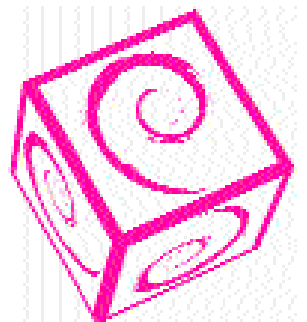
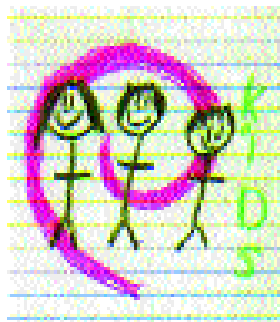
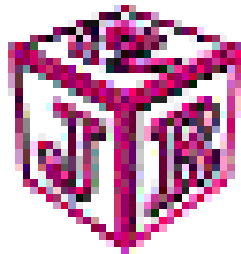
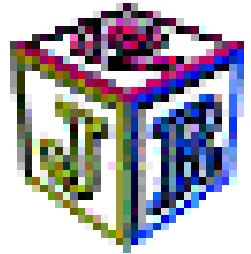
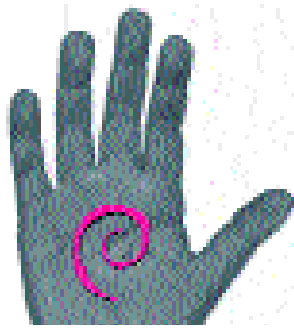


From tiny acorns grows

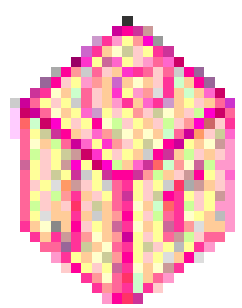
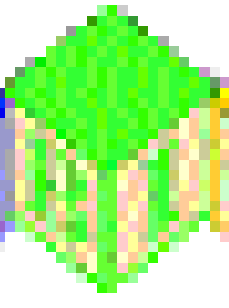
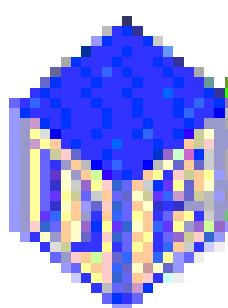
THE GNU GENERATION

RICHARD SMEDLEY

Having looked at individual kids' packages in previous issues, let's see how to get everything on your system at once. We introduce the Debian Jr project and interview Ben Armstrong, the lead developer.



The proposed logos for Debian Jr



Common Ground

Many people are drawn to GNU/Linux by the community support. The help and sharing of knowledge make learning new ways of using a computer enjoyable.

Most Linux distributions are put together by commercial enterprises, or by small groups of individuals. Debian is unique in being actively developed by a worldwide community of over 500 developers acting in chaotic co-operation, on the agreed common ground of the Debian Social Contract. This document supports the open development of a totally Free Operating System, of the best possible quality.

It works extremely well, as Debian GNU/Linux provides a stable OS found on many servers and a cutting edge unstable branch which graces the desktop of many a developer. Yet the notorious installation aside, Debian is not too difficult a distribution for most users. Debian Jr's wider aim is to make the distribution a better experience for users aged two-99.

The primary goal of the Debian Jr project is to make Debian an OS our children *want* to run, initially focussing on children aged two to eight. At this age a child will have an older sibling or adult acting as sysadmin (systems administrator) for them. It is at their technically competent sysadmin that a large part of the initial release is aimed, giving them a starting point for a children's system. However, once Debian 3.0 is released, and users start to provide feedback on Debian Jr., much more can be done. You can download the project now if you run a Debian (or Progeny or Stormix) system by including Woody (testing) in your `/etc/apt/sources.list` and installing the junior packages. When Woody is released you can simply select Debian Jr. in the `tasksel` menu during installation.

Birth of a project

Linux Magazine How did the project get started? You were surely aware that a GNU/Linux distribution for two-99 year olds was a *vast* thing to take on – were you feeling particularly bloody-minded, or did

you start with a thought "my children need such-and-such" and then the project just happened?

Ben Armstrong Precisely. Nobody had any grandiose plans. There was just this thought that with no fewer than seven computers in the house running Debian, and all of my children already using them, we had already a start at using Debian as a base and tailoring it to suit their needs.

I saw that we were relying on serendipity to supply us with more packages suitable for the children. Yet I knew there must be more out there that could be packaged for Debian. Also, it was plain to see that any integration issues that had particularly to do with children were not going to happen by accident. And finally, I realized it must not be entirely obvious to a new Debian user which packages would be good for the kids and which would not.

I asked some other Debian developers about it, and was encouraged by them to go and start something. Up until that point I was thinking "we" should do this and "we" should do that and hadn't

[left]
Two computers are better than one

[right]
Maria, aged nine, plays Rocks'N'Diamonds



really considered where I fitted into it. But it made sense for me to make a commitment, so I kicked off the discussion on the developer's mailing list. This got some favourable responses and got people thinking about what such a thing would look like. Then it was just a matter of getting our own mailing list and website set up and try to figure out where to go next. And so the project was born.

Debian and children

LM What level of support have you had from Debian developers for this project?

BA I have encountered plenty of Debian developers (and not just those with children) who think this is a great idea. We have received assistance from those not working directly on Debian Jr whenever we have needed it. I have been surprised to have some of these people approaching me personally to ask "How does this issue affect Debian Jr?" or "What features should go here to make this package child-friendly?" So I think the mere existence of the project is helping Debian developers to consider "...and what about the children?"

LM What about recognition from the wider Free Software community?

BA If you look at our Web page, I list a number of other, related projects. Members of several of these keep in touch with me and continue to support and encourage me. In particular, SEUL/edu is a great help. They keep me inspired and help me to kick-start the project whenever I let it stall out.

LM As a Debian developer you obviously choose to follow the guidelines on Free Software. There are proprietary children's apps that run on Linux that could be included in *non-free* or *contrib*. What are your feelings on the importance of kids' (and everyone's) software being free?

BA Debian Jr is not actively making an effort to package non-free software. Every Debian developer is free, of course, to package anything that can be placed in *contrib* or *non-free*, but I feel that to do so myself would be dissipating my energies into stuff that is off focus for the project. There is no reason we should not be able to shape something entirely out of free software that will appeal to children. I

Childproof is no education

One of the project's aims is to balance childproofing with the goal of allowing children to try things (and yes, break things). Armstrong is very much in favour of exploring and learning by mistakes:

BA Well, if you're an adult looking for the "Linux experience" you are necessarily going to end up tweaking configs and breaking things yourself. It seems only natural to me to share that with my children.

As preposterous as that concept might seem to the uninitiated, I really have found that those who only know how to fix something by removing it and re-installing it, or by rebooting their system get far less out of their system, and are far more dependent on those with more experience for technical help. With a small amount of poking around and exploring things on your own, even if it means breaking it even worse than when you started, there is an opportunity for you to learn how things work. Of course, you need to find a community of helpful people with more experience than yourself to fall back on, but these days

you don't have to go far to find that. You go as far as you can go on your own until you are stuck. When you are truly stuck, you tell your support group what you did and hopefully you get answers that lead you to the solution. The next time you're faced with a similar problem, you are better equipped to deal with it. That's the dynamic I see in my LUG played out over and over again. And the members truly grow in this environment.

Thus children, who need to learn about computers just as adults do, should be given the same opportunity to learn. Instead of shielding them from the computer by hiding all the details inside black boxes, we encourage the children to look at the config files, read the comments, and try different values. They take great pleasure in constantly tweaking their Web browser settings or game settings to suit their mood or just their curiosity. And yes, sometimes this means they break things. Well, that's great. If they tweak a config in a way that the author didn't anticipate and it causes a program to core dump, the child, along with their observant and helpful sysadmin will jump on this opportunity and send a bug report in. The end result is not only a learning

experience for the child, but also more robust free software.

Or at least that's how I envision it. Perhaps that further illuminates my previous objection to making a tight system for non-technical sysadmins. I'm a technologist, and as such, I have slightly different agendas than non-techs. I am raising my children with geek values. I am raising them to learn how to deal with computers when they do the unexpected, and to be comfortable with that and know what to do, even if they don't know how to fix it themselves.

I allow that not all people in the world are geeks or see the value in making things break and improving them. They just want stuff to work, period. When stuff breaks, it is disorienting, irritating, frustrating, and they don't see it as a learning opportunity or software-improvement opportunity at all. If that's where a user is, maybe they should consider a different operating system. Or maybe that user just needs to enlist the help of a technical friend who is there to help when things break. If that friend is also willing and has time to help when his children break things, then maybe, just maybe, he will get along fine with a Debian Jr system.

Woody, Potato, Sid?

Many of you will have noticed that Debian releases are named after characters from the film *Toy Story*.

1.1	Buzz
1.2	Rex
1.3	Bo
2.0	Hamm
2.1	Slink
2.2	Potato
3.0	Woody
unstable	Sid

This is because Bruce Perens (see interview *Linux Magazine 9*), one of the founders of Debian, worked at Pixar Studios for 12 years. The unstable (development) branch of Debian is now permanently called Sid, after the boy next door who destroyed toys.

don't need to re-hash all of the arguments for free versus non-free here. I believe free software is just as important for kids as it is for adults, and for much the same reasons.

Children and technology

LM Where do you see this project in relation to strictly formal education (schools) projects – such as SEUL/edu, OSiE and FSF-edu?

BA Well, the foundation of any formal education is necessarily in the home. True to the vision of "Let's build something our kids will want to use." we are more attuned to what appeals to children than what appeals to educators. Not that the two are necessarily opposed to each other. I would certainly hope that our educators, too, are looking for things that appeal to children. But by focusing on children in the home, we are working with what we know best.

LM Just how important are computers in children's development and why?

BA I think as a geek parent I don't question this nearly as much as I should. We take deliberate steps to ensure our children's minds [are] provided with a universe of books, arts, games, music, food, culture and so forth to explore. And we also take steps to remove things from their environment that we feel impede progress in those areas.

How important are computers in children's development? Well, we could live without them. We could go to the extreme and be complete Luddites, switching off the computers at home and just leaving them for "real work". After all, we have

turfed the TV. Isn't plopping our children in front of a Web browser or a video game every bit as much an impediment to their development as TV? Doesn't it have the same potential to become the default entertainment around the house, soaking up family time and leading us ultimately to a state of technology-assisted disconnection from each other?

Sure, the potential to end up this way is very real. It is a struggle when you're a geek and you already devote great swaths of your leisure time to computers. It takes energy and planning to give computers their proper place in the home, much in the same way that families who have televisions need to put effort into keeping the TV in balance with other activities.

So, where does the computer really fit, after taking these other things into consideration? I think they can be powerful tools to think with. Limitless entire universes of thought can be constructed and explored with them. With guidance, a child's computer can be a great tool for developing their mind. It can be something more than an extremely



Starting Point

As we have mentioned, the project is at an early stage, waiting to build a user base to provide feedback. The range two to eight years is quite diverse, but initial concern is with building the framework. This helps to narrow down the focus to three areas:

- 1. Identifying which packages work well with children: Much has been done here, and you will find user comments on each on the projects packaged page*
- 2. Organizing packages in a way that makes sense to children and sysadmins alike: More work needs to be done on ways of organizing the material makes sense for which age ranges. "It will be when we tackle the menu system that we shift the emphasis to the children. And then the fun begins, because age differences are going to make a big difference to what makes sense*
- 3. Making it easy to install these packages: Meta packages are easy to construct for Debian*

BA *We have deliberately avoided addressing the sub-problem of making Debian painless for non-technical parents to introduce to their kids. Simply put, to assume technical parents who are comfortable with administering a UNIX system makes our job quite a bit easier, as that's who we are ourselves. It's best to stick with what you know. But I envision Debian Jr eventually finding its way into less-technical homes. When we get there, we'll be interested in talking with that sub-group of users and trying to address their needs the best we can.*

Info

Ben Armstrong manages to combine a range of geek interests (including running a LUG, developing for Debian and helping with Xpilot) with a well-balanced home and family life. I think it's something to do with the clean Canadian air.

Debian Jr's homepage: <http://www.debian.org/devell/debian-jr>

mailing list: <http://www.debian.org/MailingLists/subscribe>

packages included: <http://www.debian.org/devell/debian-jr/packaged>

Xpilot: <http://packages.debian.org/stable/games/xpilot.html>

<http://www.xpilot.org>

SEUL/edu: <http://www.seul.org/edu>

For an honest assessment of the pros and cons of Debian:

http://www.debian.org/lintr/why_debian

job to a manageable size, while still providing a very usable *one-point-oh*.

F is for free

LM Free Software is quite an abstract concept for children to understand. My children understand that GNU/Linux is better because "you can share it." What has been your experience of discussing the principles/morality of FS with kids and adults as well as the practical benefits?

BA I think it reduces to "Well, with Linux I can do this, whereas with Windows I cannot." and furnishing examples. This brings the abstract arguments into familiar, concrete terms that children can see more readily. I find myself not having to do this with my children anyway, because all they know in this household is Linux. I cannot even remember the last time we booted to Windows to play a game. When it is time for computer turns (a familiar and highly sought-after reward once chores or homework is done), it is just assumed this means Linux computer turns. There's just so much more now that they can use on Linux that there is no need to supplement it with the (admittedly flashier in some cases, and certainly more polished both in the graphics and sound) small collection of Windows games that remain in our possession.

When it comes to reasoning with adults, I emphasize community first and then the source. Even if they don't get the source argument, they almost always comprehend the community side of things. ■

expensive TV replacement. Let televisions be televisions and game consoles be game consoles. We are building something that goes beyond an entertainment system for kids, addressing each of the aspects of computers that we feel are important for our children's development. I'm not sure we fully grok all of the possibilities yet.

Ready when it's ready

LM What's the timetable and are you ready for the Woody freeze?

BA Within Debian, we are fond of saying "It will be released when it is finished". However, in Debian Jr's case, being a small personal project within Debian, we don't have the clout to say "hold it, we still have this or that to put in". It is kind of like being pregnant. You know that ultimately that baby will come out, but it is up to nature to decide when. The longer she stays inside, the fatter Debian Jr will continue to get. Where this analogy falls apart is, I don't think Debian has ever been known to give birth prematurely. So I remain hopeful that we have time yet to put on a bit of extra fat.

But in more concrete terms, for Woody you will see just a structure for a Debian Jr system. This will be enough for the user to install and use with their children. But we need that user base before we can go much further. I don't like developing in a vacuum. We'll need to have users try this initial offering and start talking to us about what to put in there next. We'll have just a "baby Debian Jr" in Woody. The following release is where we'll see it mature.

For instance, we have deliberately left out support for Gnome and KDE in this release. Instead, we're focusing on the core, by shaping a collection of packages that will work on any system, whether you have Gnome, KDE, or neither. By keeping the first release "light" we don't devote a lot of energy into building structures that we may later find ourselves having to tear apart and rebuild. We also keep the

The author

Richard Smedley juggles a job in IT, magazine work, two eccentric children, cars (which are more interesting than functional) and an overgrown garden.

