

Plan 9 na pratica

SSH, acesso remoto, uso cotidiano e projetos pessoais

A resposta curta

O 9front atual tem **cliente SSH2**. Ele conecta do 9front para Debian, OpenBSD e outros servidores. Para entrar no 9front, prefira **drawterm/rcpu**. O SSHv1 historico do Plan 9 original nao e uma opcao segura hoje.

Complemento visual ao estudo anterior

Preparado para **Pablo Murad**
12 de setembro de 2026



Como este complemento foi pensado

O guia anterior explicou a arquitetura do Plan 9. Este volume faz outra coisa: pega as ideias e as transforma em gestos concretos. Você vai ver como entrar no sistema, como sair dele para um servidor Debian, o que os comandos realmente fazem e quais projetos valem seu tempo.

Regra de leitura

Todo exemplo traz o ambiente a que pertence. **[9front]** significa o sistema completo em VM.

[plan9port] significa ferramentas do Plan 9 rodando sobre Linux. **[host]** significa Debian, Fedora ou Windows.

Sumario

Como este complemento foi pensado	1
Sumario	2
1. A resposta direta: Plan 9 tem SSH?	5
O que funciona hoje	5
Por que havia tanta informacao contraditoria?	5
2. Antes dos comandos: o que voce vai ver	6
Como ler essa tela	6
3. O mapa mental do acesso remoto	7
Tres movimentos diferentes	7
Uma analogia honesta	7
O que o rcpu atual promete	7
4. SSH saindo do 9front: primeiro uso	8
Passo 1: testar a rede	8
Passo 2: abrir a sessao	8
Passo 3: validar a chave do host	8
Passo 4: executar um comando sem abrir shell	8
5. sshfs: transformar o Debian em uma arvore local	9
Montagem padrao	9
Montagem explicita e somente leitura	9
O detalhe tecnico que vale aprender	9
6. sshnet: importar a rede de outra maquina	10
O experimento	10
Por que testar em uma janela separada	10
Um uso pessoal plausivel	10
7. Entrando no 9front com drawterm	11
No Fedora	11
No Windows	11
Dentro da sessao	11
8. Tailscale sem instalar Tailscale no Plan 9	12
Fluxo recomendado	12
O que nao fazer	12
9. Rota A: aprender ferramentas com plan9port	13
Instalacao no Debian com interface grafica	13
Variaveis de ambiente	13
Onde fazer isso no seu caso	13
10. Acme por dentro	14
Os tres botoes	14

11. Tutorial de Acme em 15 minutos	15
1. Abrir uma pasta	15
2. Editar e salvar	15
3. Executar um comando	15
4. Abrir caminhos como hipertexto	15
5. Criar uma janela de shell	15
12. Plan9port convivendo com Linux	16
Um fluxo pessoal realista	16
13. rc: um shell pequeno, mas diferente	17
Variaveis e listas	17
Capturar a saída de um comando	17
Condicao simples	17
Quatro diferencas que evitam frustracao	17
14. Rota B: uma VM 9front no Kaiser	18
Configuracao inicial sugerida	18
Duas formas de chegar ao primeiro boot	18
Depois da instalacao	18
15. A primeira hora dentro do 9front	19
Ajuda que voce deve usar sempre	19
16. Laboratorio de namespace: duas pastas, uma visao	20
Preparar as origens	20
Compor a visao	20
O que aconteceu	20
A aplicacao pessoal	20
17. Rede como arquivos: olhando /net	21
Modelo simplificado de uma conversa	21
Exemplo de HTTP sem esconder a arvore	21
18. 9P em um exemplo pessoal	22
Como o cliente usaria	22
Por que isso e diferente de uma API REST	22
19. Compilar um programa pequeno	23
Compilar no 9front amd64	23
Compilar com plan9port	23
Linha por linha	23
20. Ideias para uso pessoal, sem fantasia	24
Minha recomendacao, em ordem	24
21. Projeto 1: estacao de rascunhos do digital garden	25
Fluxo de trabalho	25

Experimento de namespace	25
Entrega apos uma semana	25
22. Projeto 2: painel do homelab como sistema de arquivos	26
Fase 1: read-only	26
Fase 2: historico simples	26
Fase 3: comandos controlados	26
23. Projeto 3: arquivo fotografico X100VI	27
Consultas simples	27
O que fica fora	27
24. Projeto 4: uma janela Plan 9 para a Murad Radio	28
Primeira versao	28
Segunda versao, se a primeira provar valor	28
25. Projeto 5: estacao-museu e pubnix privado	29
Servicos que fazem sentido	29
Servicos que nao fazem sentido	29
Regra de moderacao	29
26. Roteiro de 30 dias	30
Diario minimo	30
Ao final, responda	30
27. Onde parar	31
Sinais de que o projeto esta saindo do trilho	31
A leitura correta	31
28. Folha de consulta rapida	32
Arquivos e caminhos que reaparecem	32
29. Fontes, data e creditos	33
Referencias tecnicas	33

1. A resposta direta: Plan 9 tem SSH?

Sim, mas a resposta depende de qual Plan 9 você está falando e da direção da conexão. O 9front atual possui um cliente SSH2 nativo. Ele também possui `sshfs`, que apresenta um servidor SFTP como uma árvore de arquivos, e `sshnet`, que apresenta a pilha TCP do servidor remoto como parte de `/net`. Isso é muito Plan 9: até o SSH vira sistema de arquivos.

ORIGEM	PROTOCOLO	DESTINO	USO CERTO
9front	SSH2	Debian / OpenBSD	Shell, comando, SFTP, TCP
Windows / Fedora	drawterm + rcpu	9front	Sessão gráfica ou textual
Plan 9 histórico	SSHv1	Sistemas antigos	Somente arqueologia

O que funciona hoje

Situação	Resposta	Ferramenta
9front entra no Debian/OpenBSD	Sim, via SSH2	ssh
9front monta arquivos do Debian	Sim, via SFTP v3	sshfs
9front usa a rede do Debian	Sim, via túnel SSH	sshnet
Windows/Fedora entra no 9front	Sim, mas não como OpenSSH comum	drawterm / rcpu
OpenSSH entra direto no 9front	Não é o fluxo nativo atual	use drawterm / rcpu
Plan 9 original usa SSH	Só a implementação histórica SSHv1	ssh1 / sshserve

Conclusão operacional

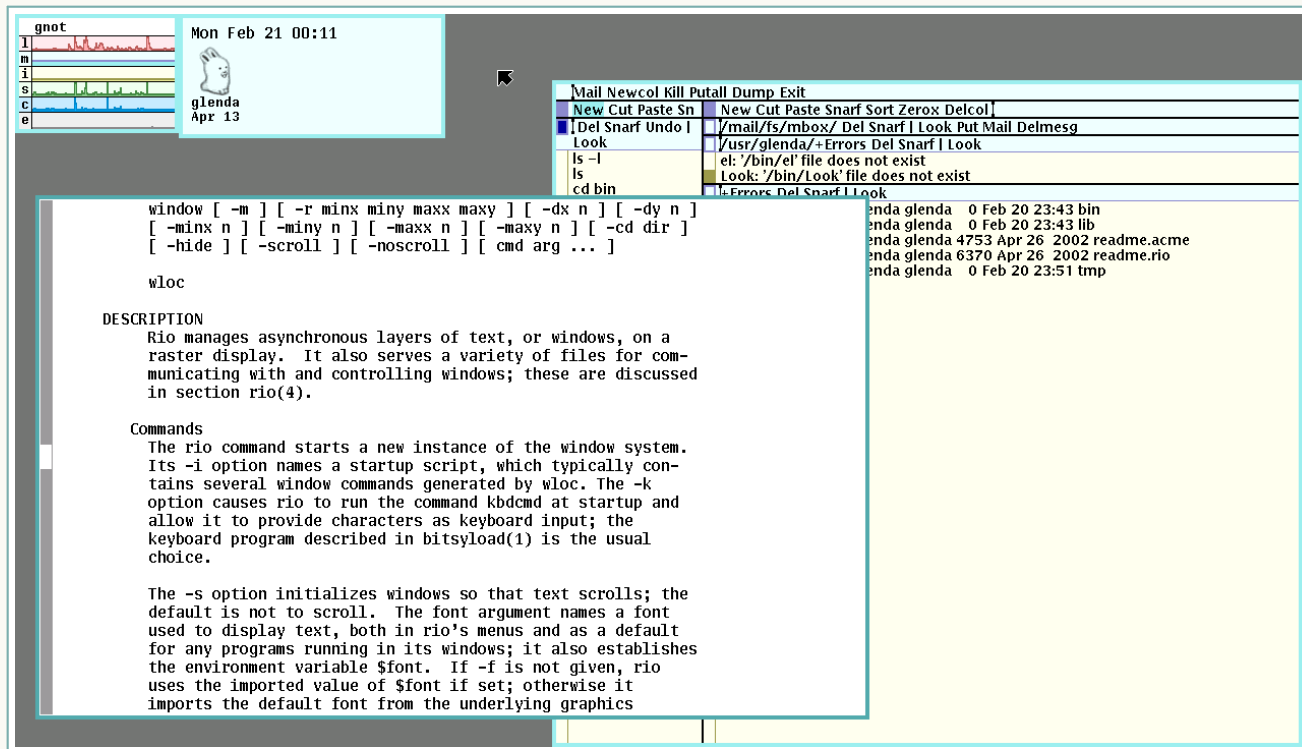
No seu homelab: use **SSH saindo do 9front** e use **drawterm entrando no 9front**. Não tente transformar a VM em mais um servidor OpenSSH exposto na internet.

Por que havia tanta informação contraditória?

A documentação antiga descreve o SSHv1 do Bell Labs. A árvore atual do 9front, consultada nesta pesquisa, contém um cliente SSH2 diferente, com RSA/SHA-256, Curve25519 e ChaCha20-Poly1305. Ao mesmo tempo, a árvore atual não apresenta um `sshd` ou `sshserve` moderno equivalente ao OpenSSH. Essa última observação é uma inferência feita a partir da árvore de fontes de 11 de setembro de 2026.

2. Antes dos comandos: o que voce vai ver

Plan 9 nao inicia em GNOME, KDE ou uma tela cheia de icones. O ambiente tradicional e o **rio**, um sistema de janelas simples em que cada janela costuma conter o shell **rc** ou uma aplicacao grafica. O mouse de tres botoes faz parte da linguagem do sistema.



Plan 9 Fourth Edition executando rio. Ha janelas com manual, Acme, arquivos e pequenos monitores. Captura: VulcanSphere. Software: Plan 9 Foundation. Licenca MIT/Expat. Fonte: Wikimedia Commons [17].

Como ler essa tela

Elemento	O que e	Analogia aproximada
rio	Sistema de janelas	Um gerenciador de janelas, mas tambem servidor de arquivos
rc	Shell padrao	O papel do Bash, com sintaxe e filosofia proprias
Acme	Editor e ambiente de trabalho	Editor + shell + hipertexto operacional
tag	Faixa de comandos de uma janela	Barra de acoes editavel
plumber	Roteador de mensagens	Decide quem abre um caminho, URL ou tipo de dado

Nao confunda austeridade com modo texto

Plan 9 foi concebido com graficos desde cedo. A interface e simples porque prioriza composicao, texto e mouse, nao porque seja apenas um terminal.

3. O mapa mental do acesso remoto

Em Unix, SSH costuma significar uma sessão remota dentro de um terminal. No Plan 9, a sessão remota pode carregar junto partes do seu ambiente local: teclado, mouse, tela e arquivos. Por isso **drawterm/rcpu** e mais próximo da ideia original do sistema do que um simples sshd.

Tres movimentos diferentes

Movimento	O que atravessa a rede	Resultado
ssh	Entrada, saída e erros do terminal	Shell ou comando em Debian/OpenBSD
sshfs	Operações SFTP traduzidas para 9P	Arquivos remotos em /n/ssh
rcpu / drawterm	Sessão, namespace local e dispositivos	Ambiente 9front remoto integrado

Uma analogia honesta

SSH e uma porta; rcpu e uma composição

SSH abre uma sessão. O rcpu executa no CPU server e exporta o contexto local para /mnt/term. Drawterm acrescenta a camada gráfica e se comporta como um terminal Plan 9 rodando fora do Plan 9.

O que o rcpu atual promete

- Conexão autenticada mutuamente e criptografada com TLS.
- Diretório atual, entrada, saída, erros, console e sinais de interrupção passam para o servidor.
- O namespace local aparece no lado remoto sob /mnt/term.
- A opção -p usa aad para proteger sessões contra quedas temporárias da conexão.

DRAWTERM

```
# [host Linux] sessão gráfica
drawterm -h 192.168.50.90 -a 192.168.50.90 -u pablo

# [host Linux] sessão somente texto e resiliente
drawterm -G -p -h 192.168.50.90 -a 192.168.50.90 -u pablo
```

Endereço de exemplo

O IP 192.168.50.90 é ilustrativo. Use o IP real da VM e mantenha o serviço em uma rede privada. Na sua configuração inicial, CPU e auth server podem ser a mesma VM.

4. SSH saindo do 9front: primeiro uso

Este é o caso mais familiar: você está no 9front e quer entrar no Forsaken, no Khan ou em outro Debian/OpenBSD. O comando do 9front atual fala SSH2, mas implementa um conjunto menor de algoritmos do que o OpenSSH. Comece com senha; depois migre para chave RSA armazenada no factotum.

Passo 1: testar a rede

TESTE DE REDE

```
# [9front]
ip/ping -n 3 192.168.50.119
```

Aqui usamos o IP conhecido do Forsaken apenas como exemplo. Se o ping falhar, SSH ainda pode funcionar caso ICMP esteja bloqueado, mas você precisa confirmar rota, endereço e firewall.

Passo 2: abrir a sessão

SSH INTERATIVO

```
# [9front]
ssh pablo@192.168.50.119
```

Passo 3: validar a chave do host

Na primeira conexão, o cliente calcula a impressão digital da chave RSA do servidor e procura em `$home/lib/sshthumbs`. Se a chave for desconhecida, ele mostra um comando para adicioná-la. **Não copie cegamente.** Compare a impressão digital por um segundo canal.

VERIFICAÇÃO

```
# [Debian, no console confiável do servidor]
ssh-keygen -lf /etc/ssh/ssh_host_rsa_key.pub

# [9front, somente depois de comparar]
# execute o comando exato sugerido pelo próprio cliente
# para acrescentar a impressão em $home/lib/sshthumbs
```

Passo 4: executar um comando sem abrir shell

COMANDO REMOTO

```
# [9front]
ssh pablo@192.168.50.119 uname -a
ssh pablo@192.168.50.119 uptime
```

Se houver erro de algoritmo

Não reative SSHv1 nem assinaturas SHA-1 apenas para fazer funcionar. Confirme se o servidor moderno possui uma chave RSA de host e aceita RSA/SHA-256. Se a interoperabilidade continuar ruim, use o Debian como ponte por outro meio e registre a limitação do cliente 9front.

5. sshfs: transformar o Debian em uma árvore local

Este é um dos exemplos mais claros da filosofia do Plan 9. Em vez de ficar preso a comandos como `scp`, o `9front` abre uma sessão SFTP e apresenta o diretório remoto como um sistema de arquivos montado. Depois disso, `ls`, `cat`, `cp` e o `Acme` trabalham sobre os dados remotos.

Montagem padrão

SFTP COMO 9P

```
# [9front]
sshfs pablo@192.168.50.119
ls /n/ssh
cat /n/ssh/README.md
```

Sem `-m`, o ponto de montagem padrão é `/n/ssh`. Caminhos relativos partem do diretório pessoal remoto.

Montagem explícita e somente leitura

MONTAGEM SEGURA

```
# [9front]
mkdir -p /n/forsaken
sshfs -R -m /n/forsaken pablo@192.168.50.119
ls /n/forsaken
cp /n/forsaken/notas/ideia.md $home/tmp/
```

Parte	Leitura
<code>-R</code>	Não permite alterações no servidor remoto
<code>-m /n/forsaken</code>	Escolhe um ponto de montagem fácil de reconhecer
<code>pablo@host</code>	Conta e endereço do servidor SFTP
<code>cp remoto local</code>	Copia um arquivo para o 9front

O detalhe técnico que vale aprender

O programa `sshfs` é ao mesmo tempo cliente SFTP e servidor de arquivos 9P local. Ele traduz operações de arquivo entre dois mundos. Nem toda operação 9P tem equivalente atômico em SFTP; portanto, uma falha durante alteração de metadados pode deixar uma mudança parcial. Para estudo e leitura, isso é excelente. Para dados críticos, continue usando ferramentas maduras no Debian.

Projeto imediato

Monte uma pasta read-only com cópias de textos do seu digital garden. Navegue e edite cópias no `Acme`, mas publique e faça backup pelo Debian.

6. sshnet: importar a rede de outra máquina

sshnet leva a ideia um passo além: ele usa encaminhamento do SSH e oferece a pilha TCP do servidor remoto como uma árvore no namespace do 9front. A resolução de nomes para novas conexões acontece do lado remoto.

O experimento

REDE VIA SSH

```
# [9front, em uma janela rio separada]
sshnet -- pablo@192.168.50.119

# novas conexões feitas por programas Plan 9 passam pela
# pilha de rede apresentada antes de /net
```

O resultado parece um proxy, mas a interface apresentada ao restante do sistema continua sendo /net. Programas não precisam aprender uma API especial de proxy: eles usam a mesma árvore de rede.

Por que testar em uma janela separada

- A montagem acontece antes de /net, mudando qual pilha de rede será encontrada primeiro.
- Um erro de DNS, autenticação ou tunel pode confundir outros testes naquele namespace.
- Fechar a janela encerra o experimento com menos chance de afetar o restante da sessão.

Um uso pessoal plausível

Imagine a VM 9front em uma rede isolada sem saída direta para a internet. Ela pode abrir uma sessão SSH para um Debian controlado e, dentro de um namespace específico, usar a rede desse Debian. O experimento ensina encaminhamento, namespaces e composição sem transformar o Plan 9 no roteador principal da casa.

Não use como VPN de produção

O objetivo aqui é observar como a abstração funciona. Para acesso sério a redes privadas, Tailscale, WireGuard, OpenSSH e as políticas do host Debian continuam sendo as ferramentas certas.

7. Entrando no 9front com drawterm

Drawterm roda no Windows, Linux e outros sistemas. Ele conecta ao CPU server e ao auth server, apresenta teclado, mouse e tela e exporta um diretório local para o namespace remoto. No 9front atual, a versão correspondente usa o protocolo **rcpu** e autenticação **dp9ik**.

Kamerad / Fedora	Tailscale	Roteador de sub-rede	VM 9front
drawterm	rede privada	Kaiser / Debian	rcpu :17019

No Fedora

DRAWTERM NO FEDORA

```
# [Fedora]
mkdir -p "$HOME/Plan9Share"
./drawterm -h 192.168.50.90 -a 192.168.50.90 \
+ -u pablo -r "$HOME/Plan9Share"
```

No Windows

DRAWTERM NO WINDOWS

```
# [PowerShell]
mkdir B:\Plan9Share
.\drawterm-amd64.exe -h 192.168.50.90 \
+ -a 192.168.50.90 -u pablo -r B:\Plan9Share
```

A sintaxe exata do caminho local pode variar conforme o binário. A ideia importante é usar `-r` para limitar o que o Plan 9 enxerga. Sem esse cuidado, drawterm pode apresentar uma raiz muito ampla do computador cliente.

Dentro da sessão

ARQUIVOS DO CLIENTE

```
# [9front remoto]
ls /mnt/term
cp /mnt/term/rascunho.md $home/notes/
acme $home/notes/rascunho.md
```

A primeira regra de segurança

Comece com uma pasta vazia e dedicada, nunca com seu perfil inteiro, `B:\NetShare` ou uma raiz contendo documentos empresariais. Aumente o escopo somente quando entender as permissões e os pontos de montagem.

8. Tailscale sem instalar Tailscale no Plan 9

Não conte com um agente Tailscale nativo e mantido na instalação base do 9front. No seu ambiente, a solução limpa é deixar o Proxmox ou outro Debian atuar como roteador de sub-rede. Assim, Kamerad, Hesse ou Poe alcançam o IP privado da VM pela tailnet, mas o 9front continua numa rede controlada.

Kamerad / Fedora	Tailscale	Roteador de sub-rede	VM 9front
drawterm	rede privada	Kaiser / Debian	rcpu :17019

Fluxo recomendado

- A VM 9front recebe um IP fixo ou reserva DHCP em uma VLAN de laboratório.
- Um no Debian/Proxmox anuncia somente essa sub-rede ao Tailscale.
- A ACL da tailnet permite apenas seus dispositivos pessoais acessarem a VM.
- O firewall deixa o rcpu/drawterm acessível somente a partir da rede de laboratório ou do roteador de sub-rede.
- A porta do serviço rcpu costuma ser 17019, conforme o arquivo de serviço citado pelo manual atual.

O que não fazer

Decisão	Motivo
Publicar 17019 diretamente na internet	Aumenta a superfície de ataque sem necessidade
Usar a VM como jump host principal	Ecossistema de hardening e auditoria é menor
Montar B:\NetShare inteiro pelo drawterm	Escopo excessivo para um laboratório
Guardar segredos empresariais na primeira VM	Você ainda estará aprendendo backups, usuários e namespaces

Separação boa

A VM aprende Plan 9. Debian e OpenBSD continuam protegendo e servindo o que importa. O laboratório pode ser refeito sem drama.

9. Rota A: aprender ferramentas com plan9port

Esta é a entrada de menor risco. Plan9port leva Acme, Sam, rc, mk, plumber e outras ferramentas para Linux. Você não ganha os namespaces do kernel Plan 9, mas consegue descobrir se gosta da linguagem de interação antes de montar a VM.

Instalação no Debian com interface gráfica

PLAN9PORT

```
# [Debian]
sudo apt update
sudo apt install -y git build-essential libfontconfig1-dev \
+ libx11-dev libxext-dev libxt-dev

mkdir -p "$HOME/opt"
git clone https://github.com/9fans/plan9port "$HOME/opt/plan9"
cd "$HOME/opt/plan9"
./INSTALL
```

Variáveis de ambiente

PRIMEIRO USO

```
# [Bash]
export PLAN9="$HOME/opt/plan9"
export PATH="$PATH:$PLAN9/bin"

# testar sem substituir seus comandos Unix
9 man intro
9 rc
9 acme
```

A documentação recomenda colocar \$PLAN9/bin no fim do PATH. Assim, comandos Plan 9 com nomes familiares não substituem por acidente cat, grep ou sed do sistema.

Onde fazer isso no seu caso

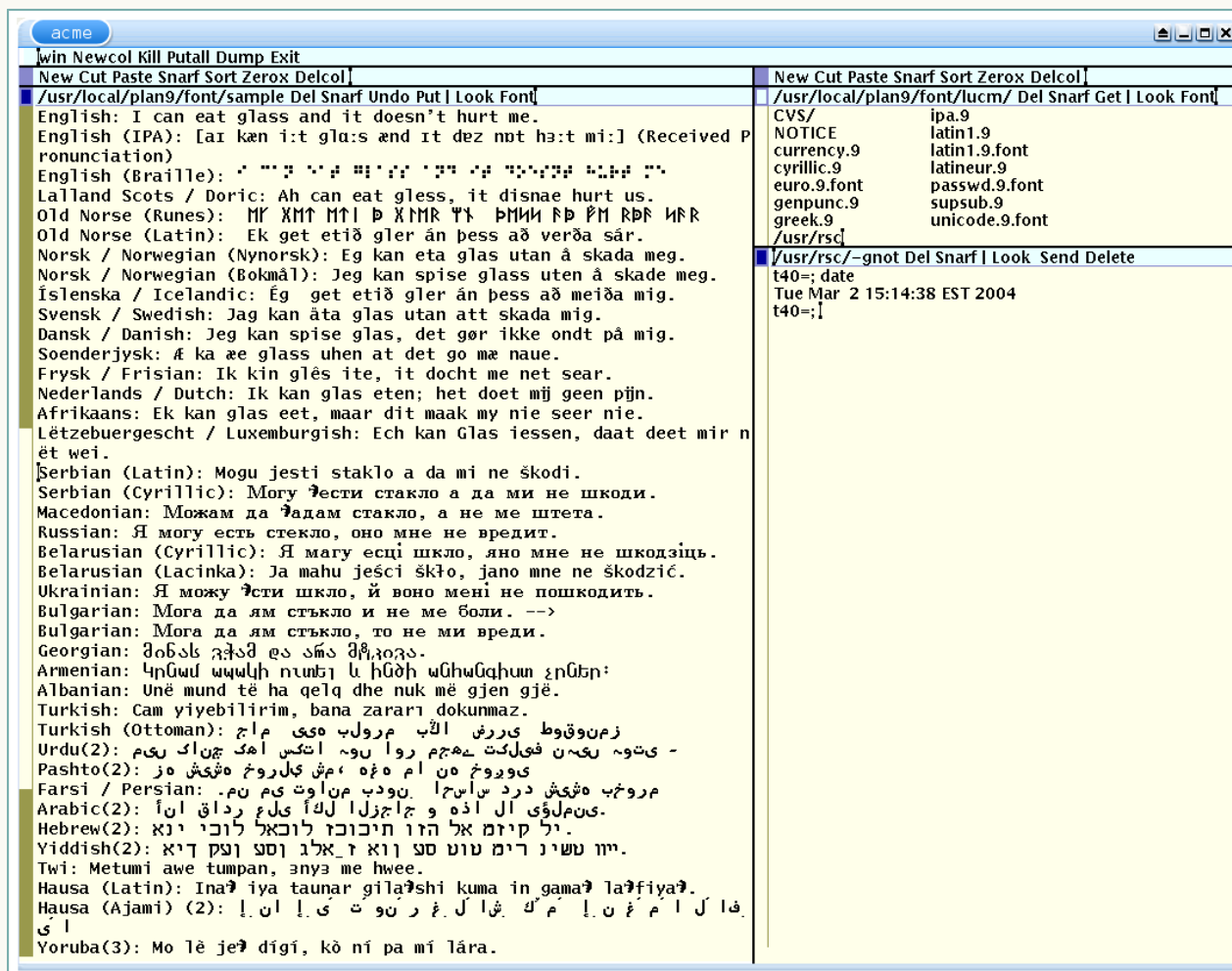
- Melhor primeira máquina: o notebook Fedora com ambiente gráfico ou outro desktop Linux.
- Debian headless serve para compilar e testar comandos textuais, mas Acme precisa de um servidor gráfico.
- Não instale primeiro em uma VPS de produção. O ganho é quase nulo e a chance de misturar PATHs é maior.

Meta da primeira noite

Abrir Acme, criar uma nota, executar um comando pelo botão 2, abrir um arquivo pelo botão 3 e entender por que texto pode ser interface.

10. Acme por dentro

Acme é o lugar em que a estranheza do Plan 9 começa a fazer sentido. As faixas no topo de cada janela são texto editável. Palavras como **Put**, **Get**, **Del** e **Look** são comandos, mas você também pode escrever seus próprios comandos ali.



Acme em Plan 9 from User Space, exibindo Unicode, uma árvore de arquivos e uma janela de comando. Fonte: galeria oficial do plan9port [12].

Os tres botoes

Botao	Acao basica	Exemplo
1: esquerdo	Selecionar texto	Marcar uma palavra ou trecho
2: meio	Executar texto	Executar date ou Put
3: direito	Procurar / abrir	Abrir caminho, arquivo ou definicao
1 + 2	Cortar a selecao	Equivale ao Cut por acorde
1 + 3	Colar	Equivale ao Paste por acorde

Sem botao do meio?

Em touchpad ou mouse de dois botoes, normalmente o botao do meio é emulado clicando os dois botoes juntos. Para aprender Acme sem friccao, use um mouse de tres botoes real.

11. Tutorial de Acme em 15 minutos

1. Abrir uma pasta

PREPARAR O LAB

```
# [plan9port]
mkdir -p "$HOME/plan9-lab/notas"
printf 'primeira nota\n' > "$HOME/plan9-lab/notas/hoje.txt"
9 acme "$HOME/plan9-lab/notas"
```

Acme abre a pasta como uma janela. O corpo mostra nomes; a tag mostra a pasta e comandos. Clique com o botão 3 em `hoje.txt` para abrir o arquivo.

2. Editar e salvar

- Clique no corpo com o botão 1 e digite uma linha.
- Na tag da janela, clique com o botão 2 sobre **Put**. O arquivo é salvo.
- Modifique o arquivo externamente e clique com o botão 2 em **Get** para recarregar.

3. Executar um comando

Escreva `date` na tag ou em qualquer corpo. Selecione a palavra e clique com o botão 2. A saída aparece numa janela de erros/comandos.

4. Abrir caminhos como hipertexto

Escreva o caminho completo de outro arquivo e clique com o botão 3. Acme pede ao plumber ou ao próprio ambiente para localizar e abrir o alvo.

5. Criar uma janela de shell

Clique com o botão 2 em **New**, escreva `win` na tag e execute com o botão 2. No `plan9port`, isso abre um shell dentro de uma janela Acme.

TAG COMO PALETA

```
# texto que voce pode manter na tag
date  pwd  git status  rg TODO

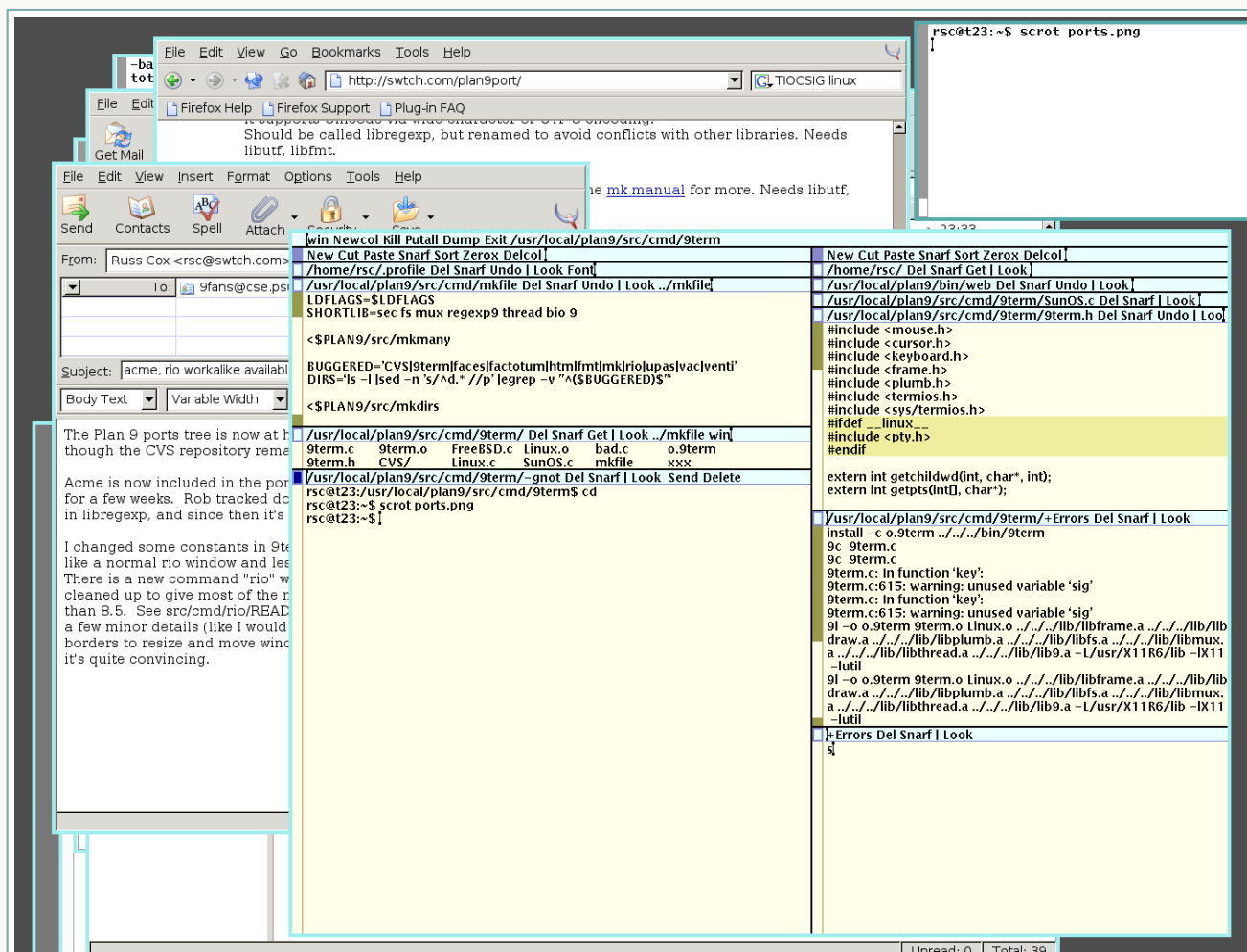
# selecione somente o comando desejado e use o botão 2
```

O estalo conceitual

No Acme, você não precisa esperar o autor do editor criar um botão. Qualquer comando textual pode virar uma ação contextual. Isso é poderoso, mas exige critério para não executar texto sem entender.

12. Plan9port convivendo com Linux

Voce nao precisa adotar um desktop Plan 9 completo. Acme, 9term e rio podem conviver com Firefox, e-mail e ferramentas normais do Linux. Essa combinacao e provavelmente mais util para voce do que tentar navegar na web moderna dentro do 9front.



Plan 9 from User Space: rio e Acme convivendo com programas X11 tradicionais. Fonte: galeria oficial do plan9port [12].

Um fluxo pessoal realista

- Escrever Markdown e código no Acme.
- Usar Firefox e ferramentas de comunicação no Linux.
- Executar builds do Zensical, Quartz ou seus projetos pelo shell hospedado no Acme.
- Manter Docker, bancos de dados e credenciais fora do 9front.

A melhor forma de adoção

Use as ideias do Plan 9 onde elas melhoram seu trabalho. Não transforme a experiência em prova de pureza.

13. rc: um shell pequeno, mas diferente

O shell rc parece familiar até você tratar variáveis e citações como faria no Bash. A diferença central é que uma variável é naturalmente uma **lista de palavras**. Isso deixa vários scripts simples, mas exige abandonar alguns reflexos.

Variáveis e listas

RC

```
# [9front ou plan9port dentro de rc]
servidores=(nyx kaiser khan forsaken)
for(s in $servidores) {
    echo servidor: $s
}
```

Capturar a saída de um comando

SUBSTITUIÇÃO

```
arquivos=`ls *.md`
for(f in $arquivos) {
    echo $f
    wc $f
}
```

Condição simples

CONDIÇÃO

```
if(test -e notas.txt) {
    echo existe
}
if not {
    echo não existe
}
```

Quatro diferenças que evitam frustração

Tema	rc	Bash
Variável	Lista de palavras	String com expansões contextuais
Atribuição	nome=valor	nome=valor
Bloco	{ ... }	then ... fi ou { ...; }
Substituição	`{comando}`	\$(comando)
Status	Texto vazio indica sucesso	Código inteiro 0 indica sucesso

Primeira semana

Escreva scripts de dez linhas. Não tente portar seu .bashrc inteiro. Aprenda listas, citação, pipes e status antes de automatizar algo importante.

14. Rota B: uma VM 9front no Kaiser

Para experimentar o sistema inteiro, use uma VM descartável no Proxmox. A recomendação do estudo anterior continua boa: **2 vCPUs, 2 GB de RAM e 16 GB de disco**. O 9front precisa de muito menos, mas esses valores deixam folga e não pesam no Kaiser.

Configuração inicial sugerida

Item	Escolha inicial	Se der problema
Arquitetura	amd64	Use a imagem 386 apenas por compatibilidade
BIOS / máquina	SeaBIOS + i440fx	Evite recursos modernos desnecessários
CPU	2 vCPUs, tipo host	1 vCPU também basta para aprender
Memória	2048 MB	512 MB a 1 GB ainda é viável
Disco	16 GB VirtIO Block	Tente SATA/IDE se o disco não aparecer
Rede	VirtIO em bridge/VLAN de lab	Tente E1000 se a NIC não aparecer
Display	Standard VGA	Use resolução simples no primeiro boot

Dois formas de chegar ao primeiro boot

- **Mais rápida:** importar a imagem `amd64.qcow2.gz` gerada no build noturno.
- **Mais didática:** baixar a ISO `amd64`, iniciar a VM e executar `inst/start` no shell do ambiente live.

INSTALADOR

```
# [9front live]
inst/start

# siga a ordem apresentada pelo instalador
# e escolha uma instalação local para o primeiro laboratório
```

Depois da instalação

- Reinicie sem a ISO e confirme que rio abre.
- Teste DHCP e depois `ip/ping -n 3 gateway`.
- Crie um snapshot Proxmox chamado `base-limpa`.
- Só depois configure CPU/auth server e drawterm.

Fonte da imagem

Em 12 de setembro de 2026, o servidor de builds oferecia ISO e qcow2 amd64 noturnos. Baixe da página oficial, confira hash quando fornecido e guarde a imagem usada no registro do laboratório.

15. A primeira hora dentro do 9front

Não tente descobrir o sistema abrindo diretórios ao acaso. Siga uma sequência: identidade, ajuda, arquivos, processos, namespace, rede, editor e desligamento.

Minuto	Comando	Pergunta que responde
0-5	cat /dev/user	Quem sou eu?
5-10	cat /dev/sysname	Qual é o nome desta máquina?
10-15	pwd; lc	Onde estou e o que existe aqui?
15-20	man 1 intro	Como os comandos são organizados?
20-25	ps	Quais processos estão rodando?
25-30	ns	De onde veio cada parte do meu namespace?
30-35	cat /net/ndb	Qual é o estado atual da rede?
35-40	ip/ping -n 3 gateway	Consigo alcançar a rede local?
40-50	acme	Como texto, comandos e arquivos se encontram?
50-60	fshalt	Como desligo preservando o sistema de arquivos?

ORIENTAÇÃO

```
# [9front]
cat /dev/user
cat /dev/sysname
pwd
lc
ps
ns
cat /net/ndb
```

Ajuda que você deve usar sempre

MANUAIS

```
man 1 intro
man rc
man rio
man acme
man ssh
man sshfs
man rcpu
```

Comandos parecidos não são idênticos

ls, grep, sed e outros nomes existem, mas opções e expressões regulares podem divergir do GNU/Linux. Quando uma opção conhecida falhar, leia o manual do Plan 9 antes de concluir que o programa é limitado.

16. Laboratório de namespace: duas pastas, uma visao

Namespaces ficam abstratos até você montar uma composição com as próprias mãos. Neste exercício, dois diretórios de origem aparecem juntos em um terceiro diretório. Não há cópia de arquivos.

Preparar as origens

PREPARAÇÃO

```
# [9front]
mkdir -p /tmp/origem-a /tmp/origem-b /tmp/visao
echo 'veio de A' > /tmp/origem-a/a.txt
echo 'veio de B' > /tmp/origem-b/b.txt
```

Compor a visao

UNIÃO

```
bind /tmp/origem-a /tmp/visao
bind -a /tmp/origem-b /tmp/visao

lc /tmp/visao
cat /tmp/visao/a.txt
cat /tmp/visao/b.txt
```

O que aconteceu

Caminho visto	Origem real
/tmp/visao/a.txt	/tmp/origem-a/a.txt
/tmp/visao/b.txt	/tmp/origem-b/b.txt

bind substituiu a visao inicial do ponto de montagem. bind -a acrescentou a segunda árvore depois da primeira. Outro processo pode receber uma composição diferente sem obrigar o sistema inteiro a enxergar a mesma coisa.

A aplicação pessoal

Seu editor pode receber uma visao composta de notas-públicas e rascunhos, enquanto um publicador recebe apenas notas-públicas. Em vez de espalhar checagens de permissão pela aplicação, você limita o mundo que o processo consegue nomear.

Faca em /tmp

O laboratório usa somente diretórios descartáveis. Quando terminar, feche a janela/sessão ou desmonte a composição. Não experimente primeiro sobre \$home, /bin ou dados montados do host.

17. Rede como arquivos: olhando /net

No Plan 9, abrir uma conexão TCP não começa com a API BSD de sockets. O programa conversa com arquivos de controle e dados dentro de /net/tcp. O protocolo 9P permite que essa árvore seja local, importada ou fornecida por outro processo.

INSPEÇÃO

```
# [9front]
lc /net
lc /net/tcp
cat /net/ndb

# depois de abrir conexões, diretórios numéricos representam
# conversas TCP; cada um pode conter ctl, data e status
```

Modelo simplificado de uma conversa

Arquivo	Papel
/net/tcp/clone	Cria uma nova conversa e devolve um identificador
/net/tcp/N/ctl	Recebe comandos como connect e announce
/net/tcp/N/data	Transporta os bytes da conexão
/net/tcp/N/status	Explica o estado atual

Exemplo de HTTP sem esconder a árvore

WEB

```
# [9front]
webfs
hget https://example.com > /tmp/example.html
page /tmp/example.html
```

webfs apresenta requisições web por arquivos em /mnt/web. hget fornece uma interface simples para baixar. Para sites modernos com JavaScript pesado, autenticação complexa ou DRM, use um navegador no Linux.

O aprendizado

O ganho não é fazer HTTP melhor que curl. É perceber que uma árvore de arquivos pode ser uma interface operacional para rede, janelas, processos e serviços.

18. 9P em um exemplo pessoal

Imagine um pequeno servidor 9P rodando no Debian e expondo o estado do homelab. Ele não precisa fornecer arquivos armazenados em disco. Cada leitura pode consultar Gatus, Docker ou uma API local e devolver um texto curto.

ÁRVORE SINTÉTICA

```
/srv/homelab/
|-- nyx/
|   |-- status      # texto: ok / alerta
|   |-- temperatura # texto: 48 C
|   `-- uptime      # texto: 12d 04h
|-- kaiser/
|   |-- status
|   `-- vms
`-- radio/
    |-- agora
    `-- ouvintes
```

Como o cliente usaria

CONSUMO

```
# [9front, depois de montar o serviço]
cat /n/homelab/nyx/status
cat /n/homelab/radio/agora
for(s in nyx kaiser) {
    echo -n $s': '
    cat /n/homelab/$s/status
}
```

Por que isso é diferente de uma API REST

API web	Serviço 9P
URL + método + JSON	Caminho + abrir + ler/escrever
Cliente interpreta esquema	Árvore comunica a organização
Biblioteca HTTP	Operações normais de arquivo
Autorização por endpoint	Namespace pode ocultar ramos inteiros

Comece somente leitura

Não crie restart, delete ou controles de produção na primeira versão. Status, logs curtos e dados sintéticos já ensinam quase tudo com risco muito menor.

19. Compilar um programa pequeno

O dialeto C do Plan 9 usa cabecalhos e convenções próprias. No amd64, os compiladores tradicionais recebem nomes ligados a arquitetura. Este exemplo imprime uma linha e termina com status de sucesso.

HELLO.C

```
#include <u.h>
#include <libc.h>

void
main(void)
{
    print("Ola do Plan 9!\n");
    exits(nil);
}
```

Compilar no 9front amd64

COMPILACAO

```
# [9front]
6c hello.c
6l -o hello hello.6
./hello
```

Compilar com plan9port

PORT PARA UNIX

```
# [Linux + plan9port]
9c hello.c
9l -o hello hello.o
./hello
```

Linha por linha

Linha	Significado
	Tipos básicos usados pelo ambiente Plan 9
	Rotinas como print e exits
void main(void)	Ponto de entrada no estilo Plan 9
print	Saída formatada UTF-8
exits(nil)	Finaliza com status de sucesso

Proximo passo certo

Antes de escrever um servidor 9P, faça três programas: ler um arquivo, percorrer um diretório e transformar texto. Depois estude lib9p e um servidor sintético mínimo.

20. Ideias para uso pessoal, sem fantasia

Nem toda ideia interessante precisa virar infraestrutura. A tabela separa valor de aprendizado, utilidade cotidiana e risco para você escolher algo que sobreviva ao entusiasmo inicial.

Ideia	Aprendizado	Utilidade	Risco	Começar onde
Acme para notas e código	Alto	Alta	Baixo	plan9port
Estação Plan 9 em VM	Muito alto	Média	Baixo	Kaiser
Painel homelab em 9P	Muito alto	Média	Baixo se read-only	Debian + 9front
Rascunhos do digital garden	Alto	Alta	Baixo	plan9port
Catálogo fotográfico X100VI	Alto	Média	Baixo	Debian + 9P
Console Murad Radio	Alto	Média	Médio	read-only primeiro
Pubnix Plan 9 privado	Muito alto	Baixa	Médio	rede interna
Servidor público de produção	Baixo	Baixa	Alto	não fazer

Minha recomendação, em ordem

- 1. Use Acme por uma semana no Linux para notas e pequenos projetos.
- 2. Crie a VM 9front no Kaiser e aprenda rio, rc, namespaces e rede.
- 3. Ative drawterm somente na rede privada e restrinja /mnt/term.
- 4. Construa um servidor 9P read-only para status do homelab.
- 5. So depois separe CPU, auth e file server em VMs descartáveis.

Critério de sucesso

Depois de 30 dias, o projeto deve ter produzido conhecimento documentado, não mais um serviço frágil que você precisa manter.

21. Projeto 1: estacao de rascunhos do digital garden

Use plan9port no Fedora para escrever Markdown no Acme. O repositório e o build continuam no Linux, onde Zensical, Quartz, Git e navegadores modernos funcionam bem. Plan 9 entra como ambiente de escrita e composição, não como hospedeiro público.

ESTRUTURA

```
notes/  
|-- inbox/  
|-- diario/  
|-- projetos/  
|-- humanas/  
|-- tecnicas/  
`-- publicavel/
```

Fluxo de trabalho

- Abrir notes/ no Acme e navegar com o botão 3.
- Manter na tag comandos como `rg TODO`, `git diff` e o build local.
- Mover para publicavel/ somente textos revisados.
- Publicar pelo pipeline normal do Debian/Git, não pelo 9front.

Experimento de namespace

Na VM, monte publicavel/ em uma visão de leitura e uma inbox/ apenas no namespace do editor. O publicador nunca precisa receber o caminho dos rascunhos.

Entrega após uma semana

Resultado	Como avaliar
10 notas editadas	Acme ajudou ou atrapalhou?
1 regra do plumber	Um caminho ou URL abriu no destino certo?
1 script rc	Automatizou uma tarefa curta sem imitar Bash?
Diário de fricções	Registrou o que falta para uso real?

Valor real

Esse projeto conversa com seu interesse por digital gardens e small web sem colocar seu site, banco ou certificados em risco.

22. Projeto 2: painel do homelab como sistema de arquivos

O Debian coleta dados de Gatus, Docker, Proxmox ou scripts locais. Um pequeno servidor 9P transforma esses dados em arquivos de texto. A VM 9front monta a árvore e vira um console silencioso para Nyx, Kaiser, Khan e seus serviços.

NAMESPACE DO PAINEL

```
/n/homelab/  
|-- nyx/{status,uptime,discos}  
|-- kaiser/{status,vms,carga}  
|-- khan/{status,containers}  
|-- forsaken/{status,talk}  
`-- alertas/ativos
```

Fase 1: read-only

- Arquivos retornam texto curto, sem JSON quando não for necessário.
- Cada leitura possui timeout e nunca bloqueia a árvore inteira.
- Falha de uma fonte vira conteúdo desconhecido, não pânico do servidor.
- Credenciais ficam no Debian; o cliente 9front vê apenas resultados filtrados.

Fase 2: historico simples

Um processo no Debian grava snapshots em intervalos. No 9front, o diretório `/n/homelab/historico/AAAA-MM-DD/` apresenta uma visão arquivada. Isso permite comparar estados usando ferramentas de texto.

Fase 3: comandos controlados

Somente depois de meses de uso, considere um arquivo `ctl`. Cada comando deve ser enumerado, autenticado, registrado e limitado. Para produção, continue preferindo as interfaces administrativas originais.

A versão que eu faria

Somente status, uptime, temperatura, espaço e número de containers. Nada de restart. O ganho é aprender 9P e produzir um painel autoral, não substituir Portainer ou Proxmox.

23. Projeto 3: arquivo fotografico X100VI

O processamento RAW continua no seu fluxo moderno. O experimento 9P atua como catálogo: um servidor Debian lê EXIF e metadados, gera miniaturas e apresenta uma árvore navegável. O 9front ou Acme viram uma interface de consulta.

CATALOGO 9P

```
/n/fotos/  
|-- 2026/  
|   |-- 09/  
|       |-- 12/  
|           |-- DSCF0001/  
|               |-- preview.jpg  
|               |-- exif  
|               |-- tags  
|               |-- nota  
|           |-- DSCF0002/  
|-- cameras/X100VI/  
|-- selecoes/favoritas/
```

Consultas simples

CONSULTA

```
# [9front]  
cat /n/fotos/2026/09/12/DSCF0001/exif  
cat /n/fotos/2026/09/12/DSCF0001/tags  
page /n/fotos/2026/09/12/DSCF0001/preview.jpg
```

O que fica fora

- RAF originais, backups e catálogo principal não dependem do 9front.
- Edição de cor, ruído, lente e exportação continuam no software fotográfico adequado.
- A árvore 9P oferece uma leitura autoral e programável do acervo.

Por que combina com Plan 9

Metadados, imagens, seleções e notas tornam-se caminhos previsíveis. Um script simples consegue navegar sem SDK específico de um catalogador.

24. Projeto 4: uma janela Plan 9 para a Murad Radio

Liquidsoap, Icecast, Postgres e a interface web continuam onde já fazem sentido. O projeto Plan 9 oferece uma segunda forma de observar a radio: uma árvore read-only voltada a automação e curiosidade.

RADIO COMO ARQUIVOS

```
/n/radio/  
|-- agora  
|-- proxima  
|-- ouvintes  
|-- historico  
|-- caller/estado  
`-- stream/url
```

MONITOR MINIMO

```
# [9front]  
while(sleep 5) {  
    cat /n/radio/agora  
    cat /n/radio/ouvintes  
}
```

Primeira versao

- Ler faixa atual, próxima faixa e número de ouvintes.
- Exibir estado do caller sem permitir colocá-lo no ar.
- Manter a URL pública como texto para o plumber abrir no player adequado.

Segunda versao, se a primeira provar valor

Um namespace administrativo poderia conter controles, mas ele nunca deve ser montado na sessão comum. Mesmo assim, a interface web e as rotinas de segurança existentes continuam sendo a fonte de verdade.

O charme

Você teria uma pequena janela rio que parece um equipamento de estúdio antigo, mas lê dados reais da rádio. É um uso autêntico da estética e da arquitetura, não uma tentativa de substituir toda a stack.

25. Projeto 5: estacao-museu e pubnix privado

Uma VM ou maquina pequena pode funcionar como ambiente permanente de preservacao. Amigos entram por drawterm, leem textos, usam Acme, experimentam rc e deixam notas. O acesso fica restrito a uma rede privada ou a uma rota Tailscale controlada.

Servicos que fazem sentido

- Biblioteca de documentos historicos do Bell Labs e manuais locais.
- Directorio read-only com seus diarios de laboratorio.
- Mural simples baseado em arquivos, com moderacao manual.
- Pequenos programas, jogos e desafios em C/rc.
- Sessao Acme preparada para explicar botoes, plumber e namespaces.

Servicos que nao fazem sentido

- E-mail publico, banco de dados empresarial, certificados ou autenticacao central.
- Proxy reverso de sites, Docker, Synapse, PeerTube ou Jellyfin.
- Conta aberta anonima sem limites, logs ou plano de recuperacao.

Regra de moderacao

Se outras pessoas puderem publicar, separe uma area de entrada de uma area publica. Novos textos ficam em quarentena ate aprovacao. O namespace do visitante nao inclui os comandos administrativos.

Nome possivel

Plan 9 Station dentro do seu ecossistema: pequena, intencional, estranha e documentada. O valor e cultural e educativo.

26. Roteiro de 30 dias

Período	Foco	Entrega
Dias 1-3	plan9port, Acme e mouse	3 notas e 1 arquivo de código
Dias 4-7	rc, manuais e plumber	1 script e 1 regra de abertura
Dias 8-10	VM 9front limpa	Snapshot base-limpa
Dias 11-14	rio, rede e arquivos	Diário dos 20 comandos usados
Dias 15-18	drawterm em rede privada	Acesso Fedora/Windows com -r restrito
Dias 19-22	ssh, sshfs e sshnet	Tres laboratorios documentados
Dias 23-26	namespaces e 9P	Arvore sintetica desenhada
Dias 27-30	prototipo read-only	Status de 2 maquinas via 9P

Diário mínimo

MODELO DE NOTA

AAAA-MM-DD

Objetivo:

Comandos usados:

0 que eu esperava:

0 que aconteceu:

0 que o Plan 9 simplificou:

0 que o Plan 9 complicou:

Proximo teste:

Ao final, responda

- Acme melhora sua escrita ou so e interessante?
- Voce consegue explicar namespace sem usar a palavra container?
- Consegue mostrar a diferenca entre sshfs e montar um 9P nativo?
- O prototipo read-only tem valor alem da demonstracao?
- Quais partes merecem permanecer e quais devem ser apagadas?

Resultado aceitavel

Concluir que apenas Acme vale a pena ja e um bom resultado. Concluir que o sistema inteiro merece um canto permanente no Kaiser tambem. O erro seria forcar utilidade onde existe apenas curiosidade.

27. Onde parar

Plan 9 recompensa curiosidade, mas pune expectativas erradas. O critério abaixo evita transformar um laboratório elegante em dívida operacional.

Necessidade	Use	Não force
Servidor web público	Debian/OpenBSD + Nginx/Caddy	rc-httpd como substituto automático
Containers e bancos	Docker + PostgreSQL/MariaDB	Portar stack moderna
Desktop diário	Fedora/Windows	9front para navegação e trabalho corporativo
Acesso remoto comum	OpenSSH/Tailscale	sshd histórico do Plan 9
Estudo de distribuição	9front + drawterm + 9P	Esconder tudo atrás de ferramentas Linux
Editor autoral	Acme via plan9port	Trocar todo o sistema sem necessidade

Sinais de que o projeto está saindo do trilho

- Você precisa enfraquecer criptografia moderna para compatibilidade.
- A VM passou a guardar a única cópia de algo importante.
- Um serviço de produção depende de um port pouco mantido.
- Você está recriando Docker, systemd, um navegador ou um banco dentro do Plan 9.
- A documentação do experimento ficou para depois.

A leitura correta

Plan 9 é uma lente. Ele mostra que nomes, arquivos, rede, identidade e interface podem ser compostos de outra maneira. Algumas ideias podem melhorar seus projetos atuais mesmo que nenhum deles rode no Plan 9.

Veredito

Coloque Plan 9 ao lado do seu ecossistema. Deixe Debian, OpenBSD e Windows fazerem o trabalho pesado. Use 9front para pensar, experimentar e construir pequenas coisas com personalidade.

28. Folha de consulta rapida

Quero...	Comando / ferramenta
ver o usuario	cat /dev/user
ver o nome da maquina	cat /dev/sysname
listar em colunas	lc
ler o manual	man comando
ver processos	ps
ver o namespace	ns
testar rede	ip/ping -n 3 host
ver configuracao de rede	cat /net/ndb
editar / trabalhar	acme
entrar num Debian	ssh usuario@host
montar SFTP	sshfs -R -m /n/nome usuario@host
usar TCP remoto	sshnet -- usuario@host
entrar no 9front	drawterm -h cpu -a auth -u usuario
ver arquivos do cliente	ls /mnt/term
desligar corretamente	fshalt

Arquivos e caminhos que reaparecem

Caminho	Significado
\$home	Diretorio pessoal
/dev/user	Identidade atual
/net	Pilha de rede apresentada como arquivos
/n	Convencao para montagens de recursos remotos
/srv	Descritores de servicos 9P publicados
/mnt/term	Arquivos e dispositivos exportados pelo terminal/drawterm
/mnt/factotum	Interface do agente de autenticao

Comando mais importante

man. Em Plan 9, a documentacao local e parte central da experiencia, e as diferencas de versao importam.

29. Fontes, data e créditos

Pesquisa verificada em **12 de setembro de 2026**. Para a pergunta sobre SSH, a fonte principal foi a árvore atual do 9front: manual, código e lista de programas. A ausência de um servidor OpenSSH compatível foi tratada como inferência da árvore atual, não como afirmação histórica universal.

Sobre comandos

O 9front evolui. Antes de copiar uma configuração sensível, leia o manual da imagem instalada. Exemplos deste guia usam endereços privados e nomes ilustrativos.

Referências técnicas

- [1] **Plan 9 from Bell Labs - Overview**
<https://plan9.io/plan9/about.html>
- [2] **Plan 9 from Bell Labs - artigo técnico do sistema**
<https://plan9.io/sys/doc/9.html>
- [3] **Plan 9 manual - protocolo 9P**
<https://plan9.io/magic/man2html/5/0intro>
- [4] **The Use of Name Spaces in Plan 9**
<https://plan9.io/sys/doc/names.html>
- [5] **9front - servidor oficial de builds noturnos**
<https://build.9front.org/>
- [6] **9front - árvore espelho atual no GitHub**
<https://github.com/9front/9front/tree/front>
- [7] **9front - manual atual do cliente SSH2**
<https://github.com/9front/9front/blob/front/sys/man/1/ssh>
- [8] **9front - código-fonte atual do cliente SSH2**
<https://github.com/9front/9front/blob/front/sys/src/cmd/ssh.c>
- [9] **9front - manual atual do sshfs**
<https://github.com/9front/9front/blob/front/sys/man/4/sshfs>
- [10] **9front - manual atual do sshnet**
<https://github.com/9front/9front/blob/front/sys/man/4/sshnet>
- [11] **9front - manual atual do rcpu/rimport/rexport**
<https://github.com/9front/9front/blob/front/sys/man/1/rcpu>
- [12] **9front drawterm - manual e código**
<https://github.com/9front/drawterm/blob/front/drawterm.1>
- [13] **Plan 9 from User Space - página, instalação e manuais**
<https://9fans.github.io/plan9port/>
- [14] **Plan 9 from User Space - notas oficiais de instalação**
<https://9fans.github.io/plan9port/man/man1/install.html>
- [15] **Plan 9 from User Space - galeria de screenshots**
<https://9fans.github.io/plan9port/screenshots/>
- [16] **Plan 9 original - manual histórico de ssh1/sshserve**
<https://9p.io/sources/plan9/sys/man/1/ssh1>
- [17] **Wikimedia Commons - screenshot de rio**
https://commons.wikimedia.org/wiki/File:Plan_9_Fourth_Edition_rio_interaction_screenshot.png
- [18] **Linux kernel - cliente v9fs/9P**
<https://docs.kernel.org/filesystems/9p.html>
- [19] **Acme: A User Interface for Programmers**
<https://plan9.io/sys/doc/acme/acme.html>
- [20] **FQA do 9front - referência geral**
<https://fqa.9front.org/fqa.html>

Documento-base: plan-9-guia-completo(1).pdf, fornecido por Pablo Murad. Este complemento preserva seu veredito: Plan 9 é excelente para estudo, hobby e preservação, mas Debian e OpenBSD continuam sendo escolhas superiores para produção.

Crédito visual: a captura de rio da Wikimedia Commons foi feita por VulcanSphere e distribuída conforme a licença MIT/Expat indicada na página do arquivo. As capturas de Acme e rio/plan9port vieram da galeria oficial do projeto Plan 9 from User Space.