Syllable

Syllable News #02 November 2007

The Syllable Newsletter

#2

Inside this magazine:

Kristian Van Der Vliet
Multi-threaded C++, part 2

Bas de Lange
Interview & Syllable User Groups

Kaj de Vos
Interview on Syllable Server & the website aftermath...

Flemming H. Sørensen
Local and National groups part 2

WEB: http://www.syllable-news.info
MAIL: sdn_syllable@yahoo.com.au

Welcome to the SDN!

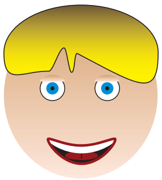
**Whatever your level of expertise, we hope you'll find something interesting to read.
SDN will cover news about Syllable and events when they happen.
There is information for users old and new, and articles for developers of all experience levels.
You'll find interviews with developers and contributors from the Syllable community.
Maybe there will be the odd exclusive...
Just like Syllable, SDN is a community effort. If there is something you would like to see in SDN, you can submit your own articles for publication.**

Thanks to everyone who has contributed to this edition of SDN. We hope you enjoy reading!

Kristian Van Der Vliet

Multi-threaded C++ part 2

In the first part of this series I covered the theory of writing a multi-threaded application and how message passing simplifies things. Now it's time to put theory into practice!



By Kristian Van Der Vliet

Every application for Syllable starts with the Application class. The Application class handles the gory details of setting up communication with the appserver and some other very low level aspects of the API that most application developers never need to worry about. The Application class is derived from the Looper class, so it can receive messages from the appserver. The code for creating an Application is very simple. Most application developers only ever worry about two methods: the constructor, and the Run() method. Below is the absolute simplest Syllable application possible:

```
// simple.cpp : A simple demonstration of the Application class
//
// Compile with "g++ simple.cpp -lsyllable -o simple"
#include <stdlib.h>
#include <iostream>
#include <util/application.h>
// All libsyllable classes live in the "os" namespace
using namespace os;
class SDNApplication : public Application
{
    public:
            SDNApplication( const char *zName );
};
SDNApplication::SDNApplication( const char *zName ) : Application( zName )
{
    std::cerr << "Hello from SDNApplication!" << std::endl;
}
int main( void )
{
    SDNApplication *pcApplication;
    // Note the form of the argument. The text is free-form, but there is a
    // prefered form which we use here.
    pcApplication = new SDNApplication( "application/x-vnd.syllable-SDN" );
    if( pcApplication )
            pcApplication->Run();
    return EXIT_SUCCESS;
}
```

The SDNApplication is derived directly from the Application class, and then instantiated in main(). The Run() method starts the Looper that waits for messages to arrive from the appserver. If you compile and run the application you'll find that once it has printed the message to stderr, the application does not quit. This is because the Application class will loop until it is sent a message telling it to stop: we will cover this later. For now you can interrupt it with Ctrl+C from the Terminal.

One very important point is that the SDNApplication class does not define a destructor, and the code never deletes the SDNApplication instance that is created with new. This is a special property of the Looper class, and any class that is derived from Looper. Because Looper creates and runs itself in a new thread, deleting a Looper instance will leave its thread running, which will almost certainly lead to a crash. Instead the Looper will stop its thread and delete itself when required.

The application currently does not do much, but now we have an Application instance we can use with other classes.
If we want a GUI the application requires a Window.

```
// window.cpp : A simple demonstration of the Window class
//
// Compile with “g++ window.cpp -lsyllable -o window”
#include <stdlib.h>
#include <iostream>
#include <util/application.h>
#include <gui/window.h>
// All libsyllable classes live in the “os” namespace
using namespace os;
class SDNWindow : public Window
{
	public:
		SDNWindow( const Rect &cFrame, const String &cName, const String &cTitle );
		bool OkToQuit( void );
};
SDNWindow::SDNWindow( const Rect &cFrame, const String &cName, const String &cTitle ) : Window( cFrame, cName, cTitle )
{
	// Nothing, yet. Everything is taken care of by the Window class and the appserver
}
bool SDNWindow::OkToQuit( void )
{
	// Tell the Application instance to quit
	Application::GetInstance()->PostMessage( M_QUIT );

	// Let the Looper (the base class of this Window) know it’s OK to quit
	return true;
}
// The SDNApplication just creates an SDNWindow and displays it
class SDNApplication : public Application
{
	public:
		SDNApplication( const char *zName );
	private:
		SDNWindow *m_pcWindow;
};
SDNApplication::SDNApplication( const char *zName ) : Application( zName )
{
	m_pcWindow = new SDNWindow( Rect( 50, 50, 250, 250 ), “sdn_window”, “SDN” );
	if( m_pcWindow )
	{
		m_pcWindow->Show();
		m_pcWindow->MakeFocus();
	}
}
int main( void )
{
	SDNApplication *pcApplication;
	// Note the form of the argument. The text is free-form, but there is a
	// prefered form which we use here.
	pcApplication = new SDNApplication( “application/x-vnd.syllable-SDN” );
	if( pcApplication )
		pcApplication->Run();
	return EXIT_SUCCESS;
}
```

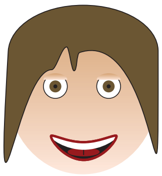
The application now has an SDNWindow class, which is derived from Window. Like the Application class, Window is also derived from Looper. Creating and displaying the Window is all handled by the Window class and the appserver, so the code is very simple. All the Application must do is create a new instance, show the Window and give it focus. Two threads are used, one for the Application instance and one for the Window instance. There is no locking or synchronisation between the two threads.

The OkToQuit() method is a callback which is called when the user clicks on the windows close button. This gives the Window in question the opportunity to confirm that the window may close: normally the method just returns true and the Window closes. The application could also perform other actions here, e.g. displaying a confirmation dialog box that allows the user to cancel the action, or save their changes.

If the user has clicked the close button on the main window, they would normally expect the application to quit. Remember that the Application instance will keep running until it is told to quit, so the OkToQuit() method sends an M_QUIT Message to the Application instance. This will cause the Application to exit cleanly when the window closes. The Application and Window instances will stop their threads and delete themselves. This is a simple example of message passing.

Syllable Server

Since a few weeks there's a Syllable server environment, based on Linux... I just had to ask Kaj, "WHY?". Kaj knows how to explain, another interview.



Kaj de Vos

Why Syllable Server?

I could give lots of different answers to this, but strategically, Server has two purposes. First, we have always planned to build application and network functionality on top of Syllable. Our road map speaks of groupware.
To build that properly, a desktop operating system is not enough. Modern applications are more and more intertwined with networks, so you need servers, too. We could try to make use of existing groupware systems, or build our own applications on existing server systems, but this would lead to similar integration problems that made us decide to develop our own desktop operating system. Software development today has huge problems which are accepted and well documented. We don't have solutions for everything, but we have improved a large number of these issues in Syllable Desktop, and we want to do it with our server and network platform, too.

Second, doing things better necessarily means doing them differently, and doing them later. Obviously, we are behind on mainstream systems, so at some point we need to catch up. That means having people really use our system, and that means offering applications. When we embarked on the Syllable project five years ago, we had hoped and expected that people would gradually step in to build native applications - which is really the point of our design. But the reality is that developers have ignored us. A few people have helped the core developers over the years, and we are grateful for that, but we are no closer to the goal of having an increasing number of native applications. This is all the more frustrating because Syllable itself is shaping up quite nicely, and we have created a rather nice development environment for people, but that leaves us hardly any time to build a thousand applications, too. I strongly suspect that this has happened because during our development, Linux has become "good enough". Until a few years ago, there was a lot of interest in better desktop systems, because Linux was an obvious disaster in that area. Had people put their money where their mouth was, and helped us build applications, they would have had their superior desktop right now. As it stands now, there is a new monoculture of Linux trying to fight the monoculture of Windows... Syllable Server is a tour de force to break this state of things. People want Linux? Fine, we'll give them Linux, if that's what it takes to get them to develop Syllable applications. That may sound unintuitive to you, but the fact is that Syllable is itself the perfect migration path from the established way of developing applications to our way of doing things.

Do you think there is a market/need for another server system?

To this, too, a number of different answers are possible. An easy answer is that Syllable will create its own market. Syllable Desktop is distinctly different from currently popular systems, so there is the potential for a significant market. It can be a better desktop and take a part of that market, but it can also go to some places that other systems can't go, for example because it is much more efficient. Syllable Server is meant to be as similar to Syllable Desktop as is practical, so if Desktop becomes popular, this will create a market for Server to be a companion system. People like having similar systems, because it's easier to manage.

A harder answer to give is how Server will stand on its own. In its current, first release it's just a very basic Linux system, that doesn't have many obvious advantages over other Linux systems yet. But like Server is supposed to support application development for Desktop by offering a solid Linux platform, Desktop will be lending its unique features to Server, as well. Desktop is a very userfriendly system that is achieved through simplicity, and so is Server.
It doesn't show immediately yet, because Server doesn't have Desktop's graphical environment yet. But if you're not afraid of the command line, Server is actually already very easy to install, configure and manage. In this, it is quite different from most Linux systems. It doesn't hide its complexity, but it actually reduces it, making it much easier to really understand and control the system. Once the graphical environment is ported, it will become even easier, without loosing the control.

Many people think that open source and Linux are taking over the world, but a lot of water will flow to the sea before this happens. Linux has been going for a long time, and its growth is actually slowing down. Most research points towards the market share for Linux servers and workstations leveling out at roughly about 15%. Similar patterns exist for open source in general. Desktop Linux is growing now, after all these years, and it's hard to measure its market share, but it still seems to be under one percent. So for starters, there is still a huge opportunity for a better desktop system to jump in and take most of the open source market share. But there's also no fundamental reason that not almost all software, at least standard software, could be open source, so there must be different reasons that it tends to level out at around 15% market share. We think that ease of use, or rather the lack of it, is an important factor in this. There are people who will go through some pain to gain freedom in their software, or to avoid paying for it, but apparently this demographic is limited to about 15% of people. As technophiles, we should realise that while our software means the world to us, it's not the most important thing in other people's lifes. An easier to use system will be able to go beyond this market share. How far, we don't know until we get there.

Why is the Syllable Server based on the Linux kernel?

Again, a number of reasons. Technically, it's logical because Desktop is already based on much of the same software that Linux systems use, from the GNU project and others. Apart from a few details, we made no changes to Desktop to make Server and Desktop as similar as they are. On the other hand, desktop and server duties are quite different. The Desktop kernel was always meant to be optimised for desktop use. It can do server tasks, but it's inefficient at it and not stable enough. Even if we would have the man power to develop it into a good server kernel - which we definitely don't - we wouldn't want to do that, because it would hinder the optimisation for the desktop. Other kernels, such as from BSD systems, don't match our GNU userland as well, although it would be possible to use them with a lot of work.

Strategically, as I said previously, it's very important these days to be Linux, just because everyone focuses on it. Now that we have a Linux, we can get invited to all the penguin balls. The other smokings will look weird at us, but we're used to that. :-) Syllable is on the DistroWatch waiting list now, for example. It's already working for us. Server is special because it derives from Syllable, and Desktop is getting new attention because it has a Linux buddy now.

Do you think the Syllable Server could be faster and safer than Linux, free of virusses etc.?

Partly. As you know, Desktop starts very fast and is very responsive during use. Part of starting fast is through being simple, and this has transfered to Server. Server starts faster than other Linux systems, but it doesn't start as fast as Desktop, because Linux simply doesn't. When our graphical environment is ported to Server, apart from starting much faster than most Linux desktop environments, it will be similarly responsive as on Desktop. It can't be as optimal as on our desktop kernel, but we will have to see how it behaves in practice.

There won't be much difference in security between Server and other Linux systems. Our Linux kernel will be open to the same vulnerabilities as other Linux systems. Of course, Linux is quite secure and virus-free. It is well known that it helps to reduce the amount of software installed and running on your system, so the fact that Server takes a minimalistic approach will hopefully help to make it one of the most secure Linux systems.

We are also making it suitable for running other systems under virtualisation, which can be used as a security feature. For example, where you would run several server tasks on the same Linux platform with other Linux systems, Syllable Server makes it easy to run several versions of itself in virtual machines, each handling only one server task. This would compartmentalise these tasks, so if one would get compromised, it would only affect the one virtual machine, not the others and not the host system. This modern way of working also makes it very easy to move tasks between machines, for example to balance load or to upgrade to better hardware.

Are Linux systems too complex for small-server depending companies?

We think so. Traditionally, operating systems have tried to do more and more tasks. Preferably on the same machine, because hardware was expensive. Unix and Linux have always been good at that. But tasks have changed, and hardware has become cheaper and more powerful. Many systems now need to be operated by people who are not professional system administrators. In some cases, tasks have become even more complex, so there is little time to spend on any one of them, or in other cases they have become simpler by compartmentalising them in separate devices or in virtual machines. In all cases, flexibility and ease of use, for non-expert users, have become much more important. We have been using Linux in our professional lifes for many years, and we see it has a considerable gap towards meeting these needs. The big commercial success of Linux has been in replacing older systems based on several traditional, heavy-duty Unix versions that were used in larger organisations. In more modern tasks that require more flexibility, especially for smaller organisations, it is still failing to a considerable extent. Ease of use is a hard problem, but flexibility is for a large part defined by the Linux distributions and can be fixed. Syllable Server is a modern distribution that is more flexible by allowing only necessary complexity.

Would it be possible to make Syllable Server communicate with Windows, Linux and even Mac OSX?

Definitely. There are many ways to do that. Server is a Linux, so it will be able to communicate in all the ways that Linuxes communicate. We do still have to put a lot of the required software on Server. Ways to communicate with Windows and Mac include exchanging data via FAT-formatted storage media and partitions, and over the network with the SaMBa package, that we provide. I am currently looking into several remote desktop applications. Later, we will also implement our own applications that will be able to communicate between these systems.

Server is meant not only as a server system for Syllable Desktop systems, but definitely also as a server for other systems such as Linux, BSD, Windows and Mac. How that would work depends on what you want to do.

What kind of computers do companies need (to buy) to use the Syllable Server?

Basically any Intel-compatible PC. The system requirements are comparable to Desktop and thus very low. Minimum memory requirements are laughable: 8 MB for the current text mode environment, and probably 32 MB when the graphical environment is ported. Unlike Desktop, Server will eventually work on a 486 processor with a floating point coprocessor, although the current version requires a 686 class processor. Since Server is a Linux, it is potentially easy to port to other machines with different processor architectures, but apart from a 64-bit version, we have no immediate plans to do this ourselves.

When Linux became popular, one of the reasons was that it was so efficient, and made your hardware do more for you. This was in the time that it was usually used exclusively in a text mode environment, such as the first Syllable Server release is now. Since then, Linux has gotten graphical environments, and unfortunately, has largely lost its efficiency advantage. Syllable is going to change this and is bringing a similar efficiency revolution to graphical environments. You can try it on any old equipment you have lying around, and like with Linux in its finest hour, you may well find it becomes full production equipment again.

Would it be possible to have a live CD of this server environment, which can be installed with the settings saved?

We will definitely make a live CD of Server later on, but it may take us a while to get to it. Its functionality will progress over time. One day it may be possible to install directly from that live CD, maybe

even with setting changes as you say, but that's fairly tricky to do, so we haven't fixed any plans for it yet.
A separate installation CD can be installed on smaller systems because it is more efficient, so we may stick with that approach.

What applications are needed for setting up and maintaining this server?

Nothing beyond what's included. Currently that gives you a text mode environment without special configuration tools, so beyond the basic configuration you will have to know your way around a Linux system. This will change when we introduce our graphical environment. This will include the configuration tools from Desktop. After that, we will have to write extra configuration tools for managing settings that are specific to Server. This will all progress over time, and as always, we could use some help with it.
We offer a number of extra software packages, and more can be built with our Builder tool. Some of these may require their own configuration, just like on other systems where this software also runs.

How could the Syllable graphic environment be implemented?

Kristian is working on porting the entire graphical environment from Desktop. He started last year with a prototype running on Fedora, programmed on SDL.
I included SDL in Server, so he is now moving the prototype over to Server. Once it runs there, he will change it from SDL to DirectFB. After that, it will need to be completed and optimised. All Syllable applications will need to be built and tested on Server. Some problems are sure to pop up during that, so a lot of work is still left.

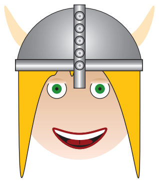
What would it look like? Gnome, KDE, SUN Solaris?

It will look almost exactly like Syllable Desktop. Some people think that we will use a Linux window manager and desktop, but this is not the case. Server is special because it will use the entire Syllable graphical environment and desktop.

Thanks Kaj!

Local & National Groups part 2

In the last issue of the new SDN, I said a few words about the local and national groups, and how their strength is their ability to communicate with the users, in their native language. They are, however, only of little use, if the rest of the system is not available in the users native language.



By Flemming H. Sørensen

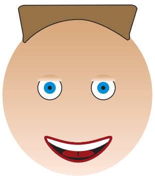
Syllable is currently fully translated to nine languages, and partly translated to another eight languages.
Some might ask the question; isn't it a bit early to provide translations for Syllable, when we know for sure that much will change, before the release of version 1. My answer is; No. It's never too early, to create the fundament for the most basic of all end-user experiences. When people see Syllable, they should see something that clearly states that they are our main focus, and one of the ways to do that, is to acknowledge the fact that not everybody are comfortable with an English operating system. We don't have support for right-to-left languages, but we do support most other languages, atleast in the translations. We got a full Russian translation, and an almost full Greek translation. It is worth noticing that we don't support CJK input, but we are able to use the translations when made on other systems, and saved as UTF8, as seen with the Simplified Chinese translation of AClock.

How translations are done.
The first thing that happens, when I get contacted by a new translator, is they are pointed to the Syllable Translation site, where they are asked to register. At that site, they can download translatable catalogs, for each program, and they can see what languages each program has already been translated to. The site is also used to provide information about updates, so when a translation is made, all that is left to do, is watch the site, and when a translation needs to be updated, there will be a short notice about it, and the strings that needs to be updated. It has worked like that for some time now, and it has turned out to work really well.

How translations will be done.
It might not continue like that, however. Having all the translators come to the same site, works as long as it's only a few languages, and a few translators, but as Syllable grows, we can expect more languages, and many more translators. At the same time, we hope the local and national groups will grow too, and most of the translations could be moved to these groups. One way to organize it, could be to let each group assign one person, who is responsible for the translation. That person would get the catalogs, or updated strings from us, and then send them to his/her own team of translators. This will also make it easier to add peer review on translations, and hopefully give us all even better translations.
I don't see this happen in the near future, however, because the groups are still too small for that, but it can change in just a matter of weeks, for some of the groups.

Interview with Bas de Lange

Bas de Lange is quite a busy bumblebee, he organizes and visits nearly every Open-Source conference, to spread the word, called: Syllable. What does Bas think and say about Syllable? His answers in this next interview.



Bas de Lange

Where did you hear or read about Syllable?

In autumn 2004 I read about Syllable in the Dutch Linux Magazine. Syllable fascinated me because it was so different from other projects. I posted a request to get into contact with a Dutch developer. Because I wanted to ask them to write an article in the Dutch Openmagazine. That's how I met Kaj de vos. The funny part is, that I had already heard about AtheOS in 2002. At that time there was an article on Slashdot or OSNews where Kurt Skauen was asked if he was aware that AtheOS sounded like Atheist; Kurt was not aware of that. That was remarkable for me. It was also the reason why the name AtheOS inspired me to name a dance track after that.

What do you do for Syllable?

I organise conferences (Syl-Con 2007+2008) try to build a Dutch Syllable community (Syllable.nl), try to get Syllable in the press (OpenMagazine, Source21.nl and Syl-Con videos), try to arrange important contacts for Syllable, be present at meetings to promote Syllable (FOSDEM, Software Freedom Day, T-DOSE). I hope one day to make a living out of Syllable.

What future do you see for Syllable?

One of the biggest hurdles for Syllable will be to become a professional OS in the perception of the outside world. And to make clear Syllable is not one of the few thousand Linux distributions. The philosophy of Syllable has to become more widespread. Because that is the big difference between other operating systems and Syllable.

What does Syllable still need?

Lack of normal everyday applications is the biggest problem, like an office suite. And we have to try to build a healthy community around Syllable. That is what I am now trying to do in the Netherlands. To organise gatherings like Syl-Con, HCC Syllable Amersfoort and more. Be present at meetings to represent Syllable, like Be-Geistert and T-DOSE (maybe FOSDEM 2008?) Besides that I am participating in the Dutch OpenDoc Society for promotion of Open Document Formats.

Where does Syllable excel and where does it lack?

Syllable excels in ease of use, hardware recognition. Example: for some reason, my Ubuntu installation does not recognise new networks instantly. Rebooting into Syllable helps me gaining network presence without configuring anything. After that Ubuntu also works.

What kind of apps are missing in Syllable that users are asking for?

Syllable lacks an (Open)Office suite, and a well-designed chat client. What I would also like to see is something like a MediaCenter solution like MythTV or something. Besides all that, it would also be really nice when Gnash, which is now Syllable supported, would become a more mature solution, so that it supports more Flash possibilities, which is also becoming a more and more generally used solution.

What is your advice to developers who want to help Syllable?

Make yourself heard on the forum and or mailing list. We would also really like having people from the creative industry, artists and such, making artwork, "spreading the word", making creative applications for music, video and such.

What technical and other challenges are lying ahead?

I can not speak for the technical part, not being a programmer. But one thing for sure is patience. Putting an operating system in the market and creating a serious market share is not something to be done in one or two years, more something like twenty years (AtheOS started somewhere in 1994). Practically speaking, travelling to conferences is not that easy. Last Syl-Con we had hurdles in family planning, taking care of little tortoises, paying taxes, simply the money etc. :)

Birth of a Syllable User Group

Monday November 19 will be a day to remember, because the Dutch HCC Syllable User Group will have its first meeting. The HCC is the Dutch Hobby Computer Club which was founded in the seventies inspired by the Homebrew Computer Club. And which nowadays organises in every city in the Netherlands monthly meetings, for computer interested people. In Amersfoort, where I live, there is already an HCC Linux chapter, in which I also cooperate. But now Syllable is becoming more and more mature I decided to set up an HCC Syllable User Group Amersfoort. So that I can help people who have problems with installing Syllable Desktop or Server, and so that I can point people to this meeting. "You want to know more about Syllable? Go to the monthly HCC Syllable meeting in Amersfoort!"

In the beginning I guess it will be very silent, because people do not know about Syllable, but once people hear more about it, I expect more and more people. It's the same with the HCC Linux User Group. In the beginning we were just sitting there 'evangelising' people. Now we are there (HCC Linux) with at least four guys, helping people answering questions. And HCC Linux is also a known phenomenon within the HCC. Also because we are one of the younger people there. That is also why HCC likes this HCC Linux User Group. Because they bring "fresh blood".

What will I be doing on such an evening?

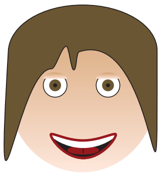
I will have a table, with my Syllable laptop, assisting people with installing Syllable or Syllable-related problems, or pointing them to solutions. Burning CD's will be a problem for me, not having a CD burner in my laptop. But that, people should also be able to do themselves. In the future I will be giving Syllable introductions to people, and once people are familiar with it, I will be giving courses on certain subjects like "starting a virtual e-mail/web-server on Syllable Server", and "organising your office with Syllable Server and Desktop".

Everyone who's interested is invited to celebrate the birth of the first Dutch Syllable User Group at the HCC meeting November 19.

For more info:
http://amersfoort.hcc-utrecht.nl/

Syllable.org down and up

A while ago the Syllable.org site was suddenly gone, what, how and why this happened seemed a big mystery at first, but then things became very clear; the site was 'unplugged'. Kaj explains what went wrong and how it was solved...



Kaj de Vos

Could you explain what exactly happened?

I was there when it happened. The week before I had noticed a few occasions where our site was almost unreachable, but it recovered and it didn't last very long. However, about a week later, the site disappeared completely. Usually, this is just annoying, but it happens and the provider will fix their systems as quickly as possible. We were waiting for that, but it took a long time, and after an hour or so we started researching, to see if we should complain. It's always possible that just your site is affected and the provider hasn't noticed. We found that their own web site was down, too, and we couldn't send email to them, either. On the other hand, Kristian's personal site with the same provider was still running. All in all, this looked like a major malfunction at the provider's site, that might have been precipitated by the earlier disturbance. That doesn't happen often at any provider, so we assumed that our provider was embarrassed and they were hard at work to fix it. Then, Kristian found that the domain name for the domain name server that was serving our syllable.org domain had expired. We could see that it had been paid for five years five years ago, so it looked like our provider had once thought it would be safe to pay in advance, but had subsequently forgotten about it. It was also serving the provider's own domain name, so this explained the extent of the blackout. We thought they had to be negotiating with their domain name registry to get their domain reregistered.

The problem with fixing domain name errors is that it may take up to two days for any changes to propagate through the entire Internet. All our infrastructure was hosted by that provider, including our own domain names, so there was nothing we could do but wait.

The domain name issue, while serious, seemed to be the only problem, because Kristian's site, that turned out to be on a different domain name server, was still running. But towards the end of the two days waiting period that we had in mind, I got restless. Normally, two days is a maximum, so it didn't look good. I figured that clients of this provider had to be discussing the issue, and after a few days it had to be picked up by search engines. So I started searching the Internet, and found a big discussion that had been going on for a week already! It turned out that the sites of most of the provider's customers had gone down a week before already, around the time that I noticed only a temporary problem with ours. The customers had managed to contact a former employee of the provider. They turned out to be a very small provider; he was probably the only employee, and he said the owner had been ill for a long time. There was no trace of him at all. People were discussing whether he was lying dead in his house, but the way the servers went down pointed to malice. The servers were running in a rented colocation space at a bigger data centre. But the data centre's bills hadn't been paid, so the servers were shut down at the end of the month. Somehow, the servers with our sites were the only ones that were still running, so we only noticed it when the domain name server expired, due to the bill of the domain registry going unpaid, too. There were a thousand customers or more, many of whom had been running their businesses through their web sites, so many of them were effectively out of business, and that included their own customers. Some of them were already starting a class action law suit against the provider. They had pleaded with the data centre to give their data back, because many didn't have backups. Data centres never do this, because they are contractually not allowed to give the content of servers to anyone but their customer who installed them. And their customer, our provider, had disappeared without any notice. However, the data centre had turned the servers back on for a number of days to allow people to backup their data through their own user accounts. We didn't have backups, either, but we didn't know time was running out, because our server had kept running. Then, when the domain name server disappeared, we knew our site was probably still running, because Kristian could get at his, but we couldn't get at the Syllable site because we didn't know what IP number it had. And because we had been waiting for the name server to return, the IP number had expired from all the caches on our computers. There was no way to get at the server ourselves, and the data centre wouldn't do it for us.

Was this expected to be happening?

Apart from the temporary disturbance I had noticed a week earlier, there was very little indication. All the customers were saying that it had been a very good provider for many years, and the problems had only begun recently, probably when the employee left. Of course, one should always expect something like this and prepare for it by making backups, but we didn't. Sadly, this is what happens in many organisations, especially undermanned volunteer ones.

Syllable.org has been "off the air" for some time. Were many people interested in Syllable lost?

The site was black for only two days. When we found out it would not be coming back, it turned out that the syllable.org domain name was not registered through our main provider, because it was donated to us a few years ago. The registrar offered a web interface to manage it, so we could quickly point it to another site. We used SourceForge, where our site was hosted until a few years ago (until we moved it because SourceForge was unreliable), to put up a simple notice that we were working on the problem. On a regular day, we get between 500 and 1000 visitors, so maybe a few thousand people got a blank site. However, most of these are probably returning visitors, who would have tried again later, so I don't think we lost many people.

Meanwhile, we discussed how to put up a new site. Our web infrastructure has always been a problem and a pain. Webmasters seem to be like drummers in a band: they keep running off. Each of them leaves semi-random web site systems behind that happened to be his personal favourite. Our main web site was very buggy and ever more flooded with spam, because it was based on an older content management system. The PHP programming and database structures were horrible. The

systems weren't integrated: for example, you had to log in separately to the new bug tracker. We always wanted to do it better but didn't have time. I had long had plans for replacing everything, but I had scheduled to do it piece by piece, most of it next year. We didn't have backups, so we couldn't restore the old systems at a new provider. I definitely didn't want to install yet more semi-random systems, only to replace them some time later, so we decided to accelerate our plans.
I started building my own favourite content management system. ;-) I do, however, claim for myself the credit of not having run off in the past six years. Whether the Curse of Webmasters will make me run off shortly, I don't know...
Two days after we put up the notice, I had a simple content management system with a first reconstruction of our front page, so we had our most important links back and we could start publishing news items again.
I had set up a new provider for the content. We also had to move our domain names to a new provider, point them to the new site and set up our email addresses again. It took a lot of work and a fair amount of time to get it all together, because these things take time for providers to process. Since then, I have been improving and expanding the site and the management system. This will continue for a long time to come. Part of the simple content of the site still needs to be restored, in particular the section for developers. The new site doesn't have any interactive functionality yet, because this is a lot more work to create. Flemming set up a standard forum system to bridge this gap temporarily. However, even if I say so myself, I think the new site already looks better than ever and is better structured.
The content management system is being written in REBOL and is incredibly much simpler and cleaner than the old systems. Eventually, the whole site will be integrated and it will also be integrated with Syllable itself. The content management system will run on Syllable, and things such as the download section will be fully integrated with my Syllable software build system. Once the content management system has matured a bit, we will make it available for others to build their own web sites on Syllable.

Was a lot of information, if not everything, lost?

Since we didn't have backups, basically everything that belonged to the web site was lost. However, besides being easily lost, digital data also has a tendency of being copied around, so there were bits and pieces to be found in every corner of the Internet.
By the way, our development systems are still on SourceForge, so Syllable itself, the source code and software packages, was unaffected. But five years of forum discussions and a year of bug tracking reports are irretrievably lost. The bug tracking affects us immediately, because we were working with that every day. The loss of the forum content is probably worst for new people who want to get acquainted with Syllable. The Syllable User Bible was lost, too, but I'm actually glad about that, because the content was horribly out-dated and nobody ever bothered to do anything about it. The new FAQs that Kristian wrote are much better.

How was this loss of info solved?

The Internet Archive will store parts of a site, but the copies of the Syllable site were years old. I could, however, take a few small things such as style sheets and a few pictures from that archive. Then I spent days trying to think of pages of our site that I wanted to rescue, remembering a few key words that were on them, and using those to find the page by searching with Google. If I didn't wait too long, there would still be a copy of that one page in Google's cache. However, I had to edit the results extensively to reconstruct the original, and it doesn't include pictures. We had changed all the pictures and colours of the site this year, including the logo, so my first reconstruction had all the old styling. Fortunately, Kristian had a copy of some pages with half of the new pictures, and I reconstructed the colours and logo by going back to Ruud's emails in which he sent them. I also retrieved a lot of screenshots from my own archives.
Further, I cobbled together a new site quickly by linking to a lot of external sites that had pieces we needed, so things such as the screenshot section go to SourceForge for the moment being.

Could you use any backups to revive the site?

Only for some loose parts that people happened to have lying around. A big problem in general is that many of what people dare call content management systems don't manage content at all. They're just incredibly primitive web editors that are a web page themselves. Proper client-side web editors tend to be just that: they don't manage entire web sites, either, just single pages. The result is that web sites are edited page by page, live on the web site. The pages on the web server are the only copy. If anything happens to the web server, no part of the system automatically has a copy elsewhere.

Will there be a backup system for such mishaps in the future?

In contrast, my new content management system, eventhough it is still very small and simple, works with a source version of the entire site, a staging version, and the published version. The source version consists of templates, that are processed into a working site in the staging area of your local computer. You can edit your web site off-line, sitting in the park or anywhere, and preview it in the staging area. When you are satisfied with the combined changes, they can be uploaded to the real web site, almost at once.
Although this makes for three copies of the site, it is not a real backup scheme, so there is also a facility for synchronising all the source templates to a private server. In the future, I will add version control to this, so all committed versions of the site will be preserved, and multiple people will be able to work on the site.
Also, the site is now hosted on a distributed system that replicates the site to multiple data centres over the world, so we already have multiple levels of redundancy. It also improves performance, enables us to take on huge waves of visitors when Syllable gets published in a big place and takes care of low-level management for us.

The differences of the Syllable entrance and the forum are obvious, will they get a unified look eventually?

Yes, the eventual forum will get the integrated look of our original forum, from before the crash. The look of our site was quite OK; it was the underlying technology that was gruesome.

Is the current forum generator (phpBB) the final version or are there plans to change to another?

The current forum is temporary and will be replaced when I get the chance to write our own. This is a fairly long-term plan, but eventually I want to combine the forum and the mailing lists into one system, so they're easier to follow and to manage. Our communications are too fragmented now, and people have been asking for it.

How could SPAM be banned in the future?

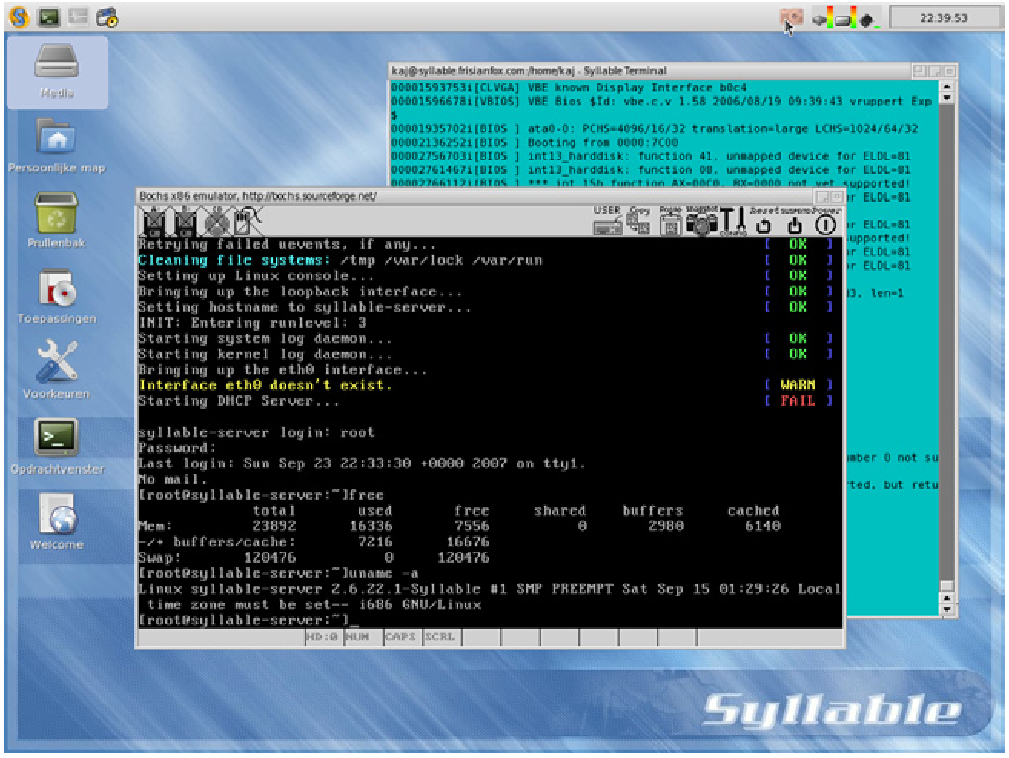
This is another reason for writing our own site and forum. The mainstream technology of PHP and MySQL is way too complex and prone to many attacks, which are often well-known. The temporary forum was immediately overrun with spam, much worse even than on our old, fragile site. Flemming has managed to mostly stop it now, but we had to sacrifice anonymous posting for it and it's probably impossible to make it watertight. Our own technology is much simpler, which makes it far easier to secure, and attackers will not even go after it because they won't know the technology and won't bother with the small number of users. That is, until everybody will make their web sites on Syllable, of course. ;-)

Software Round-up

Kaj de Vos

Quite a lot of software was released since newsletter #1, but not all of it is small enough to be suitable for the archive that accompanies this magazine. We will at least mention the releases.

Syllable Server 0.1



```
00001593753i[CLVGA] VBE known Display Interface b0c4
00001596678i[VBIOS] VBE Bios $Id: vbe.c.v 1.58 2006/08/19 09:39:43 vruppert Exp
$
00001935702i[BIOS ] ata0-0: PCHS=4096/16/32 translation=large LCHS=1024/64/32
00002136252i[BIOS ] Booting from 0000:7C00
00002756703i[BIOS ] int13_harddisk: function 41. unmapped device for ELDL=81
00002761467i[BIOS ] int13_harddisk: function 08. unmapped device for ELDL=81
```

```
Retrying failed uevents, if any...                                    [  OK  ]
Cleaning file systems: /tmp /var/lock /var/run                        [  OK  ]
Setting up Linux console...                                           [  OK  ]
Bringing up the loopback interface...                                 [  OK  ]
Setting hostname to syllable-server...                                [  OK  ]
INIT: Entering runlevel: 3
Starting system log daemon...                                         [  OK  ]
Starting kernel log daemon...                                         [  OK  ]
Bringing up the eth0 interface...
Interface eth0 doesn't exist.                                         [ WARN ]
Starting DHCP Server...                                               [ FAIL ]

syllable-server login: root
Password:
Last login: Sun Sep 23 22:33:30 +0000 2007 on tty1.
No mail.
[root@syllable-server:~]free
              total       used       free     shared    buffers     cached
Mem:          23892      16336       7556          0       2980       6140
-/+ buffers/cache:        7216      16676
Swap:        120476          0     120476
[root@syllable-server:~]uname -a
Linux syllable-server 2.6.22.1-Syllable #1 SMP PREEMPT Sat Sep 15 01:29:26 Local
 time zone must be set-- i686 GNU/Linux
[root@syllable-server:~]_
```

The most important release for the project was the very first version of Syllable Server. Regular readers will know that this is a Linux distribution that is built in the same way as Syllable. It is meant as a server companion, and the regular Syllable is now called Syllable Desktop. Several of the additional software packages that are available for Syllable Desktop were also released in a version for Syllable Server, such as Developer's Delight and Network Necessities. Later, Syllable Server was also released as disk images for several emulators. It can even run on Syllable Desktop!

Syllable Desktop 0.6.4 Premium CD

It was a long time ago that we made the Premium CD that you can order through the Syllable Internet shop at Lulu (http://www.lulu.com/syllable). We have now made time to produce a new CD for Syllable 0.6.4. Again, it is an extended version of the installation CD that contains all our official software packages. This time, it is filled to the brim with the source code for all those packages.

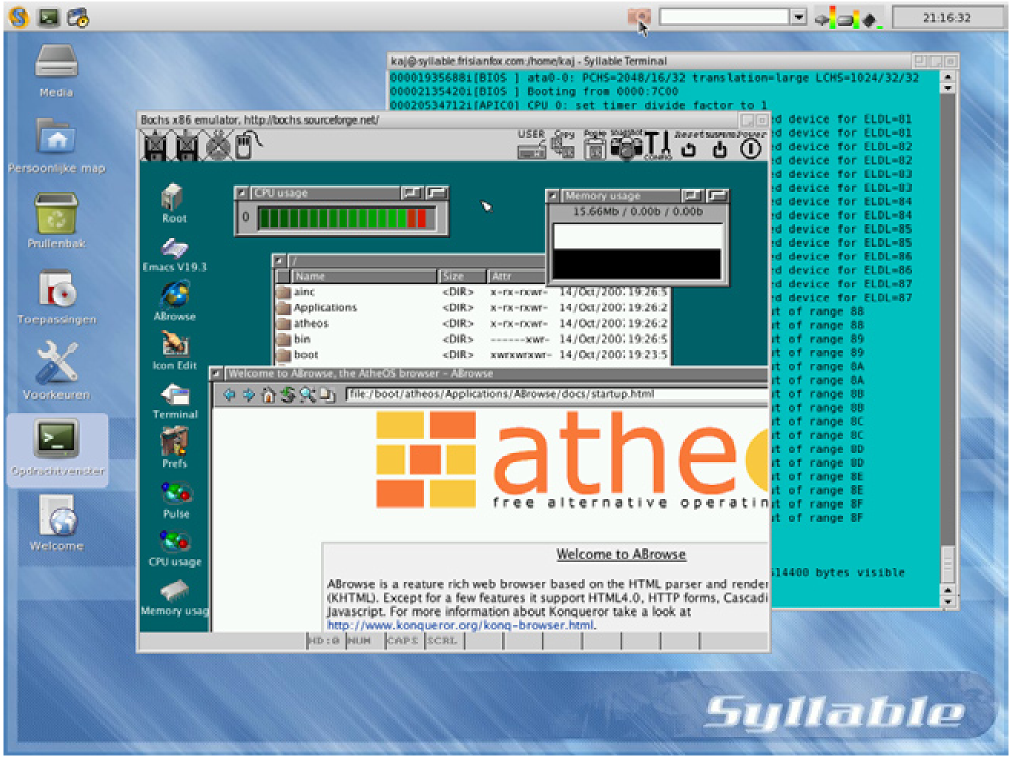
Blue decorator

John Aspras wrote another window decorator, Blue.

Gnash and other ports

Several people got active and ported SDL-based applications. This is a great way to start contributing to Syllable. The other side of the coin is that it's more work for the Syllable team to integrate ports from less experienced contributors into the build system. The production and publication of Syllable Server and the new web site took a major amount of time, so there are no packages for the new ports yet. The most notable port was Gnash and all the libraries it needs, by Kelly Wilson, so Syllable now has a stand-alone Flash player (some Flash movies could already be played with the Media Player).

AtheOS emulator images



```
00001935688i[BIOS ] ata0-0: PCHS=2048/16/32 translation=large LCHS=1024/32/32
00002135420i[BIOS ] Booting from 0000:7C00
00020534712i[APIC0] CPU 0: set timer divide factor to 1
d device for ELDL=81
d device for ELDL=81
d device for ELDL=82
d device for ELDL=82
d device for ELDL=83
d device for ELDL=83
d device for ELDL=84
d device for ELDL=84
d device for ELDL=85
d device for ELDL=85
d device for ELDL=86
d device for ELDL=86
d device for ELDL=87
d device for ELDL=87
t of range 88
t of range 88
t of range 89
t of range 89
t of range 8A
t of range 8A
t of range 8B
t of range 8B
t of range 8C
t of range 8C
t of range 8D
t of range 8D
t of range 8E
t of range 8E
t of range 8F
t of range 8F

514400 bytes visible
```

Name	Size	Attr	
ainc	<DIR>	x-rx-rxwr-	14/Oct/200: 19:26:5
Applications	<DIR>	x-rx-rxwr-	14/Oct/200: 19:26:2
atheos	<DIR>	x-rx-rxwr-	14/Oct/200: 19:26:2
bin	<DIR>	-------xwr-	14/Oct/200: 19:26:5
boot	<DIR>	xwrxwrxwr-	14/Oct/200: 19:23:5

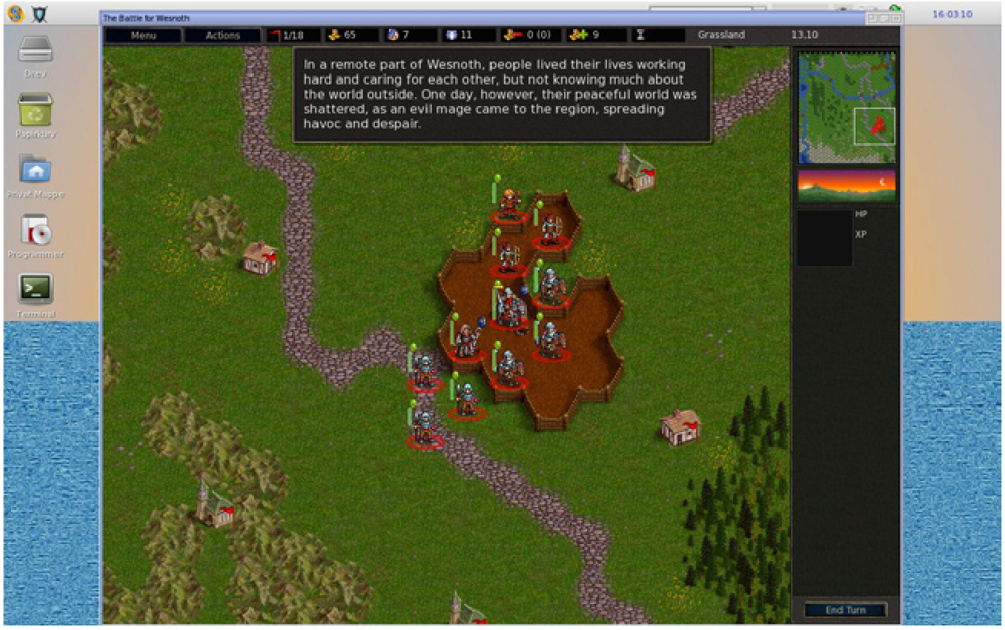
file:/boot/atheos/Applications/ABrowse/docs/startup.html

Welcome to ABrowse

ABrowse is a reature rich web browser based on the HTML parser and render (KHTML). Except for a few features it support HTML4.0, HTTP forms, Cascadi Javascript. For more information about Konqueror take a look at http://www.konqueror.org/konq-browser.html.

For old times' sake, we released disk images of the last AtheOS version, 0.3.7.
They can be used with several emulators.
New readers should know that AtheOS was the predecessor project of Syllable, until 2002.

Battle for Wesnoth

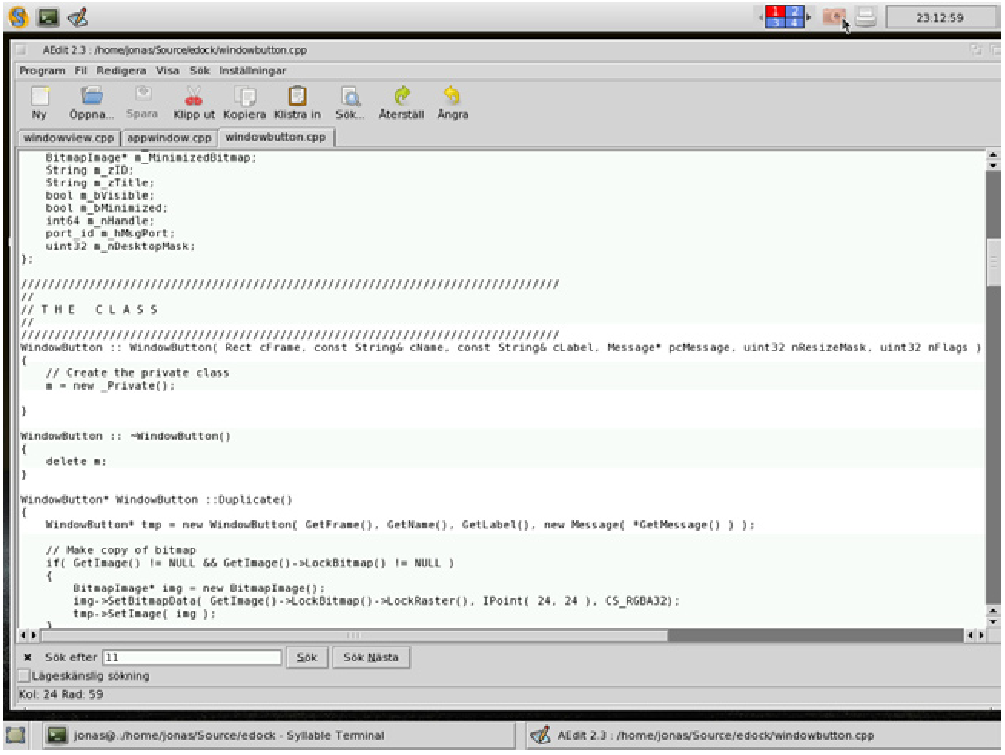


In a remote part of Wesnoth, people lived their lives working hard and caring for each other, but not knowing much about the world outside. One day, however, their peaceful world was shattered, as an evil mage came to the region, spreading havoc and despair.

Flemming ported several versions of Battle for Wesnoth, culminating with 1.3.8 for the moment being.
It's a great game, so it's well worth checking out the download at his Syllable Norden site.

The following items are included in the software archive for this newsletter.

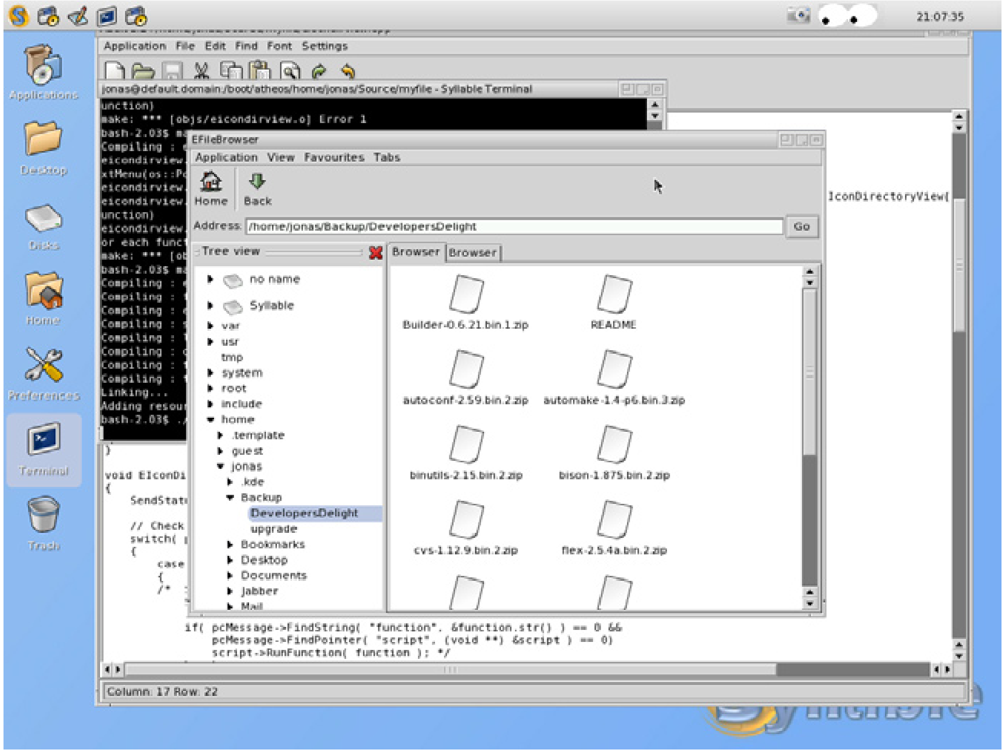
EDock (binary and source)


23:12:59
AEdit 2.3 : /home/jonas/Source/edock/windowbutton.cpp
Program Fil Redigera Visa Sök Inställningar
Ny Öppna... Spara Klipp ut Kopiera Klistra in Sök... Återställ Ångra
windowview.cpp appwindow.cpp windowbutton.cpp
Sök efter 11 Sök Sök Nästa
Lägeskänslig sökning
Kol: 24 Rad: 59
jonas@../home/jonas/Source/edock - Syllable Terminal
AEdit 2.3 : /home/jonas/Source/edock/windowbutton.cpp

```
    BitmapImage* m_MinimizedBitmap;
    String m_zID;
    String m_zTitle;
    bool m_bVisible;
    bool m_bMinimized;
    int64 m_nHandle;
    port_id m_hMsgPort;
    uint32 m_nDesktopMask;
};

////////////////////////////////////////////////////////////////////////////////
//
// T H E   C L A S S
//
////////////////////////////////////////////////////////////////////////////////
WindowButton :: WindowButton( Rect cFrame, const String& cName, const String& cLabel, Message* pcMessage, uint32 nResizeMask, uint32 nFlags )
{
    // Create the private class
    m = new _Private();

}

WindowButton :: ~WindowButton()
{
    delete m;
}

WindowButton* WindowButton ::Duplicate()
{
    WindowButton* tmp = new WindowButton( GetFrame(), GetName(), GetLabel(), new Message( *GetMessage() ) );

    // Make copy of bitmap
    if( GetImage() != NULL && GetImage()->LockBitmap() != NULL )
    {
        BitmapImage* img = new BitmapImage();
        img->SetBitmapData( GetImage()->LockBitmap()->LockRaster(), IPoint( 24, 24 ), CS_RGBA32);
        tmp->SetImage( img );
    }
```

Jonas wrote a welcome companion for the Dock. It shows an extra panel at the bottom of the screen with a more extensive window list than the Dock shows, and a button to minimise all windows and show the desktop.
Unzip the package and start the dock with “./edock &”.

EFileBrowser (binary and source)



```
unction)
make: *** [objs/eicondirview.o] Error 1
bash-2.03$ m
Compiling : 
eicondirview.
xtMenu(os::P
eicondirview.
eicondirview.
unction)
eicondirview.
or each func
make: *** [o
bash-2.03$ m
Compiling : 
Compiling : 
Compiling : 
Compiling : 
Compiling : 
Compiling : 
Compiling : 
Compiling : 
Linking...
Adding resou
bash-2.03$ .
```

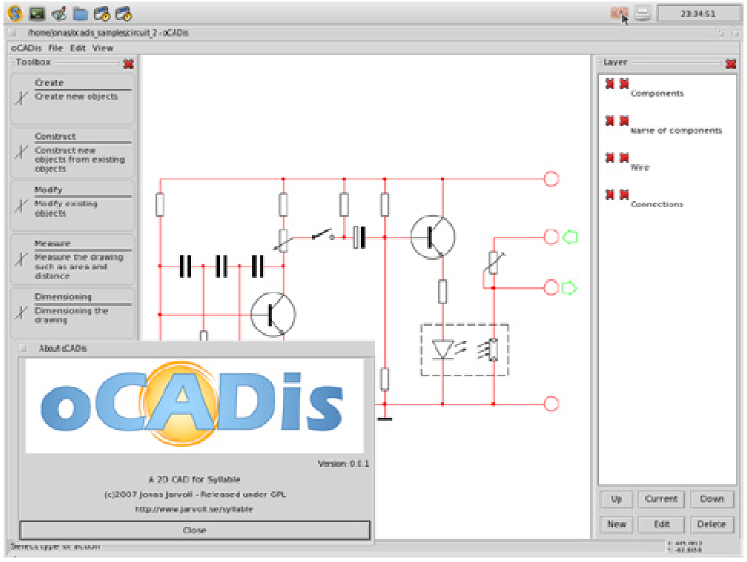
```
IconDirectoryView(
}

void EIconDi
{
    SendStat

    // Check
    switch( 
    {
        case
        {
        /*
            if( pcMessage->FindString( "function", &function.str() ) == 0 &&
                pcMessage->FindPointer( "script", (void **) &script ) == 0)
                script->RunFunction( function ); */
```

Furthermore, Jonas released a few applications that he had been sitting on for a long time, including his already popular extended filer. Jonas asks to keep in mind that they’re not finished yet.

oCADis (binary and source)

Continuing with his CAD program. Jonas gives these tips:

- *You need a mouse with a wheel button.*
  Using the wheel you zoom in and out of your drawing and you pan (move) the drawing by pressing the wheel and moving the mouse.
- *You can enter coordinates manually in the appropriate textview. oCADis also handles calculation so you can enter something like 10+10/2. Pressing Enter moves the cursor to the next entry, i.e. click first in the X box, type your value, press Enter and the cursor will be moved to Y. Type your Y value and press Enter again to make oCADis accept your new items.*
- *In the menu View/Dialogs you find many things that can help you improve your drawing: layers, object snap points, snap/grid etc.*
- *Load and save should work fairly OK.*
- *Load and save in any other format than oCADis is not supported yet!*



Version: 0.0.1

A 2D CAD for Syllable

(c)2007 Jonas Jarvoll - Released under GPL

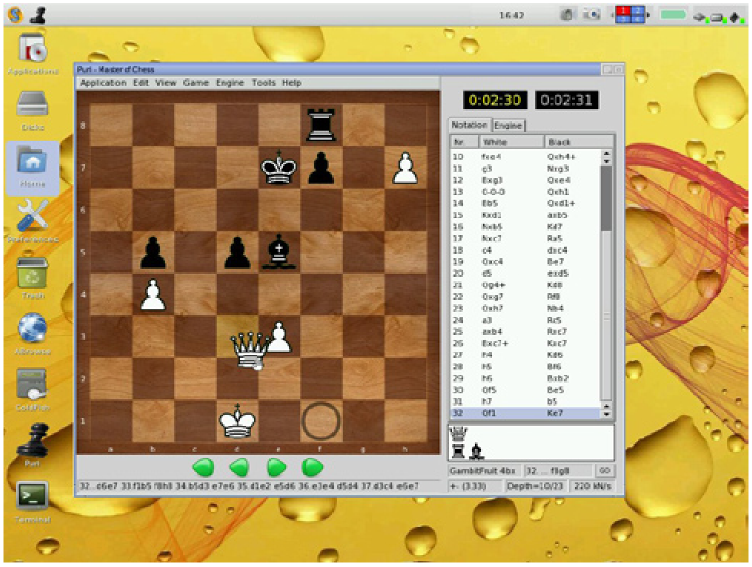
http://www.jarvoll.se/syllable

Hex Editor 0.1 (source only)

Andrew Kennan wrote a new application, a very useful utility for viewing and modifying any file in hexadecimal mode, where all bytes are shown as numbers. Only the source code is currently available, so you have to compile it.

Puri 0.1.8

Ciprian released an improved version of his Puri Master of Chess, a native chess application for Syllable.



Nr	White	Black
10	fxe4	Qxh4+
11	g3	Nxg3
12	Bxg3	Qxe4
13	0-0-0	Qxh1
14	Bb5	Qxd1+
15	Kxd1	axb5
16	Nxb5	Kd7
17	Nxc7	Ra5
18	c4	dxc4
19	Qxc4	Be7
20	d5	exd5
21	Qg4+	Kd8
22	Qxg7	Rf8
23	Qxh7	Nb4
24	a3	Rc5
25	axb4	Rxc7
26	Bxc7+	Kxc7
27	h4	Kd6
28	h5	Bf6
29	h6	Bxb2
30	Qf5	Be5
31	h7	b5
32	Qf1	Ke7

32. .. f8g8

+- (3.33) Depth=10/23 220 kN/s

32...d6e7 33.f1b5 f8h8 34.b5d3 e7e6 35.d1e2 e5d6 36.e3e4 d5d4 37.d3c4 e6e7

Python 2.5.1

Python was updated from its previous, rather old port to the newest version.
The original AtheOS parts of the port were cleaned up, and it was submitted to the Python project at their request.

Bochs 2.3.5

A slightly updated version of Bochs was released.
Most notable is the included configuration file, that contains configuration examples for several disk images.

Programming tutorial

Also in the archive are the two source code examples from Kristian’s programming tutorial in this issue.