

DeforaOS: Un voyage dans le développement de système d'exploitation

MiNET
Telecom SudParis
24 mai 2018

Pierre Pronchery <khoben@defora.org>



Commençons par moi

- Pierre Pronchery
- Consultant indépendant en sécurité informatique depuis 2007
- Basé à Berlin (Allemagne)
- Développeur Open Source, notamment DeforaOS
- GitHub: `khorben` et DeforaOS



Qu'est ce que DeforaOS?

- Projet de système d'exploitation
- 3 parties:
 1. Système distribué
Objectif principal du projet
 2. Environnement de bureau
Pour expérimenter et illustrer les concepts
 3. Environnement auto-hébergé (self-hosting)
Utile, passionnant, et j'ai des idées là aussi



Quelques mots sur l'auto-hébergement

- libc multi-plateformes:
 - Linux, NetBSD, OpenBSD, FreeBSD, Solaris, MinGW...
 - Intel x86, amd64, ARM 32-bits, SPARC 32 et 64 bits...
- Framework de compilation:
 - Bibliothèque de pre-processing (DeforaOS CPP)
 - Assembleur modulaire ELF, PE... (DeforaOS Asm)
 - Compilateur C (DeforaOS C99)
 - Debugging, Reverse-Engineering



Framework distribué

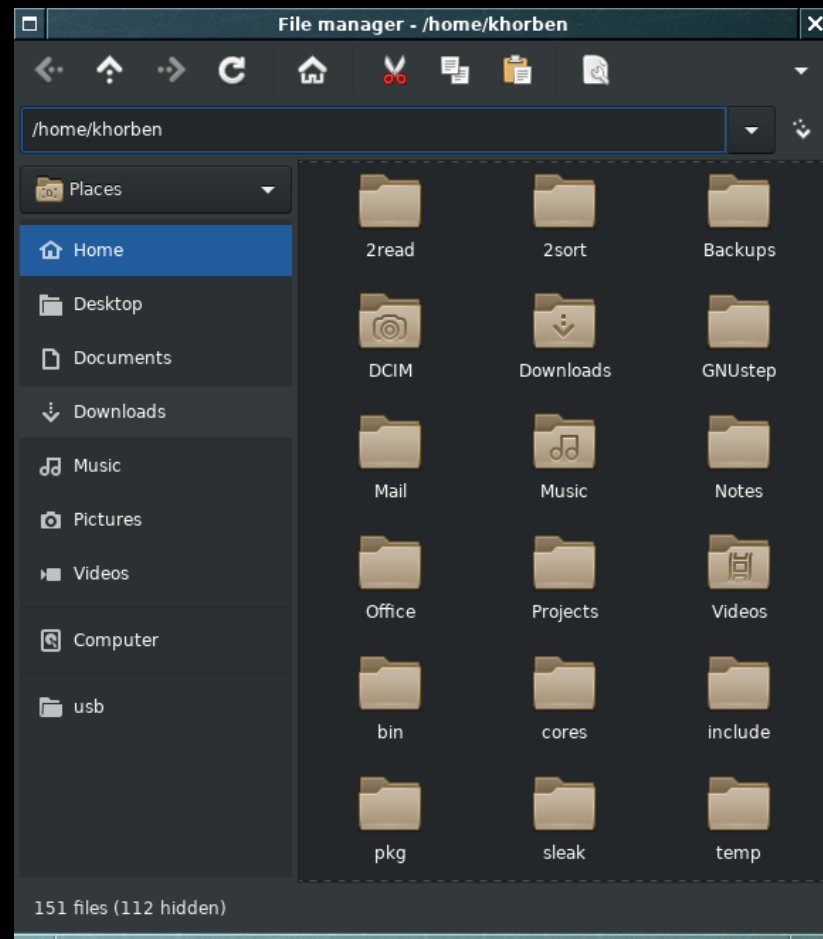
- Init
- Session
- VFS
- VPN
- Probe (monitoring)
- GServer (serveur graphique)

Je vais expliquer



Environnement de bureau

- Applications graphiques:
 - Gestion de fichiers
 - Panel
 - Navigateur web
 - Visionneuse PDF
 - Préférences (volume, wifi...)
 - Téléphonie...
- Gtk+ 2 et 3



L'histoire

« Un peuple qui oublie son passé
se condamne à le revivre »

– *Winston Churchill*



1999

- Au lycée
- Repris le club informatique local (CCCM)
- Internet 56K (musique!)
- Premier CD d'installation Debian GNU/Linux (2.0 "hamm")
- Mes parents se demandent ce que je fais de ma vie avec un pingouin
- Auto-formation en C, HTML, Javascript...
- D'autres étudiants squattent *nos machines*
- On veut jouer à DOOM aussi
- Ping flood
- Premiers exploits



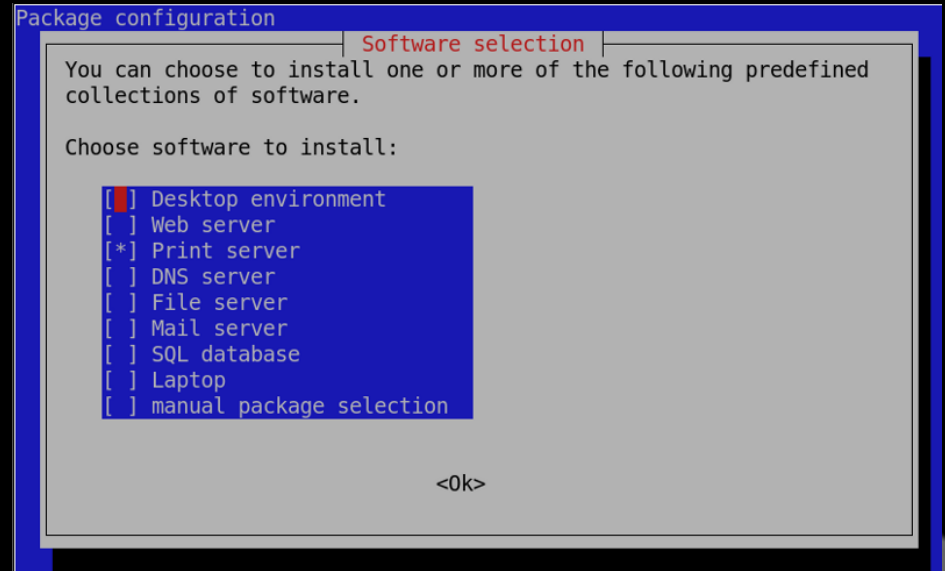
2000-2002

- En IUT Génie Logiciel
- Premier jour, « qui connaît Linux? »
- Terminaux X sur Solaris:
 - Un serveur SPARC, 4 CPUs
 - 120 sessions FVWM, Netscape 4.7, C++, Java
 - Et moi qui compile et joue à DOOM en mode fenêtré
 - Solide, sauf pour “xhost +” par défaut



Premières frustrations, premières innovations

- Debian c'est le bien TM
- L'installation est compliquée,
sans bonne raison
<entrée> <entrée> <entrée>
<entrée> <entrée> <entrée>
<entrée> <entrée> <entrée>
<entrée> <entrée> <entrée>
<entrée> <entrée> <entrée>
- Je suis perfectionniste...



Defora GNU/Linux

- « DEbian FOR All »
(voilà maintenant vous savez)
- Live-CD avant l'heure :
 - X par défaut, installeur Gtk+, partitionne et formate /dev/hda1, décompresse une archive tar.gz avec bureau GNOME
 - Évidemment, une autre version avec Quake II, Unreal Tournament...
 - CD-RW 4x, El Torito avec disquette de boot embarquée (1.44 MB!)
 - Premier site web avec PHP-Nuke



2002-2003

- Oxford Brookes University, Bsc Computer Science
- Cours d'administration UNIX « create a minimal University Network »
 - Debian : trop facile
 - OpenBSD : il manque des morceaux
 - Ce sera Linux From Scratch (LFS)
- Professeur sysadmin « have fun on the University's network »
 - ARP spoofing
 - DNS hijacking
 - SSL MitM (Man-in-the-Middle)



Defora GNU/Linux, sans Debian

- Le nom est tout de même resté
- Release « oxford »
- pkgr, gestionnaire de packages écrit en C++
 - Mon premier et dernier projet en C++
 - En-tête meta-data et tarball, le tout compressé en BZIP2
- Deuxième release avortée :
« Le packaging ? Facile, presque tout utilise GNU Autotools! »



GNU Autohell

- En théorie, portable et éprouvé
- Personne ne l'utilise de la même façon
- Quasiment personne ne s'en sert *correctement*
- Complexe.
- Pas portable dès qu'on l'étend



DeforaOS configure

- Tout DeforaOS utilise de simples Makefiles
- Générés pour GNU et BSD make :
 - C/C++, assembleur, Objective-C, Verilog
 - Libtool (optionnel)
 - Compilation hors des sources (optionnelle)
 - Scripts pour documentation, traductions, tests...
- C'est dans Apps/Devel/src/configure, ou configure.git



2003-2005

- INSIA, école privée en ingénierie informatique (alternance)
- Cours de développement système : mini-libc
- Pourquoi s'arrêter là ?
- DeforaOS libc :
 - Linux, NetBSD, OpenBSD, FreeBSD, Solaris, macOS...
 - Intel x86, amd64, ARM 32-bits, SPARC 32- et 64-bits



Defora libc

```
System/src/libc/libc-git$ \ls -d src/kernel/*/*/
src/kernel/darwin/amd64/          src/kernel/netbsd/arm/
src/kernel/freebsd/amd64/        src/kernel/netbsd/i386/
src/kernel/freebsd/i386/         src/kernel/netbsd/sparc/
src/kernel/linux/amd64/          src/kernel/netbsd/sparc64/
src/kernel/linux/arm/            src/kernel/openbsd/arm/
src/kernel/linux/i386/           src/kernel/openbsd/i386/
src/kernel/linux/sparc/          src/kernel/solaris/sparc/
src/kernel/netbsd/amd64/         src/kernel/whitix/i386/

$ cat project.conf
package=libc
version=0.4.1
config=h,sh

subdirs=doc,include,src/ssp,src,src/dl,src/math,src/pthread,src/regex,
src/rt,src/socket,tools,tests
dist=Makefile,COPYING,README.md,config.h,config.sh
```



IRC

- Luxe : un laptop ***et*** un desktop
- Plus de temps sur IRC que je ne voudrais admettre
- Solution pour synchroniser les logs ?
- Concept DeforaOS
 ^-- nouvelles lettres
- (pourquoi s'arrêter à la libc ?)



DeforaOS AppProtocol

- Séparer l'interface utilisateur de la logique
 - Le front-end se connecte au backend (comme XChat sur un bouncer)
- Synchroniser l'état interne entre les instances (serveurs, canaux, conversations, messages lus...)
- Adapter l'affichage à l'Interface Homme Machine (IHM)
 - Comme HTML 1.0, et plus jamais depuis
- Le tout avec Gtk+ si possible (C, look & feel)



DeforaOS libApp

- Définition du protocole (RPC)
- Implémente la couche réseau, client et serveur
- Génération de l'interface à implémenter par le programme
- Informatique distribuée sans écrire de code réseau !

(voir aussi CORBA, ou les promesses de GNOME jamais tenues)



Expérimentations

- Où on retrouve VFS
- Exposition de l'API stdio
- Équivalent de Samba/NFS...

```
[call::access]  
ret=INT32  
arg1=STRING,filename  
arg2=UINT32,mode
```

```
[call::chmod]  
ret=INT32  
arg1=STRING,filename  
arg2=UINT32,mode
```

```
[call::chown]  
ret=INT32  
arg1=STRING,filename  
arg2=UINT32,owner  
arg3=UINT32,group
```



Expérimentations

- Maintenant, VPN
- On expose l'API BSD socket
- Ah oui, un VPN

```
[call::connect]  
ret=INT32  
arg1=UINT32, protocol  
arg2=STRING, name
```

```
[call::gethostbyname]  
ret=INT32  
arg1=STRING, name  
arg2=INT32_OUT, address
```

```
[call::recv]  
ret=INT32  
arg1=INT32, fd  
arg2=BUFFER_IN, buf  
arg3=UINT32, size  
arg4=UINT32, flags
```



Expérimentations

- Et OpenGL ?!?
- On a X (ou Wayland)
- Wrapper shell pour AppProtocol
 - Oui, 3D avec `sh(1)`

```
[call::glClearColor]  
arg1=FLOAT, c
```

```
[call::glClearStencil]  
arg1=INT32, s
```

```
[call::glClientActiveTexture]  
arg1=UINT32, texture
```

```
[call::glColor3b]  
arg1=INT8, red  
arg2=INT8, green  
arg3=INT8, blue
```

```
[call::glColor3d]  
arg1=DOUBLE, red  
arg2=DOUBLE, green  
arg3=DOUBLE, blue
```



Premières révélations

- Un seul protocole suffit pour tout !
- Ce protocole peut se transmettre (TODO : Proxy.interface)
- En pratique, Proxy est une nouvelle couche de multiplexage
- Quelle est sa place dans le modèle OSI ?

Layer		Protocol data unit (PDU)
Host layers	7. Application	Data
	6. Presentation	
	5. Session	
	4. Transport	Segment, Datagram
Media layers	3. Network	Packet
	2. Data link	Frame
	1. Physical	Symbol

OSI is fake news

- Il n'y a qu'un protocole
- « Network is RPC »
- Il n'y a qu'une couche, avec différents usages
- Elle parle à ses voisines immédiates
- Elle est répétée autant que nécessaire



Failing TCP/IP

- Le concept de port est incorrect
- Les adresses ne sont pas des noms !
- Mode connecté par défaut
 - UDP est un sous-ensemble du protocole
- Message-Passing au lieu d'octet par octet
 - Le streaming binaire est une interface spécifique
 - Meilleure performance (moins de parsing)
 - Meilleure sécurité (moins de parsing)
- Voir aussi SCTP (mais pas suffisant)



Les avantages

- Éviter les redondances (intégrité, chiffrement...)
- On peut tout nommer (plus de port)
- Adresses de taille variable :
 - Adressage topologique
 - Allocations temporaires (anonymisation)
- On garde la connexion en changeant de réseau ! (mobilité)



Trois possibilités

1. Je suis un génie
ÉCOUTEZ-MOI TOUS
2. Non je dois être full of <poo emoji>
3. Quelqu'un d'autre a eu l'idée avant



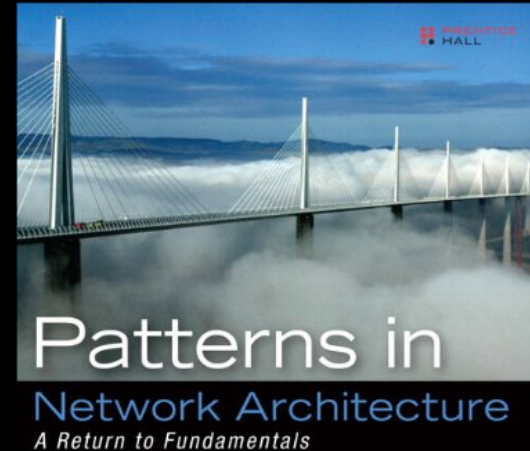
Clean-Slate Internet

- Programme de Stanford University en 2008
- Comment ferait-on Internet aujourd'hui ?
 - Sécurité
 - Mobilité
 - Vie privée
 - Droits d'auteur
 - Le pr0n^Wstreaming vidéo
 - J'en passe...



Ma nouvelle bible

- Patterns in Network Architecture, par John Day de Boston University, chez Prentice Hall
- Lisez-le !
- Vous donne la vérité
- En voici certaines...



JOHN DAY

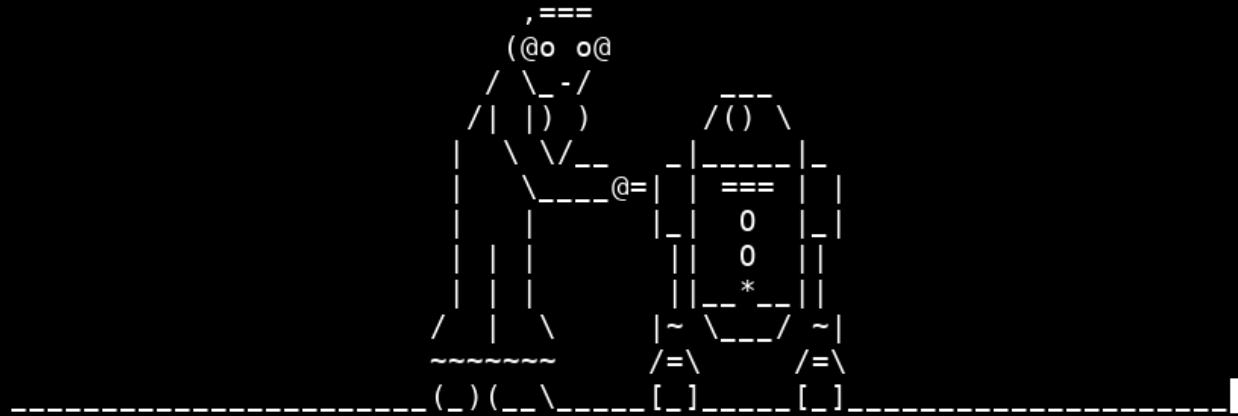


IPv6 n'est pas la solution

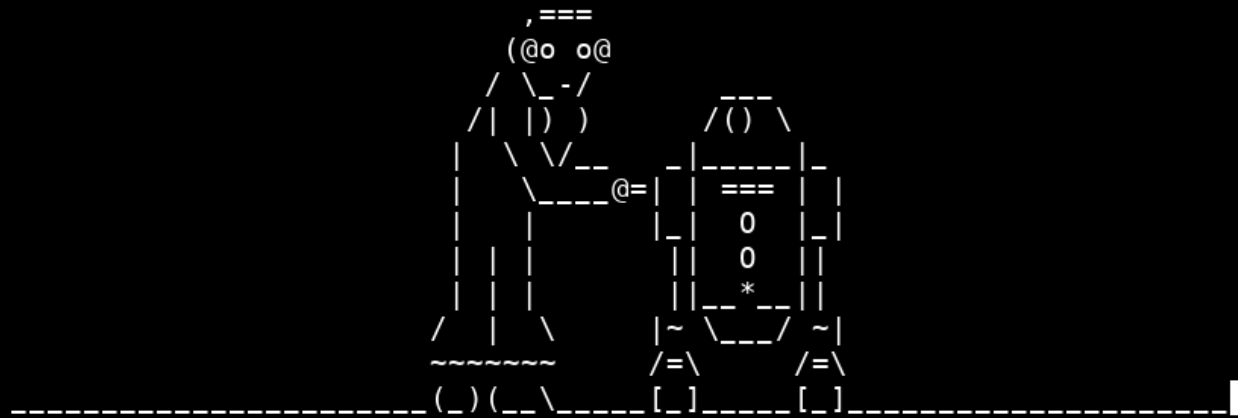
- Migration en Dual Stack, sans garantie de cohérence
 - Tortue KAME plus jolie en IPv6
 - MENSONGE TROMPERIE de ASCII Star Wars !
- Tout a une adresse publique par défaut...
...dont l'adresse MAC ?!?
- Vie privée ? (extension « en cours »)
- Et mes appareils super-sécurisés dès la mise en marche ?
- Quel consommateur × sait configurer le pare-feu de son routeur ?
- Plus d'adresses ? Et le nommage, la mobilité ?



ASCII Star Wars (IPv4)



ASCII Star Wars (IPv6)



Hérésie DNS

- On utilise DNS pour savoir où se connecter
- DNS utilise TCP ou UDP, et donc IP
- Il faut une adresse IP pour se connecter au DNS
- DNS est une couche trop haut (voire plus)
 - Ethernet, PPP...



THE SPANISH INQUISITION

Remaining expected by no-one since September 22, 1970



Network Broker

- Internet est un réseau de réseaux
- Différents réseaux, chacun son espace de nommage plus d'hérésies DNS :
 - .local est réservé à Bonjour
 - Nouveaux TLD tous les jours, aucun pour usage local
- Fournisseur d'accès (pluriel) au réseau global Internet mais aussi
 - Facebook, Google, Netflix...
 - Mes appareils (laptop, desktop, serveur média, téléphone...)
 - Mes cercles familiaux, amis, travail
- VPN à la demande



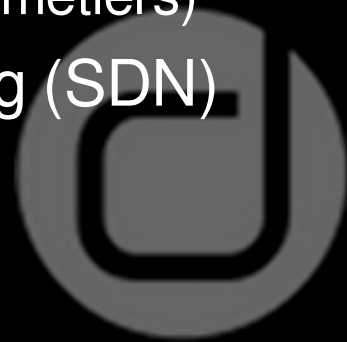
Pourquoi je ne peux pas démontrer aujourd'hui

- DONNEZ-MOI DU TEMPS
- Il faut bien manger :-)
- C'est ambitieux
 - Si c'était facile ça se saurait
 - La réalité est plus compliquée
 - Un OS c'est plus que du réseau
 - En plus il faut convaincre son auditoire



Parcours du combattant

- Choisir ses étapes :
 - Commencer par le kernel, c'est compliqué
 - Hardware is hard
- L'industrie fait ses propres choix
 - Compatibilité : VT100, vraiment? Tout IPv4+TCP+HTTP ?
 - Surveillance globale
 - Réticences aux changements (nouvelle économie, nouveaux métiers)
- Open Network Foundation : Software Defined Networking (SDN)



Retour en 2005

- INSIA, remember?
- Cours d'administration système « faites un mini réseau d'entreprise »
- Tout le monde sous Linux sauf mon groupe « trop facile pour vous »
- Serveur OpenBSD, clients Windows et NetBSD
 - Cauchemars du professeur avec NetBSD/alpha à EPITA
- « vous pouvez pirater les autres équipes »
 - CVE RedHat
 - Remote Code Execution
 - Such shell



2012



2017

**Member of the Board of Directors
The NetBSD Foundation**



NetBSD est <3

- Merveille d'architecture
 - Donc portable
- Device tree, tout est sur un bus
- Système de base cohérent et stable
- Cross-compilation *par défaut*
- RUMP Anykernel
- Android libc (bionic)
- Piles TCP/IP de Windows NT, Blackberry... (QNX)
- macOS et produits Apple
- Routeurs Dell ? (Force 10)
- Téléphones VoIP Panasonic
- License *vraiment libre*



Contributions variées

- Port de drivers OpenBSD
- Environnement de bureau
- Projet AFULTab : tablette avec bureau DeforaOS sur NetBSD
 - EuroBSDCon 2012, EHSM
- Port sur téléphone Nokia N900
- Conférences EuroBSDCon, BSDCan, puis AsiaBSDCon



Hardware is hard

- Compatibilité
 - Intel : mode réel, protégé, 64-bits... APM, ACPI, UEFI...
 - ARM : OABI, EABI, Thumb, floats, 64-bits... Platform initialization...
- Collection d'erratas et autres bugs
 - Dont sécurité : Spectre, Meltdown, Rowhammer...



Oui on peut...



Démo: Builder une image

```
$ ./build.sh -O PREFIX="/usr" \  
-O IMAGE_FILE="DeforaOS.img" \  
-O IMAGE_KERNEL="/netbsd.gz" \  
clean image
```



Démo: Runner une image

```
$ qemu-system-x86_64 \  
DeforaOS-image.img
```



Démo : uKernel

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("Hello, Telecom SudParis!\n");
```

```
    return 0;
```

```
}
```

Parce que pourquoi pas...



Remerciements

- UnderNet, LinuxFR, freenode, OFTC
- John Day
- GnuPG, Tor, Edward Snowden
- Bruce Sterling, Cory Doctorow, c-atre <3
- LANGSEC, Phenoelit, THC, ÜberWall...
- <https://wiki.osdev.org>
- Everyone neighborly



Conclusion

- Et merci à vous pour votre accueil
- Love & hate: khorben@defora.org et @khorben sur Twitter
- <https://git.defora.org> et GitHub
- Convaincus ? Un peu ?
- Have fun!

