archs@
+Section: libdevel
+Priority: standard
+Depends: @libc@ (= ${Source-Version})
+Recommends: c-compiler
+Description: GNU C Library: Development common files
+ Contains the support tools needed by the development ↩
   package for the
+ GNU C Library.
+
 Package: @libc@-dbg
 Architecture: @archs@
 Section: libdevel
@@ -56,6 +68,16 @@
  boot floppies. If you are not making your own set of ↩
     Debian boot floppies
  using the 'boot-floppies' package, you probably don't↩
      need this package.

+Package: @libc@-common
+Architecture: @archs@
+Section: base
+Priority: required
+Replaces: @libc@ (<< 2.3.2.ds1-20multiarch0)
+Conflicts: @libc@ (<< 2.3.2.ds1-20multiarch0)
+Description: GNU C Library: Common files
+ Contains common files for the standard libraries that↩
    are used by
+ nearly all programs on the system.  Timezone data is ↩
   also included.
+
 Package: @libc@-udeb
 XC-Package-Type: udeb
 Architecture: @archs@


--- orig/debian/debhelper.in/libc-dbg.install
+++ mod/debian/debhelper.in/libc-dbg.install
@@ -1 +1 @@
-TMPDIR/lib/*.so* usr/lib/debug
```

```
+TMPDIR/lib/*.so* usr/lib/DEB_BUILD_GNU_TYPE/debug


--- orig/debian/debhelper.in/libc-dev.install
+++ mod/debian/debhelper.in/libc-dev.install
@@ -1,26 +1,24 @@
-debian/tmp-libc/usr/bin/gencat usr/bin
-debian/tmp-libc/usr/bin/mtrace usr/bin
-debian/tmp-libc/usr/bin/rpcgen usr/bin

-debian/tmp-libc/usr/lib/libanl.a usr/lib
-debian/tmp-libc/usr/lib/libBrokenLocale.a usr/lib
-debian/tmp-libc/usr/lib/libbsd-compat.a usr/lib
-debian/tmp-libc/usr/lib/libc.a usr/lib
-debian/tmp-libc/usr/lib/libc_nonshared.a usr/lib
-debian/tmp-libc/usr/lib/libcrypt.a usr/lib
-debian/tmp-libc/usr/lib/libdl.a usr/lib
-debian/tmp-libc/usr/lib/libg.a usr/lib
-debian/tmp-libc/usr/lib/libieee.a usr/lib
-debian/tmp-libc/usr/lib/libm.a usr/lib
-debian/tmp-libc/usr/lib/libmcheck.a usr/lib
-debian/tmp-libc/usr/lib/libnsl.a usr/lib
-debian/tmp-libc/usr/lib/libpthread.a usr/lib
-debian/tmp-libc/usr/lib/libpthread_nonshared.a usr/lib
-debian/tmp-libc/usr/lib/libresolv.a usr/lib
-debian/tmp-libc/usr/lib/librpcsvc.a usr/lib
-debian/tmp-libc/usr/lib/librt.a usr/lib
-debian/tmp-libc/usr/lib/libutil.a usr/lib
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/libanl.a ↩
    usr/lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/↩
    libBrokenLocale.a usr/lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/↩
    libbsd-compat.a usr/lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/libc.a usr/↩
    lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/↩
    libc_nonshared.a usr/lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/libcrypt.a ↩
    usr/lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/libdl.a usr↩
    /lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/libg.a usr/↩
    lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/libieee.a ↩
    usr/lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/libm.a usr/↩
    lib/DEB_BUILD_GNU_TYPE
```

```
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/libmcheck.a↩
     usr/lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/libnsl.a ↩
    usr/lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/libpthread.↩
    a usr/lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/↩
    libpthread_nonshared.a /usr/lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/libresolv.a↩
     usr/lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/librpcsvc.a↩
     usr/lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/librt.a usr↩
    /lib/DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/libutil.a ↩
    usr/lib/DEB_BUILD_GNU_TYPE

-debian/tmp-libc/usr/lib/*.o usr/lib
-debian/tmp-libc/usr/lib/*.so usr/lib
-debian/tmp-libc/usr/include/* usr/include
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/*.o usr/lib↩
    /DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/*.so usr/↩
    lib/DEB_BUILD_GNU_TYPE
+
+debian/tmp-libc/usr/include/DEB_BUILD_GNU_TYPE/* usr/↩
    include/DEB_BUILD_GNU_TYPE


--- orig/debian/debhelper.in/libc.docs
+++ mod/debian/debhelper.in/libc.docs
@@ -1,10 +0,0 @@
-log-test-*
-DEB_SRCDIR/BUGS
-DEB_SRCDIR/FAQ
-DEB_SRCDIR/INTERFACE
-DEB_SRCDIR/NEWS
-DEB_SRCDIR/NOTES
-DEB_SRCDIR/PROJECTS
-DEB_SRCDIR/README
-DEB_SRCDIR/hesiod/README.hesiod
-debian/TODO


--- orig/debian/debhelper.in/libc.install
+++ mod/debian/debhelper.in/libc.install
@@ -1,23 +1,3 @@
-debian/tmp-libc/lib/*.so* lib
-debian/tmp-libc/usr/lib/gconv/*.so usr/lib/gconv
```

```
-debian/tmp-libc/usr/lib/gconv/gconv-modules usr/lib/↩
   gconv
-debian/tmp-libc/usr/share/zoneinfo/* usr/share/↩
   zoneinfo
-debian/tmp-libc/usr/bin/iconv usr/bin
-debian/tmp-libc/usr/bin/locale usr/bin
-debian/tmp-libc/usr/bin/localedef usr/bin
-debian/tmp-libc/usr/bin/getent usr/bin
-debian/tmp-libc/usr/bin/getconf usr/bin
-debian/tmp-libc/usr/bin/catchsegv usr/bin
-debian/tmp-libc/usr/bin/tzselect usr/bin
-debian/tmp-libc/usr/bin/ldd* usr/bin
-debian/tmp-libc/usr/sbin/zdump usr/bin
-debian/tmp-libc/usr/sbin/rpcinfo usr/bin
-
-debian/tmp-libc/usr/sbin/zic usr/sbin
-debian/tmp-libc/usr/sbin/iconvconfig usr/sbin
-
-debian/tmp-libc/sbin/ldconfig sbin
-
-debian/tmp-libc/usr/lib/pt_chown usr/lib
-
-debian/local/usr_sbin/tzconfig usr/sbin
+debian/tmp-libc/lib/DEB_BUILD_GNU_TYPE/*.so* lib/↩
   DEB_BUILD_GNU_TYPE
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/gconv/*.so ↩
   usr/lib/DEB_BUILD_GNU_TYPE/gconv
+debian/tmp-libc/usr/lib/DEB_BUILD_GNU_TYPE/gconv/↩
   gconv-modules usr/lib/DEB_BUILD_GNU_TYPE/gconv


--- orig/debian/debhelper.in/libc.links
+++ mod/debian/debhelper.in/libc.links
@@ -1 +0,0 @@
-etc/localtime usr/share/zoneinfo/localtime

--- orig/debian/rules
+++ mod/debian/rules
@@ -35,6 +35,7 @@
 # The minimum package version with which these ↩
    packages are compatible.
 shlib_dep_ver = 2.3.2.ds1-4
 shlib_dep = $(libc) (>= $(shlib_dep_ver))
+export DH_VERBOSE=1

 # Beyond here you shouldn't need to customise anything↩
    :

@@ -47,6 +48,7 @@
```

```
 DEB_BUILD_GNU_CPU     ?= $(shell dpkg-architecture ↩
    -qDEB_BUILD_GNU_CPU)
 DEB_BUILD_GNU_TYPE    ?= $(shell dpkg-architecture ↩
    -qDEB_BUILD_GNU_TYPE)
 DEB_BUILD_GNU_SYSTEM  ?= $(shell dpkg-architecture ↩
    -qDEB_BUILD_GNU_SYSTEM)
+GCC_ARCH              ?= $(shell gcc -dumpmachine)

 DEB_HOST_GNU_CPU_ALT  ?=
 DEB_HOST_GNU_TYPE_ALT ?=
@@ -65,7 +67,7 @@
 KERNEL_HOST_CPU := $(subst powerpc,ppc,↩
    $(DEB_HOST_GNU_CPU))

 # How many makes to run at once?
-NJOBS = 1
+NJOBS = 2

 # Default setup
 GLIBC_PASSES ?= libc
@@ -73,15 +75,17 @@
 prefix=/usr
 bindir=$(prefix)/bin
 datadir=$(prefix)/share
-includedir=$(prefix)/include
+includedir=$(prefix)/include/$(DEB_BUILD_GNU_TYPE)
 infodir=$(prefix)/share/info
-libdir=$(prefix)/lib
+libdir=$(prefix)/lib/$(DEB_BUILD_GNU_TYPE)
 docdir=$(prefix)/share/doc
 mandir=$(prefix)/share/man
 sbindir=$(prefix)/sbin
+slibdir=/lib/$(DEB_BUILD_GNU_TYPE)
 libexecdir=$(prefix)/lib
+extra_libdir=/lib:$(prefix)/lib

-BUILD_CC = gcc-3.3
+BUILD_CC = gcc

 # Set CC for cross-compiling
 ifneq ($(DEB_HOST_ARCH),$(DEB_BUILD_ARCH))
@@ -112,7 +116,7 @@
 # Which build pass are we on?
 curpass = $(filter-out %_,$(subst _,_ ,$@))

-DEB_ARCH_REGULAR_PACKAGES = $(libc) $(libc)-dev $(libc↩
   )-dbg $(libc)-prof $(libc)-pic
+DEB_ARCH_REGULAR_PACKAGES = $(libc) $(libc)-dev $(libc↩
   )-dbg $(libc)-prof $(libc)-pic $(libc)-common $(libc↩
```

```
   )-dev-common
 DEB_INDEP_REGULAR_PACKAGES = glibc-doc locales
 DEB_UDEB_PACKAGES = $(libc)-udeb libnss-dns-udeb ↩
    libnss-files-udeb



--- orig/debian/rules.d/build.mk
+++ mod/debian/rules.d/build.mk
@@ -23,12 +23,18 @@
         echo "mandir = $(mandir)"       >> ↩
            $(DEB_BUILDDIR)/configparms
         echo "infodir = $(infodir)"     >> ↩
            $(DEB_BUILDDIR)/configparms
         echo "libexecdir = $(libexecdir)" >> ↩
            $(DEB_BUILDDIR)/configparms
+        echo "libdir = $(libdir)"      >> ↩
    $(DEB_BUILDDIR)/configparms
+        echo "includedir = $(includedir)" >> ↩
    $(DEB_BUILDDIR)/configparms
         echo "LIBGD = no"               >> ↩
            $(DEB_BUILDDIR)/configparms
         echo "sysconfdir = /etc"        >> ↩
            $(DEB_BUILDDIR)/configparms
         echo "rootsbindir = /sbin"      >> ↩
            $(DEB_BUILDDIR)/configparms
 ifneq ($(call xx,slibdir),)
         echo "slibdir = $(call xx,slibdir)" >> ↩
            $(DEB_BUILDDIR)/configparms
 endif
+ifneq ($(call xx,extra_libdir),)
+        echo "extra_libdir = $(call xx,extra_libdir)" ↩
    >> $(DEB_BUILDDIR)/configparms
+
+endif

         # Prevent autoconf from running unexpectedly by↩
             setting it to false.
         # Also explicitly pass CC down - this is needed↩
             to get -m64 on


--- orig/debian/rules.d/debhelper.mk
+++ mod/debian/rules.d/debhelper.mk
@@ -17,7 +17,7 @@
         install --mode=0644 build-tree/$(DEB_HOST_ARCH)↩
            -libc/libresolv.map debian/$(libc)-pic/usr/↩
            lib/libresolv_pic.map
```

```
 # Some per-package extra files to install.
-define $(libc)_extra_debhelper_pkg_install
+define $(libc)-common_extra_debhelper_pkg_install
        install --mode=0644 $(DEB_SRCDIR)/ChangeLog ↩
           debian/$(curpass)/usr/share/doc/$(curpass)/↩
           changelog
        install --mode=0644 $(DEB_SRCDIR)/linuxthreads/↩
           README debian/$(curpass)/usr/share/doc/↩
           $(curpass)/README.linuxthreads
        install --mode=0644 $(DEB_SRCDIR)/linuxthreads/↩
           ChangeLog debian/$(curpass)/usr/share/doc/↩
           $(curpass)/ChangeLog.linuxthreads
@@ -32,6 +32,31 @@
        install --mode=0644 debian/FAQ debian/$(curpass↩
           )/usr/share/doc/$(curpass)/README.Debian
 endef

+$(patsubst %,$(stamp)binaryinst_%,$(libc)):: $(stamp)↩
   debhelper
+       mkdir -p debian/$(curpass)/usr/share/doc
+       mkdir -p debian/$(curpass)/lib/↩
   $(DEB_BUILD_GNU_TYPE)
+       mkdir -p debian/$(curpass)/usr/lib/↩
   $(DEB_BUILD_GNU_TYPE)
+       ln -sf $(libc)-common debian/$(curpass)/usr/↩
   share/doc/$(curpass)
+       if [ "$(DEB_BUILD_GNU_TYPE)" != "$(GCC_ARCH)" ↩
   ]; then \
+               ln -sf $(DEB_BUILD_GNU_TYPE) debian/↩
   $(curpass)/lib/$(GCC_ARCH) ;\
+               ln -sf $(DEB_BUILD_GNU_TYPE) debian/↩
   $(curpass)/usr/lib/$(GCC_ARCH) ;\
+       fi
+
+$(patsubst %,$(stamp)binaryinst_%,$(libc)-dev):: ↩
   $(stamp)debhelper
+       dh_installdirs -p$(curpass)
+       dh_install -p$(curpass)
+       mkdir -p debian/$(curpass)/usr/share/doc
+       ln -sf $(libc)-dev-common debian/$(curpass)/usr↩
   /share/doc/$(curpass)
+
+       if [ "$(DEB_BUILD_GNU_TYPE)" != "$(GCC_ARCH)" ↩
   ]; then \
+               ln -sf $(DEB_BUILD_GNU_TYPE) debian/↩
   $(curpass)/usr/include/$(GCC_ARCH) ;\
+       fi
+
```

```
+       for file in debian/$(curpass)/usr/lib/↩
    $(DEB_BUILD_GNU_TYPE)/*.o; do \
+               dh_link -p$(curpass)  $${file##debian/↩
    $(curpass)/} \
+               /usr/lib/$${file##*/} ; \
+       done
+
 define locales_extra_debhelper_pkg_install
        install --mode=0644 $(DEB_SRCDIR)/localedata/↩
           ChangeLog debian/$(curpass)/usr/share/doc/↩
           $(curpass)/changelog
 endef
@@ -115,6 +140,14 @@

        dh_installdeb -p$(curpass)
        # dh_shlibdeps -p$(curpass)
+       #dh_installlib -p$(curpass)
+       # Hack
+       if [ "$(curpass)" = "$(libc)" ]; then \
+               ln -sf $(DEB_BUILD_GNU_TYPE)/↩
    $$(basename $$(find debian/$(curpass)/lib -type f ↩
    -name ld-\*)) \
+               debian/$(curpass)/lib/$$(basename ↩
    $$(find debian/$(curpass)/lib -type l -name ld-\*)) ↩
    ;\
+       fi
+       #dh_installinclude -p$(curpass)
+
        dh_gencontrol -p$(curpass) -- $($(curpass)↩
           _control_flags)
        dh_md5sums -p$(curpass)
        dh_builddeb -p$(curpass)
@@ -163,6 +196,7 @@
          sed -e "s#TMPDIR#debian/tmp-libc#" -i $$z; \
          sed -e "s#DEB_SRCDIR#$(DEB_SRCDIR)#" -i $$z; ↩
             \
          sed -e "s#LIBC#$(libc)#" -i $$z; \
+         sed -e "s#DEB_BUILD_GNU_TYPE#↩
    $(DEB_BUILD_GNU_TYPE)#g" -i $$z; \
          case $$z in \
            *.install) sed -e "s/^#.*//" -i $$z ;; \
          esac; \
@@ -209,7 +243,7 @@
          cat debian/debhelper.in/libc-otherbuild.↩
             install >>$$z; \
          sed -e "s#TMPDIR#debian/tmp-$$x#" -i $$z; \
          sed -e "s#DEB_SRCDIR#$(DEB_SRCDIR)#" -i $$z; ↩
             \
-         sed -e "s#DESTLIBDIR#/tls#" -i $$z; \
```

```
+         sed -e "s#DESTLIBDIR#/$(DEB_BUILD_GNU_TYPE)/↩
   tls#g" -i $$z; \
          case $$z in \
            *.install) sed -e "s/^#.*//" -i $$z ;; \
          esac; \


--- orig/debian/sysdeps/depflags.mk
+++ mod/debian/sysdeps/depflags.mk
@@ -4,7 +4,9 @@
          perl debian/sysdeps/depflags.pl

 libc_control_flags = $(shell $(depflags) libc)
+libc_common_control_flags = $(shell $(depflags) ↩
   libc_common)
 libc_dev_control_flags = $(shell $(depflags) libc_dev)
+libc_dev_common_control_flags = $(shell $(depflags) ↩
   libc_dev_common)

 # If there's a -DDepends for libc-dev, add this to it.↩
      If there isn't
 # then the control file's depends line (which contains↩
     this) will be used.


--- orig/debian/sysdeps/depflags.pl
+++ mod/debian/sysdeps/depflags.pl
@@ -148,10 +148,29 @@
 if ($libc ne "libc6") {
     push @{$libc_dev_c{'Provides'}}, 'libc6-dev';
 }
+
+# Multiarch
+if ($libc eq "libc6") {
+    push @{$libc_c{'Depends'}}, "libc6-common";
+    push @{$libc_common_c{'Replaces'}}, 'libc6 (<< ↩
   2.3.2.ds1-20multiarch0)';
+    push @{$libc_common_c{'Conflicts'}}, 'libc6 (<< ↩
   2.3.2.ds1-20multiarch0)';
+    push @{$libc_common_c{'Depends'}}, 'libc6 (>= ↩
   2.3.2.ds1-20multiarch0)';
+    push @{$libc_dev_c{'Depends'}}, "libc6-dev-common↩
   ";
+    push @{$libc_dev_common_c{'Replaces'}}, 'libc6-dev↩
    (<< 2.3.2.ds1-20multiarch0)';
+    push @{$libc_dev_common_c{'Conflicts'}}, '↩
   libc6-dev (<< 2.3.2.ds1-20multiarch0)';
+    push @{$libc_dev_common_c{'Depends'}}, 'libc6-dev ↩
   (>= 2.3.2.ds1-20multiarch0)';
```

```
+} else {
+     die "Multiarch for $libc not implemented";
+}
+
 if ($type eq "libc") {
     %pkg = %libc_c;
+} elsif ($type eq "libc_common") {
+    %pkg = %libc_common_c;
 } elsif ($type eq "libc_dev") {
     %pkg = %libc_dev_c;
+} elsif ($type eq "libc_dev_common") {
+    %pkg = %libc_dev_common_c;
 } else {
     die "Unknown package $type";
 }


--- /dev/null
+++ mod/debian/debhelper.in/libc-common.docs
@@ -0,0 +1,10 @@
+log-test-*
+DEB_SRCDIR/BUGS
+DEB_SRCDIR/FAQ
+DEB_SRCDIR/INTERFACE
+DEB_SRCDIR/NEWS
+DEB_SRCDIR/NOTES
+DEB_SRCDIR/PROJECTS
+DEB_SRCDIR/README
+DEB_SRCDIR/hesiod/README.hesiod
+debian/TODO
--- /dev/null
+++ mod/debian/debhelper.in/libc-common.install
@@ -0,0 +1,21 @@
+debian/tmp-libc/usr/share/zoneinfo/* usr/share/↩
   zoneinfo
+debian/tmp-libc/usr/bin/iconv usr/bin
+debian/tmp-libc/usr/bin/locale usr/bin
+debian/tmp-libc/usr/bin/localedef usr/bin
+debian/tmp-libc/usr/bin/getent usr/bin
+debian/tmp-libc/usr/bin/getconf usr/bin
+debian/tmp-libc/usr/bin/catchsegv usr/bin
+debian/tmp-libc/usr/bin/glibcbug usr/bin
+debian/tmp-libc/usr/bin/tzselect usr/bin
+debian/tmp-libc/usr/bin/ldd* usr/bin
+debian/tmp-libc/usr/sbin/zdump usr/bin
+debian/tmp-libc/usr/sbin/rpcinfo usr/bin
+
+debian/tmp-libc/usr/sbin/zic usr/sbin
+debian/tmp-libc/usr/sbin/iconvconfig usr/sbin
```

```
+
+debian/tmp-libc/sbin/ldconfig sbin
+
+debian/tmp-libc/usr/lib/pt_chown usr/lib
+
+debian/local/usr_sbin/tzconfig usr/sbin
--- /dev/null
+++ mod/debian/debhelper.in/libc-common.links
@@ -0,0 +1 @@
+etc/localtime usr/share/zoneinfo/localtime
--- /dev/null
+++ mod/debian/debhelper.in/libc-common.manpages
@@ -0,0 +1,13 @@
+debian/local/manpages/catchsegv.1
+debian/local/manpages/getent.1
+debian/local/manpages/getconf.1
+debian/local/manpages/iconv.1
+debian/local/manpages/iconvconfig.8
+debian/local/manpages/ldconfig.8
+debian/local/manpages/ldd.1
+debian/local/manpages/locale.1
+debian/local/manpages/localedef.1
+debian/local/manpages/rpcinfo.8
+debian/local/manpages/tzselect.1
+debian/local/manpages/zdump.1
+debian/local/manpages/zic.8
--- /dev/null
+++ mod/debian/debhelper.in/libc-dev-common.install
@@ -0,0 +1,4 @@
+
+debian/tmp-libc/usr/bin/gencat usr/bin
+debian/tmp-libc/usr/bin/mtrace usr/bin
+debian/tmp-libc/usr/bin/rpcgen usr/bin
--- /dev/null
+++ mod/debian/debhelper.in/libc-dev-common.manpages
@@ -0,0 +1,3 @@
+debian/local/manpages/gencat.1
+debian/local/manpages/mtrace.1
+debian/local/manpages/rpcgen.1
--- /dev/null
+++ mod/debian/patches/.arch-ids/99_multiarch-ld.dpatch↩
   .id
@@ -0,0 +1 @@
+Tollef Fog Heen <tfheen@idi.ntnu.no> Wed Jan 26 ↩
   13:44:04 2005 8998.0
--- /dev/null
+++ mod/debian/patches/99_multiarch-ld.dpatch
@@ -0,0 +1,49 @@
+#! /bin/sh -e
```

```
+
+# All lines beginning with `# DP:' are a description ↩
   of the patch.
+# DP: Description: Multiarch support
+# DP: Author: Tollef Fog Heen <tfheen@debian.org>
+# DP: Upstream status: Not submitted
+# DP: Date: 2005-01-20
+
+if [ $# -ne 2 ]; then
+    echo >&2 "`basename $0`: script expects -patch|↩
   -unpatch as argument"
+    exit 1
+fi
+case "$1" in
+    -patch) patch -d "$2" -f --no-backup-if-mismatch ↩
   -p1 < $0;;
+    -unpatch) patch -d "$2" -f --no-backup-if-mismatch↩
    -R -p1 < $0;;
+    *)
+       echo >&2 "`basename $0`: script expects -patch|↩
   -unpatch as argument"
+       exit 1
+esac
+exit 0
+
+# append the patch here and adjust the -p? flag in the↩
    patch calls.
+--- glibc-2.3.2/Makeconfig     2004-05-10 ↩
   19:32:23.000000000 +0200
++++ glibc-2.3.2/Makeconfig     2004-05-10 ↩
   19:30:33.000000000 +0200
+@@ -178,6 +178,12 @@
+ endif
+ inst_slibdir = $(install_root)$(slibdir)
+
++# Extra places to look for libraries
++ifndef extra_libdir
++extra_libdir := $(exec_prefix)/lib/$(shell gcc ↩
   -dumpmachine):/lib/$(shell gcc -dumpmachine)
++endif
++
++
+ # Prefix to put on files installed in $(libdir).  For↩
    libraries `libNAME.a',
+ # the prefix is spliced between `lib' and the name, ↩
   so the linker switch
+ # `-l$(libprefix)NAME' finds the library; for other ↩
   files the prefix is
+@@ -482,6 +488,10 @@
```

```
+ default-rpath = $(libdir)
+ endif
+
++ifdef extra_libdir
++default-rpath += :$(extra_libdir)
++endif
++
+ ifndef link-extra-libs
+ ifeq (yes,$(build-shared))
+ ifneq ($(common-objpfx),$(objpfx))
--- orig/debian/debhelper.in/libc-dev.manpages
+++ /dev/null
@@ -1,3 +0,0 @@
-debian/local/manpages/gencat.1
-debian/local/manpages/mtrace.1
-debian/local/manpages/rpcgen.1
--- orig/debian/debhelper.in/libc.manpages
+++ /dev/null
@@ -1,13 +0,0 @@
-debian/local/manpages/catchsegv.1
-debian/local/manpages/getent.1
-debian/local/manpages/getconf.1
-debian/local/manpages/iconv.1
-debian/local/manpages/iconvconfig.8
-debian/local/manpages/ldconfig.8
-debian/local/manpages/ldd.1
-debian/local/manpages/locale.1
-debian/local/manpages/localedef.1
-debian/local/manpages/rpcinfo.8
-debian/local/manpages/tzselect.1
-debian/local/manpages/zdump.1
-debian/local/manpages/zic.8
```

A.2 GCC 3.4

```
--- orig/debian/changelog
+++ mod/debian/changelog
@@ -1,3 +1,22 @@
+gcc-3.4 (3.4.3-7multiarch3) multiarch; urgency=low
+
+  * Build-dep on 3.4-stuff since we want to be sure to↩
    have a multiarched
+    gnatgcc, among other things.
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Mon, 21 Feb ↩
   2005 15:37:40 +0100
+
+gcc-3.4 (3.4.3-7multiarch2) multiarch; urgency=low
+
```

```
+  * Add multiarched search paths to default search ↩
   path.
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Mon, 21 Feb ↩
   2005 15:05:03 +0100
+
+gcc-3.4 (3.4.3-7multiarch1) multiarch; urgency=low
+
+  * Multiarch build
+
+ -- Tollef Fog Heen <tfheen@debian.org>  Tue,  8 Feb ↩
   2005 16:05:33 +0100
+
 gcc-3.4 (3.4.3-7) unstable; urgency=low

   * Updated to gcc-3.4 CVS 20050108.

--- orig/debian/control.m4
+++ mod/debian/control.m4
@@ -33,21 +33,21 @@
 ifdef(`TARGET',`dnl cross
 Build-Depends: LIBC_BUILD_DEP, m4, autoconf2.13, ↩
    autoconf, automake1.4, automake1.7, libtool, ↩
    autotools-dev, gawk, bzip2, dpkg-cross (>= 1.18.1),↩
     BINUTILS_BUILD_DEP, debhelper (>= 4.1), bison (>= ↩
    1:1.875a-1) | bison (<< 1:1.50), flex, realpath (>=↩
     1.9.12)`'TARGETBD
 ',`dnl native
-Build-Depends: LIBC_BUILD_DEP, libc6-dev-sparc64 [↩
   sparc],
-  libc6-dev-s390x [s390], amd64-libs-dev [i386], ↩
   ia32-libs-dev
-  [amd64], libunwind7-dev (>= 0.98.3-3) [ia64], ↩
   libatomic-ops-dev
-  [ia64], m4, autoconf2.13, autoconf, automake1.4, ↩
   automake1.7,
-  libtool, autotools-dev, gawk, dejagnu (>= 1.4.3) [↩
   check_no_archs],
-  expect (>= 5.38.0) [check_no_archs], bzip2, ↩
   BINUTILS_BUILD_DEP,
-  binutils-hppa64 [hppa], debhelper (>= 4.1), gperf (↩
   >= 2.7-3), bison
-  (>= 1:1.875a-1) | bison (<< 1:1.50), flex, gettext, ↩
   texinfo (>= 4.3),
-  zlib1g-dev, libgc-dev [libgc_no_archs], xlibs-dev, ↩
   gnat-3.3
-  [ada_no_archs] | gnat-3.4 [ada_no_archs] | gnat [↩
   i386 powerpc sparc],
```

```
-  libncurses5-dev [pascal_no_archs], libgmp3-dev, ↩
   tetex-bin
-  [pascal_no_archs], locales [locale_no_archs !↩
   hurd-i386], procps
-  [check_no_archs], help2man [pascal_no_archs], ↩
   sharutils, libgtk2.0-dev
-  (>= 2.4.4-2) [java_no_archs], libart-2.0-dev [↩
   java_no_archs], g++-3.3
-  [!amd64], g77-3.3 [!amd64], gobjc-3.3 [!amd64], ↩
   realpath (>= 1.9.12)
+Build-Depends: LIBC_BUILD_DEP, libc6-dev-sparc64 [↩
   sparc],
+  libc6-dev-s390x [s390], amd64-libs-dev [i386], ↩
   ia32-libs-dev [amd64],
+  libunwind7-dev (>= 0.98.3-3) [ia64], ↩
   libatomic-ops-dev [ia64], m4,
+  autoconf2.13, autoconf, automake1.4, automake1.7, ↩
   libtool,
+  autotools-dev, gawk, dejagnu (>= 1.4.3) [↩
   check_no_archs], expect (>=
+  5.38.0) [check_no_archs], bzip2, BINUTILS_BUILD_DEP,↩
    binutils-hppa64
+  [hppa], debhelper (>= 4.1), gperf (>= 2.7-3), bison ↩
   (>= 1:1.875a-1)
+  | bison (<< 1:1.50), flex, gettext, texinfo (>= 4.3)↩
   , zlib1g-dev,
+  libgc-dev [libgc_no_archs], xlibs-dev, gnat-3.4 [↩
   ada_no_archs] |
+  gnat-3.4 [ada_no_archs] | gnat [i386 powerpc sparc],↩
    libncurses5-dev
+  [pascal_no_archs], libgmp3-dev, tetex-bin [↩
   pascal_no_archs], locales
+  [locale_no_archs !hurd-i386], procps [check_no_archs↩
   ], help2man
+  [pascal_no_archs], sharutils, libgtk2.0-dev (>= ↩
   2.4.4-2)
+  [java_no_archs], libart-2.0-dev [java_no_archs], g++↩
   -3.4 [!amd64],
+  g77-3.4 [!amd64], gobjc-3.4 [!amd64], realpath (>= ↩
   1.9.12)
 Build-Depends-Indep: doxygen (>= 1.3.9.1)
 ')dnl


--- orig/debian/patches/multiarch-include.dpatch
+++ mod/debian/patches/multiarch-include.dpatch
@@ -43,15 +43,18 @@
 diff -urN gcc.old/config/i386/t-linux64 gcc/config/↩
    i386/t-linux64
```

```
 --- gcc.old/config/i386/t-linux64      2002-11-28 ↩
    15:47:02.000000000 +0100
 +++ gcc/config/i386/t-linux64  2004-07-10 ↩
    01:44:11.000000000 +0200
-@@ -6,7 +6,7 @@
+@@ -6,7 +6,9 @@

  MULTILIB_OPTIONS = m64/m32
  MULTILIB_DIRNAMES = 64 32
 -MULTILIB_OSDIRNAMES = ../lib64 ../lib
-+MULTILIB_OSDIRNAMES = x86_64-linux i486-linux
++# lib64 is here for backwards compatibility purposes.
++# MULTILIB_OSDIRNAMES seems to be relative to libdir.↩
     They are ordered.
++MULTILIB_OSDIRNAMES = ../x86_64-linux .

  LIBGCC = stmp-multilib
  INSTALL_LIBGCC = install-multilib
+# OLD: x86_64-linux i486-linux ../lib64
 diff -urN gcc.old/config/t-linux gcc/config/t-linux
 --- gcc.old/config/t-linux     2003-09-23 ↩
    20:55:57.000000000 +0200
 +++ gcc/config/t-linux 2004-07-10 01:48:41.000000000 ↩
    +0200
@@ -62,7 +65,7 @@
 +
 +MULTILIB_OPTIONS = m32
 +MULTILIB_DIRNAMES = 32
-+MULTILIB_OSDIRNAMES = $TARGET_ARCH
++MULTILIB_OSDIRNAMES = .
 +
 +LIBGCC = stmp-multilib
 +INSTALL_LIBGCC = install-multilib
@@ -105,3 +108,18 @@
  #ifdef STANDARD_INCLUDE_DIR
      /* /usr/include comes dead last.  */
      { STANDARD_INCLUDE_DIR, ↩
         STANDARD_INCLUDE_COMPONENT, 0, 0, 1 },
+--- gcc/gcc.c~ 2005-02-21 13:23:22.000000000 +0100
++++ gcc/gcc.c  2005-02-21 14:22:49.985527617 +0100
+@@ -3222,6 +3222,12 @@
+                 PREFIX_PRIORITY_LAST, 0, NULL, 0);
+       add_prefix (&startfile_prefixes, ↩
    gcc_exec_prefix, "GCC",
+                 PREFIX_PRIORITY_LAST, 0, NULL, 0);
++      temp = xmalloc(strlen(gcc_exec_prefix) + 1 + ↩
    strlen(spec_machine) + 1);
++      sprintf(temp,"%s/%s", gcc_exec_prefix, ↩
    spec_machine);
```

```
++
++      add_prefix (&startfile_prefixes, temp, "GCC",
++                  PREFIX_PRIORITY_LAST, 0, NULL, 0);
++      free(temp);
+     }
+
+   /* COMPILER_PATH and LIBRARY_PATH have values

--- orig/debian/rules.d/binary-fortran.mk
+++ mod/debian/rules.d/binary-fortran.mk
@@ -51,11 +51,11 @@
        $(gcc_lib_dir)/include/g2c.h

 ifeq ($(with_lib64g2c),yes)
-       dirs_g2c   += $(PF)/$(lib64)
-       files_g2c  += $(PF)/$(lib64)/libg2c.so.*
+       dirs_g2c   += $(PF)/$(lib64dir)
+       files_g2c  += $(PF)/$(lib64dir)/libg2c.so.*

-       dirs_g2cd  += $(PF)/$(lib64)
-       files_g2cd += $(PF)/$(lib64)/{libg2c.{a,la,so},↩
   libfrtbegin.a}
+       dirs_g2cd  += $(PF)/$(lib64dir)
+       files_g2cd += $(PF)/$(lib64dir)/{libg2c.{a,la,↩
   so},libfrtbegin.a}
 endif

 # ↩
    ----------------------------------------------------------------------↩

@@ -121,10 +121,10 @@

 #ifeq ($(biarch),yes)
 #  ifeq ($(DEB_TARGET_GNU_CPU),i386)
-#      mv $(d)/$(PF)/$(lib64)/libg2c.{a,la,so} $(d)/↩
   $(gcc_lib_dir)/64
+#      mv $(d)/$(PF)/$(lib64dir)/libg2c.{a,la,so} $(d)↩
   /$(gcc_lib_dir)/64
 #      ln -sf ../../../../lib64/libg2c.so.$(F77_SONAME↩
    ) \
 #               $(d)/$(gcc_lib_dir)/64/libg2c.so
-#      mv $(d)/$(PF)/$(lib64)/libfrtbegin.a $(d)/↩
   $(gcc_lib_dir)/64
+#      mv $(d)/$(PF)/$(lib64dir)/libfrtbegin.a $(d)/↩
   $(gcc_lib_dir)/64
 #  endif
 #endif
        rm -rf $(d_g77)
```

```
--- orig/debian/rules.d/binary-gcc-cross.mk
+++ mod/debian/rules.d/binary-gcc-cross.mk
@@ -68,9 +68,9 @@
         rm -f $(d)/$(PF)/$(libdir)/libgcc_s.so
         ln -sf /$(PF)/$(DEB_TARGET_GNU_TYPE)/$(libdir)/↩
            libgcc_s.so.$(GCC_SONAME) $(d)/$(gcc_lib_dir↩
            )/libgcc_s.so
 ifeq ($(biarch),yes)
-        rm -f $(d)/$(PF)/$(lib64)/libgcc_s.so
-        ln -sf /$(PF)/$(DEB_TARGET_GNU_TYPE)/$(lib64)/↩
    libgcc_s.so.$(GCC_SONAME) $(d)/$(gcc_lib_dir)/↩
    libgcc_s_64.so
-        ln -sf /$(PF)/$(DEB_TARGET_GNU_TYPE)/$(lib64)/↩
    libgcc_s.so.$(GCC_SONAME) $(d)/$(gcc_lib_dir)/64/↩
    libgcc_s.so
+        rm -f $(d)/$(PF)/$(lib64dir)/libgcc_s.so
+        ln -sf /$(PF)/$(DEB_TARGET_GNU_TYPE)/$(lib64dir↩
    )/libgcc_s.so.$(GCC_SONAME) $(d)/$(gcc_lib_dir)/↩
    libgcc_s_64.so
+        ln -sf /$(PF)/$(DEB_TARGET_GNU_TYPE)/$(lib64dir↩
    )/libgcc_s.so.$(GCC_SONAME) $(d)/$(gcc_lib_dir)/64/↩
    libgcc_s.so
 endif

         DH_COMPAT=2 dh_movefiles -p$(p_gcc) $(files_gcc↩
            )


--- orig/debian/rules.d/binary-gcc.mk
+++ mod/debian/rules.d/binary-gcc.mk
@@ -78,10 +78,10 @@

         rm -f $(d)/$(PF)/$(libdir)/libgcc_s.so
         ln -sf /$(libdir)/libgcc_s.so.$(GCC_SONAME) $(d↩
            )/$(gcc_lib_dir)/libgcc_s.so
-ifeq ($(biarch),yes)
-        rm -f $(d)/$(PF)/$(lib64)/libgcc_s.so
-        ln -sf /$(lib64)/libgcc_s.so.$(GCC_SONAME) $(d)↩
    /$(gcc_lib_dir)/libgcc_s_64.so
-        ln -sf /$(lib64)/libgcc_s.so.$(GCC_SONAME) $(d)↩
    /$(gcc_lib_dir)/64/libgcc_s.so
+ifneq (,$(findstring(yes, $(biarch) $(multiarch))))
+        rm -f $(d)/$(PF)/$(lib64dir)/libgcc_s.so
+        ln -sf /$(lib64dir)/libgcc_s.so.$(GCC_SONAME) ↩
    $(d)/$(gcc_lib_dir)/libgcc_s_64.so
+        ln -sf /$(lib64dir)/libgcc_s.so.$(GCC_SONAME) ↩
    $(d)/$(gcc_lib_dir)/64/libgcc_s.so
 endif
```

```
 ifeq ($(biarch_ia32),yes)
        mkdir -p $(d_gcc)/$(gcc_lib_dir)


--- orig/debian/rules.d/binary-java.mk
+++ mod/debian/rules.d/binary-java.mk
@@ -100,17 +100,17 @@
        $(PF)/$(libdir)/lib-org-*.so

 ifeq ($(with_lib64gcj),yes)
-       dirs_jlib    += $(PF)/$(lib64)
-       files_jlib   += $(PF)/$(lib64)/libgcj*.so.* \
-                       $(PF)/$(lib64)/lib-org-*.so.*
-
-       dirs_jlibx   += $(PF)/$(lib64)
-       files_jlibx  += $(PF)/$(lib64)/↩
    lib-gnu-java-awt-*.so.*
-
-       dirs_jdev    += $(PF)/$(lib64)
-       files_jdev   += $(PF)/$(lib64)/libgcj*.{a,so,la↩
    } \
-                       $(PF)/$(lib64)/lib-gnu-*.{a,so,↩
    la} \
-                       $(PF)/$(lib64)/lib-org-*.{a,so,↩
    la}
+       dirs_jlib    += $(PF)/$(lib64dir)
+       files_jlib   += $(PF)/$(lib64dir)/libgcj*.so.* ↩
    \
+                       $(PF)/$(lib64dir)/lib-org-*.so↩
    .*
+
+       dirs_jlibx   += $(PF)/$(lib64dir)
+       files_jlibx  += $(PF)/$(lib64dir)/↩
    lib-gnu-java-awt-*.so.*
+
+       dirs_jdev    += $(PF)/$(lib64dir)
+       files_jdev   += $(PF)/$(lib64dir)/libgcj*.{a,so↩
    ,la} \
+                       $(PF)/$(lib64dir)/lib-gnu-*.{a,↩
    so,la} \
+                       $(PF)/$(lib64dir)/lib-org-*.{a,↩
    so,la}
 endif

 # ↩
     ----------------------------------------------------------------------↩
```

```
--- orig/debian/rules.d/binary-libffi.mk
+++ mod/debian/rules.d/binary-libffi.mk
@@ -22,10 +22,10 @@
        $(PF)/$(libdir)/libffi.{a,so,la}

 ifeq ($(with_lib64ffi),yes)
-       dirs_ffi   += $(PF)/$(lib64)
-       files_ffi  += $(PF)/$(lib64)/libffi.so.*
-       dirs_ffid  += $(PF)/$(lib64)
-       files_ffid += $(PF)/$(lib64)/libffi.{a,so,la}
+       dirs_ffi   += $(PF)/$(lib64dir)
+       files_ffi  += $(PF)/$(lib64dir)/libffi.so.*
+       dirs_ffid  += $(PF)/$(lib64dir)
+       files_ffid += $(PF)/$(lib64dir)/libffi.{a,so,la↩
    }
 endif

 $(binary_stamp)-libffi: $(install_stamp)


--- orig/debian/rules.d/binary-libgcc-cross.mk
+++ mod/debian/rules.d/binary-libgcc-cross.mk
@@ -17,9 +17,9 @@

 ifeq ($(with_shared_libgcc),yes)
 files_lgcc = \
-       $(PF)/$(DEB_TARGET_GNU_TYPE)/lib/libgcc_s.so.↩
    $(GCC_SONAME)
+       $(PF)/lib/$(DEB_TARGET_GNU_TYPE)/libgcc_s.so.↩
    $(GCC_SONAME)
 files_l64gcc = \
-       $(PF)/$(DEB_TARGET_GNU_TYPE)/lib64/libgcc_s.so.↩
    $(GCC_SONAME)
+       $(PF)/lib/$(DEB_TARGET_GNU_TYPE)/libgcc_s.so.↩
    $(GCC_SONAME)
 endif

 # ↩
    ------------------------------------------------------------------------↩

@@ -91,7 +91,7 @@
        rm -rf $(d_l64gcc)
        dh_installdirs -p$(p_l64gcc) \
                $(docdir)/$(p_l64gcc) \
-               $(PF)/$(DEB_TARGET_GNU_TYPE)/lib64
+               $(PF)/lib/$(DEB_TARGET_GNU_TYPE)

 ifeq ($(with_shared_libgcc),yes)
```

```
        DH_COMPAT=2 dh_movefiles -p$(p_l64gcc) ↩
           $(files_l64gcc)
@@ -137,7 +137,7 @@
        rm -rf $(d_l32gcc)
        dh_installdirs -p$(p_l32gcc) \
                $(docdir)/$(p_l32gcc) \
-               $(PF)/$(DEB_TARGET_GNU_TYPE)/lib32
+               $(PF)/lib/$(DEB_TARGET_GNU_TYPE)

 ifeq ($(with_shared_libgcc),yes)
        DH_COMPAT=2 dh_movefiles -p$(p_l32gcc) ↩
           $(files_l32gcc)


--- orig/debian/rules.d/binary-libgcc.mk
+++ mod/debian/rules.d/binary-libgcc.mk
@@ -19,7 +19,7 @@
 files_lgcc = \
        $(libdir)/libgcc_s.so.$(GCC_SONAME)
 files_l64gcc = \
-       lib64/libgcc_s.so.$(GCC_SONAME)
+       $(lib64dir)/libgcc_s.so.$(GCC_SONAME)
 endif

 # ↩
    ------------------------------------------------------------------------↩

@@ -38,7 +38,7 @@
                $(libdir)

 ifeq ($(with_shared_libgcc),yes)
-       mv $(d)/$(PF)/lib/libgcc_s.so.$(GCC_SONAME) $(d↩
   )/$(libdir)/.
+       mv $(d)/$(PF)/lib/$(DEB_BUILD_GNU_TYPE)/↩
   libgcc_s.so.$(GCC_SONAME) $(d)/$(libdir)/
        DH_COMPAT=2 dh_movefiles -p$(p_lgcc) ↩
           $(files_lgcc)
 endif

@@ -79,15 +79,14 @@
        dh_testdir
        dh_testroot
        mv $(install_stamp) $(install_stamp)-tmp
-
        rm -rf $(d_l64gcc)
        dh_installdirs -p$(p_l64gcc) \
                $(docdir)/$(p_l64gcc) \
-               lib64
+               $(lib64dir)
```

```
 ifeq ($(with_shared_libgcc),yes)
-       install -d $(d)/lib64
-       mv $(d)/$(PF)/lib64/libgcc_s.so.$(GCC_SONAME) ↩
   $(d)/lib64/.
+       install -d $(d)/$(lib64dir)
+       mv $(d)/$(PF)/$(lib64dir)/libgcc_s.so.↩
   $(GCC_SONAME) $(d)/$(lib64dir)/.
 endif
         DH_COMPAT=2 dh_movefiles -p$(p_l64gcc) ↩
            $(files_l64gcc)



--- orig/debian/rules.d/binary-libobjc.mk
+++ mod/debian/rules.d/binary-libobjc.mk
@@ -16,10 +16,10 @@
 endif

 ifeq ($(with_lib64objc),yes)
-       dirs_lobjc    += $(PF)/$(lib64)
-       files_lobjc   += $(PF)/$(lib64)/libobjc.so.*
+       dirs_lobjc    += $(PF)/$(lib64dir)
+       files_lobjc   += $(PF)/$(lib64dir)/libobjc.so.*
    ifeq ($(with_objc_gc),yes)
-       files_lobjc   += $(PF)/$(lib64)/libobjc_gc.so.*
+       files_lobjc   += $(PF)/$(lib64dir)/libobjc_gc.so↩
   .*
   endif
 endif



--- orig/debian/rules.d/binary-libstdcxx.mk
+++ mod/debian/rules.d/binary-libstdcxx.mk
@@ -45,13 +45,13 @@

 dirs_lib64 = \
         $(docdir) \
-       $(PF)/lib64
+       $(PF)/$(lib64dir)

 files_lib = \
         $(PF)/$(libdir)/libstdc++.so.*

 files_lib64 = \
-       $(PF)/lib64/libstdc++.so.*
+       $(PF)/$(lib64dir)/libstdc++.so.*
```

```
 dirs_dev = \
        $(docdir)/$(p_base)/C++ \
@@ -83,8 +83,8 @@
     dirs_dev += $(gcc_lib_dir)/64/
     files_dev += $(gcc_lib_dir)/64/libstdc++.{a,so} \
         $(gcc_lib_dir)/64/libsupc++.a
-    dirs_dbg += $(PF)/lib64/debug
-    files_dbg += $(PF)/lib64/debug/libstdc++.*
+    dirs_dbg += $(PF)/$(lib64dir)/debug
+    files_dbg += $(PF)/$(lib64dir)/debug/libstdc++.*
     dirs_pic += $(gcc_lib_dir)
     files_pic += $(gcc_lib_dir)/64/libstdc++_pic.a
 endif
@@ -284,13 +284,13 @@
        mv $(d)/$(PF)/$(libdir)/libstdc++_pic.a $(d)/↩
           $(gcc_lib_dir)/

        rm -f $(d)/$(PF)/$(libdir)/debug/libstdc++_pic.↩
           a
-       rm -f $(d)/$(PF)/lib64/debug/libstdc++_pic.a
+       rm -f $(d)/$(PF)/$(lib64dir)/debug/libstdc++↩
   _pic.a

        : # remove precompiled headers
        -find $(d) -type d -name '*.gch' | xargs rm -rf

 ifeq ($(with_lib64cxx),yes)
-       mv $(d)/$(PF)/lib64/lib*c++*.{a,so} $(d)/↩
   $(gcc_lib_dir)/64/.
+       mv $(d)/$(PF)/$(lib64dir)/lib*c++*.{a,so} $(d)/↩
   $(gcc_lib_dir)/64/.
 endif
 ifeq ($(biarch_ia32),yes)
        mv $(d)/$(PF)/lib32/lib*c++*.{a,so} $(d)/↩
           $(gcc_lib_dir)/32/.
@@ -305,7 +305,7 @@
                /$(gcc_lib_dir)/libstdc++.so
 ifeq ($(with_lib64cxx),yes)
        dh_link -p$(p_dev) \
-               /$(PF)/lib64/libstdc++.so.$(CXX_SONAME)↩
    \
+               /$(PF)/$(lib64dir)/libstdc++.so.↩
   $(CXX_SONAME) \
                /$(gcc_lib_dir)/64/libstdc++.so
 endif
 ifeq ($(biarch_ia32),yes)
@@ -339,7 +339,7 @@
        dh_compress -p$(p_dev) -p$(p_pic) -p$(p_dbg) -X↩
           .txt
```

```
        dh_fixperms -p$(p_dev) -p$(p_pic) -p$(p_dbg)
 ifeq ($(with_lib64cxx),yes)
-        dh_shlibdeps -p$(p_dev) -p$(p_pic) -p$(p_dbg) ↩
    -Xlib64
+        dh_shlibdeps -p$(p_dev) -p$(p_pic) -p$(p_dbg) ↩
    -X$(lib64dir)
 else
        dh_shlibdeps -p$(p_dev) -p$(p_pic) -p$(p_dbg) ↩
           -Xlib32/debug
 endif


--- orig/debian/rules.defs
+++ mod/debian/rules.defs
@@ -8,8 +8,6 @@
 builddir_hppa64        = $(PWD)/build-hppa64
 stampdir       = stamps

-lib64          = lib64
-
 # version of default gcc compiler (e.g. 3.3 or 3.4)
 BUILD_CC_VERSION:= $(shell gcc -dumpversion | sed 's/\↩
    ([0-9]*\.[0-9]*\)\(.*\)/\1/')

@@ -142,7 +140,7 @@
 #endif

 # multiarch --------------------
-with_multiarch := no
+with_multiarch := yes

 # "old" arm ABI (with 3.5, we get a new one) ↩
    --------------------
 ifeq ($(DEB_TARGET_GNU_CPU),arm)
@@ -493,8 +491,8 @@
 endif

 # Shared libgcc --------------------
-with_libgcc := yes
-with_shared_libgcc := yes
+with_libgcc := no
+with_shared_libgcc := no

 #ifeq ($(with_common_libs),yes)
 #  with_libgcc := yes


--- orig/debian/rules2
+++ mod/debian/rules2
```

```
@@ -37,8 +37,9 @@
 # the recipient for the test summaries. Send with: ↩
    debian/rules mail-summary
 S_EMAIL = gcc@packages.debian.org gcc-testresults@gcc.↩
    gnu.org

-CPPFLAGS       =
+CPPFLAGS       =
 CFLAGS         = -g -O2 $(CPPFLAGS)
+#CFLAGS                += -DSTANDARD_INCLUDE_DIR=\"/↩
   usr/include/i386-linux\"
 LDFLAGS                =
 BOOT_CFLAGS    = -O2 $(CPPFLAGS)
 ifeq ($(with_ada),yes)
@@ -63,9 +64,16 @@

 ifeq ($(with_multiarch),yes)
   libdir       = lib/$(DEB_TARGET_GNU_TYPE)
+  ifeq ($(DEB_TARGET_GNU_TYPE),i386-linux)
+       lib64dir = lib/x86_64-linux
+  else
+       lib64dir = lib64
+  endif
 else
   libdir       = lib
+  lib64dir = lib64
 endif
+
 # /usr/libexec doesn't follow the FHS
 libexecdir     = $(PF)/$(libdir)
 buildlibdir    = $(builddir)/$(TARGET_ALIAS)
@@ -73,15 +81,21 @@
 gcc_lexec_dir  = $(libexecdir)/gcc/$(TARGET_ALIAS)/↩
    $(VER)

 ifndef DEB_CROSS
-  cxx_inc_dir  = $(PF)/include/c++/$(BASE_VERSION)
+  ifeq ($(with_multiarch),yes)
+    cxx_inc_dir        = $(PF)/include/↩
   $(DEB_TARGET_GNU_TYPE)/c++/$(BASE_VERSION)
+  else
+    cxx_inc_dir        = $(PF)/include/c++/↩
   $(BASE_VERSION)
+  endif
 else
   cxx_inc_dir  = $(PF)/$(TARGET_ALIAS)/include/c++/↩
      $(BASE_VERSION)
 endif
```

```
+
 CONFARGS = -v \
        --enable-languages=$(shell echo ↩
           $(enabled_languages) | tr -s ' ' ',') \
        --prefix=/$(PF) \
        --libexecdir=/$(libexecdir) \
+       --libdir=/$(PF)/$(libdir) \
        --with-gxx-include-dir=/$(cxx_inc_dir) \
        --enable-shared \
        --with-system-zlib \
@@ -176,7 +190,10 @@
   bootstrap_target = bootstrap-lean
 endif

-bootstrap_target = bootstrap-lean
+# Work around bug where bootstrap-lean causes stage2/ ↩
   to disappear
+# before "Building runtime libraries"
+
+bootstrap_target = bootstrap

 # Increase the timeout for one testrun on slow ↩
    architectures
 ifeq ($(DEB_TARGET_GNU_CPU), $(findstring ↩
    $(DEB_TARGET_GNU_CPU),arm hppa m68k))
@@ -983,6 +1000,14 @@
        ln -s lib $(d)/usr/lib64
 endif

+ifeq ($(DEB_TARGET_GNU_CPU), $(findstring ↩
   $(DEB_TARGET_GNU_CPU),i386))
+       mkdir -p $(d)/usr/lib $(d)/lib
+       ln -s i386-linux $(d)/usr/lib/i486-linux
+       ln -s i386-linux $(d)/lib/i486-linux
+#      mkdir -p $(d)/usr/lib/i386-linux/ $(d)/usr/lib/↩
   x86_64-linux
+#      ln -s ../x86_64-linux $(d)/usr/lib/i486-linux/↩
   x86_64-linux
+endif
+
        : # Install everything
        PATH=$(PWD)/bin:$$PATH \
          $(MAKE) -C $(builddir) \
```

A.3 Binutils

```
--- orig/debian/changelog
+++ mod/debian/changelog
@@ -1,3 +1,21 @@
```

```
+binutils (2.15-5multiarch3) multiarch; urgency=low
+
+  * Make sure to actually _apply_ the patch as well..
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Thu, 24 Feb ↩
    2005 15:46:08 +0100
+
+binutils (2.15-5multiarch2) multiarch; urgency=low
+
+  * Rebuild.
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Thu, 24 Feb ↩
    2005 14:35:58 +0100
+
+binutils (2.15-5multiarch1) multiarch; urgency=low
+
+  * Multiarch build
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Thu, 24 Feb ↩
    2005 14:14:03 +0100
+
 binutils (2.15-5) unstable; urgency=low

   * 121_ia64_unwind_fixes.dpatch: new patch from David↩
       Mosberger to fix


--- orig/debian/patches/00list
+++ mod/debian/patches/00list
@@ -15,3 +15,4 @@
 120_mips_xgot_multigot_workaround
 121_ia64_unwind_fixes
 122_m68k_undefweak_symbols
+124_multiarch

--- /dev/null
+++ mod/debian/patches/124_multiarch.dpatch
@@ -0,1 +1,41 @@
+#!/bin/sh -e
+## 000_print_debian_version.dpatch
+##
+## All lines beginning with `## DP:' are a description↩
    of the patch.
+## DP: Description: Multiarch support
+## DP: Author: Tollef Fog Heen <tfheen@debian.org>
+## DP: Upstream status: Not submitted
+## DP: Date: 2005-01-26
+
+if [ $# -ne 1 ]; then
```

```
+    echo >&2 "`basename $0`: script expects -patch|↩
   -unpatch as argument"
+    exit 1
+fi
+
+[ -f debian/patches/00patch-opts ] && . debian/patches↩
   /00patch-opts
+patch_opts="${patch_opts:--f --no-backup-if-mismatch}"
+
+case "$1" in
+       -patch) patch $patch_opts -p1 < $0;;
+       -unpatch) patch $patch_opts -p1 -R < $0;;
+        *)
+                echo >&2 "`basename $0`: script ↩
   expects -patch|-unpatch as argument"
+                exit 1;;
+esac
+
+exit 0
+
+@DPATCH@
+--- binutils-2.14.90.0.6/ld/genscripts.sh.old  2003↩
   -08-21 16:28:47.000000000 +0100
++++ binutils-2.14.90.0.6/ld/genscripts.sh      2005↩
   -01-26 16:28:48.000000000 +0100
+@@ -188,7 +188,7 @@
+   esac
+ fi
+
+-LIB_SEARCH_DIRS=`echo ${LIB_PATH} | sed -e 's/:/ /g' ↩
   -e 's/\([^ ][^ ]*\)/SEARCH_DIR(\"\1\");/g'`
++LIB_SEARCH_DIRS=`echo ${LIB_PATH} | sed -e 's/:/ /g' ↩
   -e 's/\([^ ][^ ]*\)/SEARCH_DIR(\"\1\"); ↩
   SEARCH_DIR(\"\1\/'${TOOL_LIB}'");/g'`
+
+ # We need it for testsuite.
+ case " $EMULATION_LIBPATH " in
+
```

A.4 dpkg

```
--- orig/configure.ac
+++ mod/configure.ac
@@ -10,6 +10,17 @@

 AM_INIT_AUTOMAKE([1.8 gnu])

+dpkg_archlist=''
+AC_MSG_CHECKING(Debian compatible-architecture list)
```

```
+dpkg_archlist="`awk '$1 == "'$dpkg_archset'" { $1=""; ↩
   print $0 }' $srcdir/subarchtable`"
+if test "x$dpkg_archlist" = "x"; then
+   AC_MSG_RESULT([$dpkg_archset not found in ↩
   subarchtable])
+   dpkg_archlist='{ "'$dpkg_archset'" }'
+   else
+   AC_MSG_RESULT([found])
+fi
+AC_DEFINE_UNQUOTED(ARCH_LIST, ${dpkg_archlist}, [Set ↩
   this to the list of allowable architectures for this↩
    CPU type.])
+
 AM_GNU_GETTEXT_VERSION(0.14.1)
 AM_GNU_GETTEXT()



--- orig/debian/changelog
+++ mod/debian/changelog
@@ -1,3 +1,9 @@
+dpkg (1.13.2~multiarch1) multiarch; urgency=low
+
+  * Update multiarch patches to current newest version
+
+ -- Tollef Fog Heen <tfheen@debian.org>  Tue,  8 Mar ↩
   2005 13:50:31 +0100
+
 dpkg (1.13.2~) experimental; urgency=low

   *
@@ -278,6 +284,22 @@

  -- Scott James Remnant <scott@netsplit.com>  Tue,  1 ↩
     Jun 2004 18:21:40 -0300

+dpkg (1.10.21.multiarch.20040601.01) unstable; urgency↩
   =low
+
+  * Fix (badly) problem of migrating uniarch libs to ↩
   multiarch.
+
+ -- Hugo Mills <hugo@carfax.org.uk>  Tue, 1 Jun 2004 ↩
   20:29:40 +0100
+
+dpkg (1.10.21.multiarch.20040528.01) unstable; urgency↩
   =low
+
+  * Accept Multi-Arch: field in control files.
```

```
+  * Add database of acceptable architectures for ↩
   multi-arch systems.
+  * Use libfoo:arch for package name internally for ↩
   all multi-arch
+    packages.
+  * Handle shared symlinks for multi-arch packages.
+
+ -- Hugo Mills <hugo@carfax.org.uk>  Fri, 28 May 2004 ↩
   20:18:19 +0100
+
 dpkg (1.10.21) unstable; urgency=low

   * Fix incorrect linked list node removal code that ↩
      caused every second


--- orig/dpkg-deb/build.c
+++ mod/dpkg-deb/build.c
@@ -211,9 +211,11 @@
     parsedb(controlfile, pdb_recordavailable|↩
        pdb_rejectstatus,
            &checkedinfo, stderr, &warns);
     assert(checkedinfo->available.valid);
+       /* The name of the package may be libfoo:arch ↩
   if it's a multiarch
+        * package. */
     if (strspn(checkedinfo->name,
                "abcdefghijklmnopqrstuvwxyz0123456789+-↩
                   .")
-        != strlen(checkedinfo->name))
+        < strcspn(checkedinfo->name, ":"))
       ohshit(_("package name has characters that aren't↩
          t lowercase alphanums or `-+.'"));
     if (checkedinfo->priority == pri_other) {
       fprintf(stderr, _("warning, `%s' contains ↩
          user-defined Priority value `%s'\n"),


--- orig/lib/database.c
+++ mod/lib/database.c
@@ -86,6 +86,7 @@
   pifp->conffiles= NULL;
   pifp->arbs= NULL;
   pifp->valid= 1;
+  pifp->multiarch= 0;
 }

 static int nes(const char *s) { return s && *s; }
@@ -121,6 +122,15 @@
```

```
   struct pkginfo **pointerp, *newpkg;
   char *name = strdup(inname), *p;

+/*
+  if(strchr(inname, ':') || strncmp(inname, "libfoo", ↩
   6) == 0)
+  {
+          fprintf(stderr,
+                          "Findpackage looking for an ↩
   interesting package: %s\n",
+                          inname);
+  }
+*/
+
   if (name == NULL)
     ohshite(_("couldn't allocate memory for strdup in ↩
        findpackage(%s)"),inname);
   p= name;
@@ -129,7 +139,22 @@
   pointerp= bins + (hash(name) % (BINS));
   while (*pointerp && strcasecmp((*pointerp)->name,↩
      name))
     pointerp= &(*pointerp)->next;
-  if (*pointerp) { free(name); return *pointerp; }
+  if (*pointerp)
+  {
+          free(name);
+
+/*
+          if(strchr(inname, ':') || strncmp(inname, "↩
   libfoo", 6) == 0)
+          {
+                  fprintf(stderr, "  Found package: ↩
   multiarch flags are: "
+                                  "%d (available) %d ↩
   (installed)\n",
+                                  (*pointerp)->↩
   available.multiarch,
+                                  (*pointerp)->↩
   installed.multiarch);
+          }
+*/
+
+          return *pointerp;
+  }

   newpkg= nfmalloc(sizeof(struct pkginfo));
   blankpackage(newpkg);
```

```
--- orig/lib/dpkg-db.h
+++ mod/lib/dpkg-db.h
@@ -103,6 +103,7 @@
   struct versionrevision version;
   struct conffile *conffiles;
   struct arbitraryfield *arbs;
+  int multiarch; /* The 'Multi-Arch' flag, 1=yes, 0=no↩
    */
 };

 struct perpackagestate; /* dselect and dpkg have ↩
    different versions of this */


--- orig/lib/dpkg.h
+++ mod/lib/dpkg.h
@@ -108,6 +108,7 @@
 #define DEFAULTSHELL        "sh"
 #define PAGERENV            "PAGER"
 #define DEFAULTPAGER        "pager"
+#define ARCHLISTENV         "DPKG_ARCH"

 #define IMETHODMAXLEN        50
 #define IOPTIONMAXLEN        IMETHODMAXLEN
@@ -362,6 +363,10 @@

 extern volatile int onerr_abort;

+/*** from multi-arch.c ***/
+
+int is_allowable_arch(char*);
+
 /*** from showcright.c ***/

 struct cmdinfo;


--- orig/lib/dump.c
+++ mod/lib/dump.c
@@ -42,7 +42,11 @@
   assert(pigp->name);
   if (flags&fw_printheader)
     varbufaddstr(vb,"Package: ");
-  varbufaddstr(vb, pigp->name);
+  if (pifp->multiarch) {
+       varbufaddbuf(vb, pigp->name, strcspn(pigp->name↩
   , ":"));
+  } else {
```

```
+       varbufaddstr(vb, pigp->name);
+  }
   if (flags&fw_printheader)
     varbufaddc(vb,'\n');
 }


--- orig/lib/fields.c
+++ mod/lib/fields.c
@@ -70,9 +70,24 @@
              const char *filename, int lno, FILE *↩
                 warnto, int *warncount,
              const char *value, const struct fieldinfo ↩
                 *fip) {
   const char *e;
+  char *p;
   if ((e= illegal_packagename(value,NULL)) != NULL)
     parseerr(NULL,filename,lno, warnto,warncount,pigp↩
        ,0, _("invalid package name (%.250s)"),e);
-  pigp->name= findpackage(value)->name;
+  pigp->name= strdup(value);
+  /* @@@ FIXME Check for return value of the strdup ↩
   here and bitch
+   * about it if it couldn't allocate memory */
+  /* @@@ Optimisation opportunity: allocate a single ↩
   static buffer for
+   * this purpose, instead of using strdup() */
+
+  /* @@@ Don't use findpackage here yet -- do the ↩
   tolower mangling
+   * ourselves. The findpackage will be called after ↩
   all the fields
+   * have been read in, and the architecture is known.↩
     Also, allocate
+   * our own temporary memory for the name here.
+   */
+  p= pigp->name;
+  while(*p) { *p= tolower(*p); p++; }
+
+/*  pigp->name= findpackage(value)->name; */
   /* We use the new name, as findpackage() may have
      done a tolower for us.
    */
@@ -129,11 +144,10 @@
                enum parsedbflags flags,
                const char *filename, int lno, FILE *↩
                   warnto, int *warncount,
                const char *value, const struct ↩
                   fieldinfo *fip) {
```

```
-  pifp->essential=
-    *value ? convert_string(filename,lno,_("yes/no in ↩
   boolean field"), -1,
-                            warnto,warncount,pigp,
-                            value,booleaninfos,NULL)
-           : 0;
+  if (!*value) return;
+  PKGPFIELD(pifp,fip->integer,int)=
+       convert_string(filename,lno,_("yes/no in ↩
   boolean field"),
+                                  -1,warnto,warncount,↩
   pigp,value,booleaninfos,NULL);
 }

 void f_section(struct pkginfo *pigp, struct ↩
    pkginfoperfile *pifp,


--- orig/lib/parse.c
+++ mod/lib/parse.c
@@ -69,6 +69,7 @@
   { "MD5sum",           f_filecharf,       w_filecharf↩
      ,      FILEFOFF(md5sum)         },
   { "MSDOS-Filename",   f_filecharf,       w_filecharf↩
      ,      FILEFOFF(msdosname)      },
   { "Description",      f_charfield,       w_charfield↩
      ,      PKGIFPOFF(description)   },
+  { "Multi-Arch",       f_boolean,         ↩
   w_booleandefno,   PKGIFPOFF(multiarch)     },
   /* Note that aliases are added to the nicknames ↩
      table in parsehelp.c. */
   {  NULL   /* sentinel - tells code that list is ↩
      ended */                                  }
 };
@@ -98,6 +99,7 @@
   int fieldlen= 0, valuelen= 0;
   int *ip, c;
   struct stat stat;
+  const char *tmp_name;

   if (warncount) *warncount= 0;
   newpifp= (flags & pdb_recordavailable) ? &newpig.↩
      available : &newpig.installed;
@@ -194,6 +196,7 @@
        fieldstart= nick->canon;
        fieldlen= strlen(fieldstart);
       }
+/* @@@ Start of field loading code. Everything above ↩
   here is basic parsing */
```

```
       for (fip= fieldinfos, ip= fieldencountered;
            fip->name && strncasecmp(fieldstart,fip->↩
               name, fieldlen);
            fip++, ip++);
@@ -224,6 +227,17 @@
       }
       if (EOF_mmap(dataptr, endptr) || c == '\n' || c ↩
          == MSDOS_EOF_CHAR) break;
     } /* loop per field */
+/* @@@ Done loading the package info now */
+       /* @@@ Check to see if we have a multiarch ↩
   package. If so, modify
+        * its name */
+       if (newpifp->multiarch) {
+               tmp_name= newpig.name;
+               newpig.name= m_malloc(strlen(newpig.↩
   name) + strlen(newpifp->architecture) + 2);
+               strcpy(newpig.name, tmp_name);
+               strcat(newpig.name, ":");
+               strcat(newpig.name, newpifp->↩
   architecture);
+       }
+
     if (pdone && donep)
       parseerr(NULL,filename,lno, warnto,warncount,&↩
          newpig,0,
               _("several package info entries found, ↩
                  only one allowed"));
@@ -270,9 +284,26 @@
       newpifp->conffiles= NULL;
     }

+/*
+       if(newpifp->multiarch || strncmp(newpig.name, "↩
   libfoo", 6) == 0) {
+               fprintf(stderr, "Found an interesting ↩
   package: %s %s %d\n",
+                                   newpig.name,
+                                   newpifp->architecture,
+                                   newpifp->multiarch);
+               if(flags & pdb_recordavailable)
+                       fprintf(stderr, "  Recording as↩
    available\n");
+               else
+                       fprintf(stderr, "  Recording as↩
    installed?\n");
+       }
+*/
+
```

```
     pigp= findpackage(newpig.name);
     pifp= (flags & pdb_recordavailable) ? &pigp->↩
        available : &pigp->installed;
     if (!pifp->valid) blankpackageperfile(pifp);
+       /* @@@ Throw away the old temporary name we ↩
   were using and rewrite
+        * to use the one that findpackage gave us */
+       free(newpig.name);
+       newpig.name= pigp->name;

     /* Copy the priority and section across, but don't↩
         overwrite existing
      * values if the pdb_weakclassification flag is ↩
         set.
@@ -313,7 +344,7 @@
     pdone++;
     if (EOF_mmap(dataptr, endptr)) break;
     if (c == '\n') lno++;
-  }
+  } /* Loop per package */
   pop_cleanup(0);
 #ifdef HAVE_MMAP
   munmap(data, stat.st_size);


--- orig/src/archives.c
+++ mod/src/archives.c
@@ -235,6 +235,7 @@
                            struct pkginfoperfile *↩
                               newpifp,
                            struct pkginfo *oldpigp) {
   struct dependency *dep;
+  int len;

   debug(dbg_depcon,"does_replace new=%s old=%s (%s)",↩
      newpigp->name,
         oldpigp->name,versiondescribe(&oldpigp->↩
            installed.version,
@@ -247,6 +248,18 @@
     debug(dbg_depcon,"does_replace ... yes");
     return 1;
   }
+
+  /* @@@ Implicit replaces between multiarch and ↩
   uniarch packages with
+   * the same name. Nasty hackery here. */
+  if (newpifp->multiarch) {
+         len = strcspn(newpigp->name, ":");
```

```
+         if (strncmp(newpigp->name, oldpigp->name, len↩
    ) == 0
+                 && oldpigp->name[len] == 0) {
+                 debug(dbg_depcon,"does_replace ... ↩
    yes (implicit multiarch)");
+                 return 1;
+         }
+  }
+
    debug(dbg_depcon,"does_replace ... no");
    return 0;
 }
@@ -458,16 +471,50 @@
             forcibleerr(fc_overwritedir, _("trying to ↩
                overwrite directory ‘%.250s’ "
                         "in package %.250s with ↩
                            nondirectory"),
                         nifd->namenode->name,otherpkg-↩
                            >name);
-          } else {
-            /* WTA: At this point we are replacing ↩
    something without a Replaces.
-            * if the new object is a directory and the↩
     previous object does not
-            * exist assume it’s also a directory and ↩
    don’t complain
-            */
-            if (! (statr && ti->Type==Directory))
-              forcibleerr(fc_overwrite,
-                          _("trying to overwrite ‘%.250s↩
    ’, which is also in package %.250s"),
-                          nifd->namenode->name,otherpkg-↩
    >name);
+                        continue;
+                 }
+                 /* If both packages are multiarch, ↩
    and we’re replacing a link
+                  * with a link */
+                 if (tc->pkg->available.multiarch && ↩
    otherpkg->installed.multiarch
+                          && !statr && S_ISLNK(stab.↩
    st_mode) && ti->Type == SymbolicLink) {
+                        /* If both packages have the ↩
    same name, modulo architecture */
+                        if(strncmp(tc->pkg->name,
+                                           otherpkg->↩
    name,
+                                           strcspn(tc->↩
    pkg->name, ":")
```

```
+                                                ) == 0) {
+                           /* Bah. readlink() doesn't ↩
    tell us how much space it
+                            * actually needs, so keep ↩
    trying until it works */
+                           int bufsize= 250;
+                           char *buf= ↩
    m_malloc(sizeof(char)*bufsize);
+                           int sizeread= ↩
    readlink(fnamevb.buf, buf, bufsize);
+
+                           while(sizeread == bufsize) {
+                                 bufsize *= 2;
+                                 buf= m_realloc(buf, ↩
    sizeof(char)*bufsize);
+                                 sizeread= ↩
    readlink(fnamevb.buf, buf, bufsize);
+                           }
+
+                           if(sizeread == -1) {
+                                 perror("readlink() ↩
    problem: ");
+                                 ohshite("Had trouble ↩
    reading link %.250s which previous experience ↩
    suggests existed not 100 lines above", otherpkg->↩
    name);
+                           }
+
+                           /* If both links point to the↩
     same place */
+                           if(strncmp(buf, ti->LinkName,↩
     sizeread) == 0) {
+                                 free(buf);
+                                 continue;
+                           }
+                           free(buf);
+                         }
          }
+                 /* WTA: At this point we are ↩
    replacing something without a Replaces.
+                  * if the new object is a directory ↩
    and the previous object does not
+                  * exist assume it's also a directory↩
     and don't complain
+                  */
+                 if (! (statr && ti->Type==Directory))
+                       forcibleerr(fc_overwrite,
+                      _("trying to overwrite `%.250s',↩
     which is also in package %.250s"),
```

```
+                       nifd->namenode->name,otherpkg->↩
   name);
         }
       }
     }


--- orig/src/main.c
+++ mod/src/main.c
@@ -547,6 +547,8 @@
   jmp_buf ejbuf;
   static void (*actionfunction)(const char *const *↩
      argv);

+  fprintf(stderr, "EXPERIMENTAL BIARCH DPKG 2004-06-01↩
   .01\n");
+
   standard_startup(&ejbuf, argc, &argv, DPKG, 1, ↩
      cmdinfos);
   if (!cipaction) badusage(_("need an action option"))↩
      ;



--- orig/src/processarc.c
+++ mod/src/processarc.c
@@ -84,6 +84,8 @@
   struct stat stab;
   struct packageinlist *deconpil, *deconpiltemp;

+  char *truncname;
+
   cleanup_pkg_failed= cleanup_conflictor_failed= 0;
   admindirlen= strlen(admindir);

@@ -190,6 +192,10 @@
   waitsubproc(c1,BACKEND " --control",0);
   strcpy(cidirrest,CONTROLFILE);

+/*
+  fprintf(stderr, "processarc.c: Loading package ↩
   details from %s\n", cidir);
+*/
+
   parsedb(cidir, pdb_recordavailable|pdb_rejectstatus|↩
      pdb_ignorefiles|pdb_weakclassification,
           &pkg,NULL,NULL);
   if (!pkg->files) {
@@ -208,8 +214,7 @@
```

```
   }

   if (pkg->available.architecture && *pkg->available.↩
      architecture &&
-       strcmp(pkg->available.architecture,"all") &&
-       strcmp(pkg->available.architecture,architecture)↩
   )
+         ! is_allowable_arch(pkg->available.↩
   architecture))
     forcibleerr(fc_architecture,
                 _("package architecture (%s) does not ↩
                    match system (%s)"),
                 pkg->available.architecture,↩
                    architecture);
@@ -217,6 +222,11 @@
   if (!pkg->installed.valid) blankpackageperfile(&pkg-↩
      >installed);
   assert(pkg->available.valid);

+/*
+  fprintf(stderr, "  Package %s has multiarch flags %d↩
   ,%d\n",
+                  pkg->name, pkg->available.multiarch, ↩
   pkg->installed.multiarch);
+*/
+
   for (deconpil= deconfigure;
        deconpil;
        deconpil= deconpiltemp) {
@@ -275,6 +285,7 @@
     if (psearch->up->type != dep_conflicts) continue;
     check_conflict(psearch->up, pkg, pfilename);
   }
+  /* @@@ Implied conflicts checks here? */

   ensure_allinstfiles_available();
   filesdbinit();
@@ -809,6 +820,7 @@
   pkg->installed.version= pkg->available.version;
   pkg->installed.origin = pkg->available.origin;
   pkg->installed.bugs = pkg->available.bugs;
+  pkg->installed.multiarch = pkg->available.multiarch;

   /* We have to generate our own conffiles structure. ↩
      */
   pkg->installed.conffiles= 0; iconffileslastp= &pkg->↩
      installed.conffiles;
```

```
--- /dev/null
+++ mod/lib/multi-arch.c
@@ -0,0 +1,46 @@
+/*
+ * libdpkg - Debian packaging suite library routines
+ * multi-arch.c - support for the Multi-Arch: ↩
   extensions
+ *
+ * Copyright (C) 2004 Hugo Mills <hugo@carfax.org.uk>
+ *
+ * This is free software; you can redistribute it and/↩
   or modify
+ * it under the terms of the GNU General Public ↩
   License as
+ * published by the Free Software Foundation; either ↩
   version 2,
+ * or (at your option) any later version.
+ *
+ * This is distributed in the hope that it will be ↩
   useful, but
+ * WITHOUT ANY WARRANTY; without even the implied ↩
   warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE↩
   .  See the
+ * GNU General Public License for more details.
+ *
+ * You should have received a copy of the GNU General ↩
   Public
+ * License along with dpkg; if not, write to the Free ↩
   Software
+ * Foundation, Inc., 675 Mass Ave, Cambridge, MA ↩
   02139, USA.
+ */
+
+#include <config.h>
+
+#include <string.h>
+#include <stdlib.h>
+
+#include "dpkg.h"
+
+/* List of acceptable architectures */
+const char *architectures[] = ARCH_LIST;
+
+int is_allowable_arch(char *archtest)
+{
+  const char **ptr;
+
```

```
+  if(strcmp(archtest,"all") == 0)
+       return 1;
+
+  ptr = architectures;
+  while(*ptr) {
+       if(strcmp(archtest,*ptr) == 0) return 1;
+       ptr++;
+  }
+
+  return 0;
+}
--- /dev/null
+++ mod/subarchtable
@@ -0,0 +1,2 @@
+amd64  { "amd64", "i386", NULL }
+i386   { "amd64", "i386", NULL }
```

A.5 pkg-config

```
--- orig/debian/changelog
+++ mod/debian/changelog
@@ -1,3 +1,9 @@
+pkgconfig (0.15.0-4multiarch1) multiarch; urgency=low
+
+  * Multiarchify
+
+ -- Tollef Fog Heen <tfheen@debian.org>  Wed, 16 Feb ↩
   2005 13:40:10 +0100
+
 pkgconfig (0.15.0-4) unstable; urgency=low

   * New Maintainer.

--- orig/debian/rules
+++ mod/debian/rules
@@ -9,7 +9,7 @@
 export DH_COMPAT=4


-CFLAGS = -Wall -g
+CFLAGS = -Wall -g -DSYSTEM_SEARCH_PATH="\"/usr/local/↩
   lib/pkgconfig:/usr/lib/$(DEB_BUILD_GNU_TYPE)/↩
   pkgconfig\""
 ifneq (,$(findstring noopt,$(DEB_BUILD_OPTIONS)))
       CFLAGS += -O0
 else
@@ -35,8 +35,8 @@
```

```
        ./configure --prefix=/usr --mandir=\$${prefix}/↩
           share/man \
                --infodir=\$${prefix}/share/info ↩
                   --disable-shared \
-               $(confflags) CFLAGS="$(CFLAGS)"
-       $(MAKE)
+               $(confflags) CFLAGS='$(CFLAGS)'
+       $(MAKE)

        touch build-stamp



--- orig/pkg.c
+++ mod/pkg.c
@@ -51,6 +51,10 @@
 #define PKGLIBDIR g_strconcat ↩
    (g_win32_get_package_installation_directory ↩
    (PACKAGE, NULL), "\\lib\\pkgconfig", NULL)
 #endif

+#ifndef SYSTEM_SEARCH_PATH
+#define SYSTEM_SEARCH_PATH ""
+#endif
+
 static void verify_package (Package *pkg);

 static GHashTable *packages = NULL;
@@ -188,11 +192,24 @@
 {
   static gboolean initted = FALSE;
   const char *pkglibdir;
+  char **system_search_dirs;
+  char **iter;

   pkglibdir = g_getenv ("PKG_CONFIG_LIBDIR");
   if (pkglibdir == NULL)
     pkglibdir = PKGLIBDIR;

+  system_search_dirs = g_strsplit (SYSTEM_SEARCH_PATH,↩
    G_SEARCHPATH_SEPARATOR_S, -1);
+  iter = system_search_dirs;
+  while (*iter)
+    {
+      debug_spew ("Adding directory '%s' from default ↩
   system search path\n",
+                  *iter);
+      add_search_dir (*iter);
+      ++iter;
```

```
+    }
+  g_strfreev (system_search_dirs);
+
   if (!initted)
     {
       initted = TRUE;
@@ -202,9 +219,7 @@
       path_positions = g_hash_table_new (g_str_hash, ↩
          g_str_equal);

       g_slist_foreach (search_dirs, (GFunc)scan_dir, ↩
          NULL);
-#ifdef DEBIAN
-      scan_dir ("/usr/local/lib/pkgconfig");
-#endif
+
       scan_dir (pkglibdir);
     }
 }
```

A.6 libogg

```
--- orig/debian/changelog
+++ mod/debian/changelog
@@ -1,3 +1,27 @@
+libogg (1.1.0-1multiarch4) multiarch; urgency=low
+
+  * Remove libogg0-dev.docs as well
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Thu, 10 Feb ↩
   2005 13:19:12 +0100
+
+libogg (1.1.0-1multiarch3) multiarch; urgency=low
+
+  * Make sure to install symlinks to -common in their ↩
   respective packages.
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Thu, 10 Feb ↩
   2005 13:19:12 +0100
+
+libogg (1.1.0-1multiarch2) multiarch; urgency=low
+
+  * Remove libogg0.README.Debian in clean, since it's ↩
   in the orig.tar.gz
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Thu, 10 Feb ↩
   2005 13:19:12 +0100
+
+libogg (1.1.0-1multiarch1) multiarch; urgency=low
```

```
+
+  * Multiarch build
+
+ -- Tollef Fog Heen <tfheen@debian.org>  Thu, 10 Feb ↩
   2005 13:19:12 +0100
+
 libogg (1.1.0-1) unstable; urgency=low

   * New upstream.
@@ -57,6 +81,4 @@

  -- Christopher L Cheney <ccheney@debian.org>  Sun, 29↩
      Oct 2000 01:11:57 -0500

-Local variables:
-mode: debian-changelog
-End:
+

--- orig/debian/control
+++ mod/debian/control
@@ -8,7 +8,7 @@
 Package: libogg0
 Architecture: any
 Section: libs
-Depends: ${shlibs:Depends}
+Depends: ${shlibs:Depends}, libogg0-common (= ${↩
   Source-Version})
 Description: Ogg Bitstream Library
  Libogg is a library for manipulating ogg bitstreams. ↩
      It handles
  both making ogg bitstreams and getting packets from ↩
     ogg bitstreams.
@@ -20,3 +20,22 @@
 Description: Ogg Bitstream Library Development
  The libogg-dev package contains the header files and ↩
     documentation
  needed to develop applications with libogg.
+
+Package: libogg0-common
+Architecture: any
+Section: libs
+Depends: ${shlibs:Depends}
+Description: Ogg Bitstream Library - common files
+ Libogg is a library for manipulating ogg bitstreams. ↩
    It handles
+ both making ogg bitstreams and getting packets from ↩
   ogg bitstreams.
+ .
```

```
+ This package contains the arch-independent parts of ↩
   libogg0
+
+Package: libogg-dev-common
+Architecture: any
+Section: libdevel
+Description: Ogg Bitstream Library Development
+ The libogg-dev package contains the header files and ↩
   documentation
+ needed to develop applications with libogg.
+ .
+ This package contains the arch-independent parts of ↩
   libogg-dev

--- orig/debian/libogg-dev.install
+++ mod/debian/libogg-dev.install
@@ -1,8 +1,7 @@
-debian/tmp/usr/include/ogg/config_types.h
-debian/tmp/usr/include/ogg/ogg.h
-debian/tmp/usr/include/ogg/os_types.h
-debian/tmp/usr/lib/libogg.a
-debian/tmp/usr/lib/libogg.la
-debian/tmp/usr/lib/libogg.so
-debian/tmp/usr/lib/pkgconfig/ogg.pc
-debian/tmp/usr/share/aclocal/ogg.m4
+debian/tmp/usr/include/*/ogg/config_types.h
+debian/tmp/usr/include/*/ogg/ogg.h
+debian/tmp/usr/include/*/ogg/os_types.h
+debian/tmp/usr/lib/*/libogg.a
+debian/tmp/usr/lib/*/libogg.la
+debian/tmp/usr/lib/*/libogg.so
+debian/tmp/usr/lib/*/pkgconfig/ogg.pc

--- orig/debian/libogg0.install
+++ mod/debian/libogg0.install
@@ -1 +1 @@
-debian/tmp/usr/lib/libogg.so.*
+debian/tmp/usr/lib/*/libogg.so.*

--- orig/debian/rules
+++ mod/debian/rules
@@ -43,7 +43,9 @@
         # change ../configure to ../autogen.sh for CVS ↩
            build
         cd $(objdir) && \
         ../configure --build=$(DEB_BUILD_GNU_TYPE) ↩
            --host=$(DEB_HOST_GNU_TYPE) \
-        --prefix=/usr --enable-static
+        --prefix=/usr --enable-static \
```

```
+       --libdir=/usr/lib/$(DEB_BUILD_GNU_TYPE) \
+       --includedir=/usr/include/$(DEB_BUILD_GNU_TYPE)

        touch configure-stamp

@@ -78,7 +80,8 @@
 clean:
        dh_testdir
        dh_testroot
-       rm -f build-arch-stamp build-indep-stamp ↩
    configure-stamp
+       rm -f build-arch-stamp build-indep-stamp ↩
    configure-stamp \
+           debian/libogg0.README.Debian debian/↩
    libogg-dev.docs

        # Remove build tree
        rm -rf $(objdir)
@@ -113,6 +116,8 @@
        $(MAKE) install DESTDIR=$(CURDIR)/debian/tmp

        dh_install -s --list-missing
+       dh_link -plibogg0 usr/share/doc/libogg0-common ↩
    usr/share/doc/libogg0
+       dh_link -plibogg-dev usr/share/doc/↩
    libogg-dev-common usr/share/doc/libogg-dev

 # Must not depend on anything. This is to be called by
 # binary-arch/binary-indep


--- /dev/null
+++ mod/debian/libogg-dev-common.install
@@ -0,0 +1 @@
+debian/tmp/usr/share/aclocal/ogg.m4
```

A.7 libvorbis

```
--- orig/debian/changelog
+++ mod/debian/changelog
@@ -1,3 +1,21 @@
+libvorbis (1.0.1-1multiarch3) multiarch; urgency=low
+
+  * Bumpbump.
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Thu, 24 Feb ↩
    2005 12:14:45 +0100
+
+libvorbis (1.0.1-1multiarch2) multiarch; urgency=low
```

```
+
+  * Make sure we ship the include files.
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Mon, 21 Feb ↩
   2005 12:35:43 +0100
+
+libvorbis (1.0.1-1multiarch1) multiarch; urgency=low
+
+  * Multiarch build
+
+ -- Tollef Fog Heen <tfheen@idi.ntnu.no>  Thu, 10 Feb ↩
   2005 16:11:57 +0100
+
 libvorbis (1.0.1-1) unstable; urgency=low

   * New upstream.
@@ -105,6 +123,4 @@

  -- Michael Beattie <mickyb@es.co.nz>  Mon, 26 Jun ↩
     2000 18:59:56 +1200

-Local variables:
-mode: debian-changelog
-End:
+

--- orig/debian/libvorbis-dev.install
+++ mod/debian/libvorbis-dev.install
@@ -1,16 +1,16 @@
-debian/tmp/usr/include/vorbis/codec.h
-debian/tmp/usr/include/vorbis/vorbisenc.h
-debian/tmp/usr/include/vorbis/vorbisfile.h
-debian/tmp/usr/lib/libvorbis.a
-debian/tmp/usr/lib/libvorbis.la
-debian/tmp/usr/lib/libvorbis.so
-debian/tmp/usr/lib/libvorbisenc.a
-debian/tmp/usr/lib/libvorbisenc.la
-debian/tmp/usr/lib/libvorbisenc.so
-debian/tmp/usr/lib/libvorbisfile.a
-debian/tmp/usr/lib/libvorbisfile.la
-debian/tmp/usr/lib/libvorbisfile.so
-debian/tmp/usr/lib/pkgconfig/vorbis.pc
-debian/tmp/usr/lib/pkgconfig/vorbisenc.pc
-debian/tmp/usr/lib/pkgconfig/vorbisfile.pc
-debian/tmp/usr/share/aclocal/vorbis.m4
+debian/tmp/usr/include/*/vorbis/codec.h
+debian/tmp/usr/include/*/vorbis/vorbisenc.h
+debian/tmp/usr/include/*/vorbis/vorbisfile.h
+debian/tmp/usr/lib/*/libvorbis.a
```

```
+debian/tmp/usr/lib/*/libvorbis.la
+debian/tmp/usr/lib/*/libvorbis.so
+debian/tmp/usr/lib/*/libvorbisenc.a
+debian/tmp/usr/lib/*/libvorbisenc.la
+debian/tmp/usr/lib/*/libvorbisenc.so
+debian/tmp/usr/lib/*/libvorbisfile.a
+debian/tmp/usr/lib/*/libvorbisfile.la
+debian/tmp/usr/lib/*/libvorbisfile.so
+debian/tmp/usr/lib/*/pkgconfig/vorbis.pc
+debian/tmp/usr/lib/*/pkgconfig/vorbisenc.pc
+debian/tmp/usr/lib/*/pkgconfig/vorbisfile.pc
+debian/tmp/usr/share/aclocal/vorbis.m4  usr/share/↩
   aclocal-test


--- orig/debian/libvorbis0a.install
+++ mod/debian/libvorbis0a.install
@@ -1 +1 @@
-debian/tmp/usr/lib/libvorbis.so.*
+debian/tmp/usr/lib/*/libvorbis.so.*


--- orig/debian/libvorbisenc2.install
+++ mod/debian/libvorbisenc2.install
@@ -1 +1 @@
-debian/tmp/usr/lib/libvorbisenc.so.*
+debian/tmp/usr/lib/*/libvorbisenc.so.*


--- orig/debian/libvorbisfile3.install
+++ mod/debian/libvorbisfile3.install
@@ -1 +1 @@
-debian/tmp/usr/lib/libvorbisfile.so.*
+debian/tmp/usr/lib/*/libvorbisfile.so.*


--- orig/debian/rules
+++ mod/debian/rules
@@ -43,7 +43,9 @@
        # change ../configure to ../autogen.sh for CVS ↩
           build
        cd $(objdir) && \
        ../configure --build=$(DEB_BUILD_GNU_TYPE) ↩
           --host=$(DEB_HOST_GNU_TYPE) \
-       --prefix=/usr --enable-static
+       --prefix=/usr --enable-static \
+       --libdir=/usr/lib/$(DEB_BUILD_GNU_TYPE) \
+       --includedir=/usr/include/$(DEB_BUILD_GNU_TYPE)
```

```
        touch configure-stamp

@@ -91,7 +93,7 @@
        #if test -d CVS; then \
                $(MAKE) cvs-clean ;\
        fi
-
+       rm -f debian/libvorbis-dev.docs debian/↩
   libvorbis-dev.examples
        dh_clean

 install: install-indep install-arch


--- orig/include/vorbis/Makefile.am
+++ mod/include/vorbis/Makefile.am
@@ -2,8 +2,6 @@

 AUTOMAKE_OPTIONS = foreign

-includedir = $(prefix)/include/vorbis
-
-include_HEADERS = codec.h vorbisfile.h vorbisenc.h
-
+pkgincludedir = $(includedir)/vorbis
+pkginclude_HEADERS = codec.h vorbisfile.h vorbisenc.h



--- /dev/null
+++ mod/debian/libvorbis-dev-common.install
@@ -0,0 +1,2 @@
+
+debian/tmp/usr/share/aclocal/vorbis.m4
```

A.8 alsaplayer

```
--- orig/debian/alsaplayer-alsa.install
+++ mod/debian/alsaplayer-alsa.install
@@ -1 +1 @@
-usr/lib/alsaplayer/output/libalsa_out.so
+usr/lib/*/alsaplayer/output/libalsa_out.so

--- orig/debian/alsaplayer-common.install
+++ mod/debian/alsaplayer-common.install
@@ -1,5 +1,5 @@
 usr/bin/alsaplayer
-usr/lib/alsaplayer/input/*.so
-usr/lib/alsaplayer/output/libnull_out.so
```

```
-usr/lib/alsaplayer/reader/*.so
+usr/lib/*/alsaplayer/input/*.so
+usr/lib/*/alsaplayer/output/libnull_out.so
+usr/lib/*/alsaplayer/reader/*.so
 usr/share/man/man1/alsaplayer.1

--- orig/debian/alsaplayer-daemon.install
+++ mod/debian/alsaplayer-daemon.install
@@ -1,3 +1,3 @@
-usr/lib/alsaplayer/interface/libdaemon_interface.so
+usr/lib/*/alsaplayer/interface/libdaemon_interface.so

--- orig/debian/alsaplayer-esd.install
+++ mod/debian/alsaplayer-esd.install
@@ -1 +1 @@
-usr/lib/alsaplayer/output/libesound_out.so
+usr/lib/*/alsaplayer/output/libesound_out.so

--- orig/debian/alsaplayer-gtk.install
+++ mod/debian/alsaplayer-gtk.install
@@ -1,2 +1,2 @@
-usr/lib/alsaplayer/scopes/*.so
-usr/lib/alsaplayer/interface/libgtk_interface.so
+usr/lib/*/alsaplayer/scopes/*.so
+usr/lib/*/alsaplayer/interface/libgtk_interface.so

--- orig/debian/alsaplayer-jack.install
+++ mod/debian/alsaplayer-jack.install
@@ -1 +1 @@
-usr/lib/alsaplayer/output/libjack_out.so
+usr/lib/*/alsaplayer/output/libjack_out.so

--- orig/debian/alsaplayer-nas.install
+++ mod/debian/alsaplayer-nas.install
@@ -1 +1 @@
-usr/lib/alsaplayer/output/libnas_out.so
+usr/lib/*/alsaplayer/output/libnas_out.so

--- orig/debian/alsaplayer-oss.install
+++ mod/debian/alsaplayer-oss.install
@@ -1 +1 @@
-usr/lib/alsaplayer/output/liboss_out.so
+usr/lib/*/alsaplayer/output/liboss_out.so

--- orig/debian/alsaplayer-text.install
+++ mod/debian/alsaplayer-text.install
@@ -1,2 +1,2 @@
-usr/lib/alsaplayer/interface/libtext_interface.so
+usr/lib/*/alsaplayer/interface/libtext_interface.so
```

```
--- orig/debian/alsaplayer-xosd.install
+++ mod/debian/alsaplayer-xosd.install
@@ -1,2 +1,2 @@
-usr/lib/alsaplayer/interface/libxosd_interface.so
+usr/lib/*/alsaplayer/interface/libxosd_interface.so

--- orig/debian/changelog
+++ mod/debian/changelog
@@ -1,3 +1,9 @@
+alsaplayer (0.99.76-0.3multiarch1) multiarch; urgency=↩
    low
+
+  * Multiarch build
+
+ -- Tollef Fog Heen <tfheen@debian.org>  Mon, 14 Feb ↩
    2005 15:46:37 +0100
+
 alsaplayer (0.99.76-0.3) unstable; urgency=low

    * Non-Maintainer Upload

--- orig/debian/libalsaplayer-dev.install
+++ mod/debian/libalsaplayer-dev.install
@@ -1,6 +1,6 @@
-usr/include/alsaplayer/*.h
-usr/lib/libalsaplayer.a
-usr/lib/libalsaplayer.la
-usr/lib/libalsaplayer.so
-usr/lib/pkgconfig/alsaplayer.pc
+usr/include/*/alsaplayer/*.h
+usr/lib/*/libalsaplayer.a
+usr/lib/*/libalsaplayer.la
+usr/lib/*/libalsaplayer.so
+usr/lib/*/pkgconfig/alsaplayer.pc
 usr/share/doc/alsaplayer/reference

--- orig/debian/libalsaplayer0.install
+++ mod/debian/libalsaplayer0.install
@@ -1,2 +1,2 @@
-usr/lib/libalsaplayer.so.0
-usr/lib/libalsaplayer.so.0.0.2
+usr/lib/*/libalsaplayer.so.0
+usr/lib/*/libalsaplayer.so.0.0.2

--- orig/debian/rules
+++ mod/debian/rules
@@ -2,6 +2,11 @@
```

```
 PWD=$(shell pwd)

+# These are used for cross-compiling and for saving ↩
   the configure script
+# from having to guess our platform (since we know it ↩
   already)
+DEB_HOST_GNU_TYPE      ?= $(shell dpkg-architecture ↩
   -qDEB_HOST_GNU_TYPE)
+DEB_BUILD_GNU_TYPE     ?= $(shell dpkg-architecture ↩
   -qDEB_BUILD_GNU_TYPE)
+
 configure-stamp:
        dh_testdir
        mkdir -p build
@@ -13,7 +18,9 @@
                    --enable-jack \
                    --cache-file=../config.cache \
                    --enable-oggvorbis=yes ↩
                       --enable-audiofile \
-                   --enable-static --enable-shared
+                   --enable-static --enable-shared \
+                   --libdir=/usr/lib/↩
   $(DEB_BUILD_GNU_TYPE) \
+                   --includedir=/usr/include/↩
   $(DEB_BUILD_GNU_TYPE)
        touch configure-stamp

 maintainerclean: clean

* added files
```