sirjofri.de changeblog

Fri, 18 Dec 2020 15:31:32 +0100 :

Automatically save sent files in “Sent”

Since it is what many mail clients do it might be helpful for other people to have this. As for me, I like to have all my outgoing mails automatically saved in a *Sent* directory, so that’s what I want to do.

There are multiple ways to do this. I want to present a very simple way without sending your mail to yourself or something like that.

Outgoing mail is processed via `upas/send` or `/mail/box/$user/pipefrom`, if that exists. We use this feature to build or own little filter script for outgoing mails.

The script itself is very simple. We just need a temporary place to store the mail message, then save it in the `Sent` directory and forward it to the normal send routines:

```
#!/bin/rc
rfork en
# see /sys/src/cmd/upas/filterkit/pipefrom.sample .
bind -c /mail/tmp /tmp
TMP=/mail/tmp/mine.$pid
cat > $TMP
/bin/upas/mbappend /mail/box/<yourusername>/Sent < $TMP
/bin/upas/send $* < $TMP
```

Of course you need to create the directory `/mail/box/<yourusername>/Sent` and exchange `<yourusername>` with your `$user` and make this script executable.

The `Sent` directory needs read and write access for your own user, which should be fine with the defaults.

If you want unauthenticated users to send mails to that directory you need to make this directory world-writable.

With these adjustments you can send mails with acme/marshal and they are automatically saved in your `Sent` mailbox.

Tue, 15 Dec 2020 11:33:54 +0100 :

9front on Lenovo Thinkpad Twist

A few weeks ago I removed archlinux from my remaining machine. I noticed how the new lenovo keyboards aren't good and the trackpoint is crap. That's why I still prefer the Thinkpad T61, even without battery.

Anyways, I'll try to describe the process of the installation. The installation itself went according to the FQA, I'll just add some notes.

Process

First I had to disable UEFI completely and switch to legacy BIOS. I know 9front can handle UEFI somehow, but I never got it working on any machine. To make 9front work with legacy BIOS I had to change the SSD layout from GPT to MBR. This was possible, just remove all partitions and use the command line to create DOS partitions. Then the SSD was detected as MBR/non-GPT and I could proceed with default installation.

After installation I needed to get WIFI working. Thanks to 9front developers I was able to use BSD drivers as documented in the FQA. In my case I just grabbed the `iwn-2030` drivers, placed them in `/lib/firmware` and built the kernel from scratch.

Enabling ACPI in the `plan9.ini` and starting `aux/acpi` went without errors, only `bad opcode` warnings showed up. Still, everything works as expected, so I didn't investigate further.

Issues and Troubleshooting

I had exactly **two** issues. The first is the `bad opcode` as described earlier.

The second is a big surprise. Backlight controls work out of the box! I know older machines handle this directly without using the operating system, but this was a modern machine. Still it worked with only one tradeoff:

It always prints *lapicerrors* on the console. I didn't find a good way to disable them, so I just added a hidden window in my riostart that just `cat`/dev/kprint so the errors don't fill the screen.

Bonus: conntosrv

As a bonus I have a small script that saves me lots of installation time. I have a server with my $home directory, including some configuration in my `lib/profile`. On my terminals (laptops) I just work in my $home like it was right there on my machine.

To make this happen I placed the little script in my terminal's `/cfg/$sysname/conntosrv` and called in the `/cfg/$sysname/termrc`.

The script contains:

```
#!/bin/rc

echo -n 'connect to server: '
server=`{read}

if(~ $#server 0){
        echo not connecting to services >[1=2]
        exit
}

if(! test -e /net/dns)
        ndb/dns -r

auth/factotum
for(i in $server){
        rimport -Cc $i /n/$i
}
bind -c '/n/'^$server(1)^'/usr/'^$user /usr/$user
```

As you can see, the scrips connects to all servers you input at the prompt. It takes the first to be your $home, all others are imported to `/n`.

Wed, 29 Jul 2020 10:32:43 +0200 :

Restrict RCPU User Access to Groups

This is how to restrict user access to groups. You can use this to enable `rcpu` access for all users of a specific group. All other groups will not be allowed.

To allow access only to `sys` group members: adjust your `/rc/bin/service/tcp17019`

```
#!/bin/rc
userfile=/adm/users
fn useringroup{
        grep $1 $userfile | {
                found=0
                while(~ $found 0 && line=':{read}){
                        if(~ $line(2) $2){
                                found=1
                        }
                }
                if(~ $found 1)
                        status=''
                if not
                        status='not found'
        }
}
if(~ $#* 3){
        netdir=$3
        remote=$2!'{cat $3/remote}
}
fn server {
        ~ $#remote 0 || echo -n $netdir $remote >/proc/$pid/args
        rm -f /env/'fn#server'
        . <{n='{read} && ! ~ $#n 0 && read -c $n} >[2=1]
}
exec tlssrv -a /bin/rc -c 'useringroup $user sys && server'
```

This checks if the user is in group `sys` and only then calls the `server` function. Otherwise the connection is terminated.

This is especially useful if you want a CPU server to expose filesystems *and* have cpu access for administrators only.

Thu, 16 Jul 2020 09:44:56 +0200 :

lib/profile quick hack

Some smaller change that can change your life.

There are reasons why you not run *rio* in your lib/profile. For me the main reason would be: You can no longer use `rcpu -c commands` in your shell. Rio opens and there you are, stuck in front of a gray background.

My solution:

```
case cpu
   # … lots of stuff …
   rcpucmd=‘{cat /mnt/term/env/cmd >[2]/dev/null}
   if(~ $#rcpucmd 0)
      rio
   # … lots of stuff …
```

Now I can `rcpu` and have my rio, or `rcpu -c command` and run the command without leaving my shell.

Thu, 16 Jul 2020 08:41:14 +0200 :

Mail Server Configuration

Recently I installed my mail server on 9front. Most of the time I followed the guide in the FQA, but still there are things to explain. In this document I'll go through the section of the FQA and annotate things.

Right at the beginning the FQA mentions how the executing user needs write permissions for the mail-boxes. This is *very important*! If upas can't write the mailboxes the mail server will *not* accept incoming mail!

In my setup I can skip all DNS stuff, because I have my DNS hosted somewhere else. Make sure to add proper MX records as well as (at least) an SPF record.

/mail/lib/smtpd.conf

To make things short, here are the necessary lines in my setup. The server handles authenticated incoming mail for sending to other providers as well as incoming mail for local accounts.

```
defaultdomain    sirjofri.de
norelay          on
verifysenderdom  on
saveblockedmsg   off
ourdomains       sirjofri.de
```

Note that the server is no relay for unauthenticated/untrusted requests, it will still relay if you authenticate.

At this point it might be a good idea to check your user password. Use `auth/changeuser` to add *Inferno/POP secrets* to your user accounts. Use these passwords to authenticate to the smtp server.

/mail/lib/rewrite

The program that handles sending mail uses this file to rewrite mail addresses. This file is responsible for filtering out local mail as well as sending other mails to the mailer.

In my setup I added three aliases:

```
pOsTmAsTeR    alias postmaster
aBuSe         alias abuse
wEbMaStEr     alias webmaster
```

Use regular expressions to define your domain:

```
de!(.*)   alias 1
sirjofri.de!(.*)       alias 1
```

For translating mails I added one more rule for mail address *tags*. These tags are in the form of *user+tag@example.com*. Official specifications say that everything behind that "+" must be ignored, but it can be used to automatically sort incoming mail into folders. I do this, by the way, so I describe here, how.

We need rules for those plus signs:

```
# The other translate rules are default
```

For delivering local mails, I added extra rules:

```
local!(.+)+(.+)  |  "/bin/test -d /mail/box/1/2  /bin/upas/mbappend /mail/box/1/2 || /bin/upas/mb
local!"(.+)+(.+)  |  "/bin/test -d /mail/box/1/2  /bin/upas/mbappend /mail/box/1/2 || /bin/upas/m
# leave the other rules untouched.
```

With this settings, mails to user+*tag* will be checked. If a mailbox folder for *tag* exists, mail is sent to this folder. Otherwise it is sent to the user's default inbox. **Note:** I tested, but this *does not work* with aliased mail. If my aliasmail changes *userA* to *userB*, mails to *userA+tag* will be rejected! If you know how I can

make this work, feel free to send me a mail.

/mail/lib/names.local

This file is pretty easy. Just add your alias mail addresses:

```
postmaster  sirjofri
webmaster   sirjofri
abuse       sirjofri
```

/mail/lib/remotemail

```
#!/bin/rc
shift
sender=$1
shift
addr=$1
shift
fd=‘{/bin/upas/aliasmail -f $sender}
switch($fd){
case *.*
    ;
case *
    fd=sirjofri.de
}
exec /bin/upas/smtp -h $fd $addr $sender $*
```

SMTP over TLS

I don’t use port 587. I use 25 for this. Mail servers relay mails to this port by default, so it makes sense.

/rc/bin/service/tcp25

```
#!/bin/rc
user=‘{cat /dev/user}
exec /bin/upas/smtpd -f -E -r -c /sys/lib/tls/cert -n $3
```

Don’t forget to create your TLS certificate!

IMAP4 over TLS

I did this exactly like the FQA. See there.

No.

At this point I stopped. I did not configure ratfs and have no spam handling right now. It doesn’t really matter for me, because nobody knows me and I don’t use that mail address to register anywhere.

Links:

→ https://fqa.9front.org/fqa7.html#7.7

Wed, 01 Jul 2020 18:30:46 +0200 :

Guided Replica

Today I installed on my VPS. I noticed that I can write some helper scripts around it and here they are.

You can download them from `https://sirjofri.de/files/guidedreplica`.

You can install it like that:

```
# bind your client $home to /n/rclient
# bind your server $home to /n/rserver
hget https://sirjofri.de/files/guidedreplica/guidedreplica.rc | rc
# follow the prompts
```

This will also install two helper scripts to `$home/bin/rc/replica/`. Reproto copies one proto over the other. You can choose which one you want to keep. Reupdate is helpful if there are update-update errors. It should automatically solve them (untested, but should work).

Update: *replica*(1) has issues. Often it does a bad job tracking changes, leaving removed files there and vice versa. I never encountered data loss, only inconsistencies in the copies.

Many people use *mkfs*(8), which does not overwrite changed files. At some point I will build some scripts around it and use that instead of *replica*(1).

(Files: `https://sirjofri.de/files/guidedreplica/README` `https://sirjofri.de/files/guidedreplica/guidedreplica.rc`)

Mon, 29 Jun 2020 18:39:39 +0200 :

9front on Netcup VPS

Today I installed 9front on a Netcup VPS. Here are some notes if you want to do it yourself.

I used the smallest VPS option. Currently, that's "VPS 200 G8". It costs like 2.69 Euro, but you might be able to find some way to make it cheaper.

After ordering it might take some time until the server is up and ready. By default debian was installed in a GPT, we can ignore that.

Before we can install our custom ISO we first must upload it somewhere. This is done via FTP (you get the access data from the SCP), I used windows default file explorer (`ftp://user@address`, enter password). Copy the 9front ISO in `/cdrom`. This will take some time.

Meanwhile you can delete the virtual disk and create a new one. You need your SCP password for this. This step is necessary to remove the GPT. Of course you could manually reformat the disk, but deleting the disk will save time.

In the settings you can virtually insert the iso as a DVD and verify the boot order (DVD first). Start up the machine and switch to the web VNC display.

At this point you can proceed with the default 9front installation described in the fqa. Don't forget to install the MBR and activate the partition. Otherwise there are no additional special steps besides manually configuring the `/lib/ndb/local` after installation. In my case I made an auth server.

Currently it seems to work fine. I installed the machine today, so there might be some issues I didn't find yet.

Tue, 23 Jun 2020 15:00:45 +0200 :

changeblog feed — social media²

RSS is still a thing.

Yes, there are more modern alternatives, like Atom or fancy json feeds. What I want to say is, feeds are still a thing.

That’s why you are now able to read my changeblog as an Atom feed.

Now I just need to find enough time to write my posts.

Fri, 22 May 2020 02:00:00 +0200 :

I use 9front

Today I want to share with you, that I use the plan9 distribution “9front” as my main computer.

Of course there are things that are almost impossible to do there, for example: all gamedev related stuff. This is of course an issue, because I am a game developer. I still have my windows machine with relevant tools, so I can still fiddle around with those complex things.

For gaming I also use my windows machine or some game console. Yes, there are a few games on plan9 systems.

Also most online services use javascript and heavy styling of webpages, so I also use a modern computer with a modern browser. Mothra is fine for doing basic research stuff, but in 2020 it’s almost impossible to actually do things on the web.

Anyways, let me tell you that I don’t really miss anything on plan9. I can write documents, check my email stuff, chat with people, and step by step it becomes more usable. The community is helpful and provides more applications. The system runs stable, the user interface is consistent and good to look at. Colors don’t jump in your eye and want to kill you and there’s catclock(1), our friendly companion.

Fri, 10 Jan 2020 01:00:00 +0100 :

Revived

I updated my website to Uberspace 7, but not only this: I changed the whole webpage to make it more nine-friendly.

My whole webpage management system is completely 9 based. I use oridb's git9 implementation and plan9 tools, mk, sed, cat, …

I also decided to change the main language of the website to English.