



- Caractéristiques et types des systèmes d'exploitation
- Historique de la famille des UNIX
- Généralité sur les dérivés d'UNIX
- Quelques mots sur Microsoft windows.

Récap CM1



1

Allez sur wooclap.com

2

Entrez le code d'événement dans le bandeau supérieur

Code d'événement
DBJAFM




1

Envoyez [@DBJAFM](#) au **06 44 60 96 62**

2

Vous pouvez participer

 Désactiver les réponses par SMS



Caractéristiques et types de systèmes d'exploitation



Différents types de systèmes d'exploitation, selon les services rendus

- **mono/multi-tâche** :

= capacité du système à pouvoir exécuter plusieurs processus **simultanément** ; par exemple effectuer une compilation et consulter le fichier source du programme correspondant.

Préemptif vs collaboratif

Unix, OS/2 d'IBM, Windows

- **mono/multi-utilisateurs** :


= capacité à pouvoir gérer un panel d'utilisateurs utilisant **simultanément** les mêmes ressources matérielles.

Unix/Linux, MacOS, MVS, Gecos, ...



Différents types de systèmes d'exploitation, selon les services rendus

Systeme	Codage	Mono-Utilisateur	Multi-Utilisateur	Mono-Tâche	Multi-Tâche	Type MT
DOS	8/16 bits	X		X		Mono
Windows 3.1	16/32 bits	X			X	Coopératif
Windows 95/98/Me	32 bits	X			X	Coopératif/ Préemptif
Windows NT/2000	32 bits		X		X	Préemptif
Windows XP/10	32/64 bits		X		X	Préemptif
Unix/Linux	32/64 bits		X		X	Préemptif
MacOs	32/64 bits		X		X	Préemptif
VMS	32 bits		X		X	Préemptif



Différents types de systèmes d'exploitation, selon leur capacité à évoluer

Systemes fermés (ou propriétaires) :

- Extensibilité réduite : Quand on veut rajouter des fonctionnalités à un système fermé, il faut remettre en cause sa conception et refaire une archive (système complet).

Unix, MS-Dos ...

- Il n'y a aucun ou peu d'échange possible avec d'autres systèmes de type différent, voir même avec des types identiques.

Unix, BSD et SystemV.

Android et iOS

Systemes ouverts :

- Extensibilité accrue : Il est possible de rajouter des fonctionnalités et des abstractions sans avoir à repenser le système et même sans avoir à l'arrêter sur une machine.

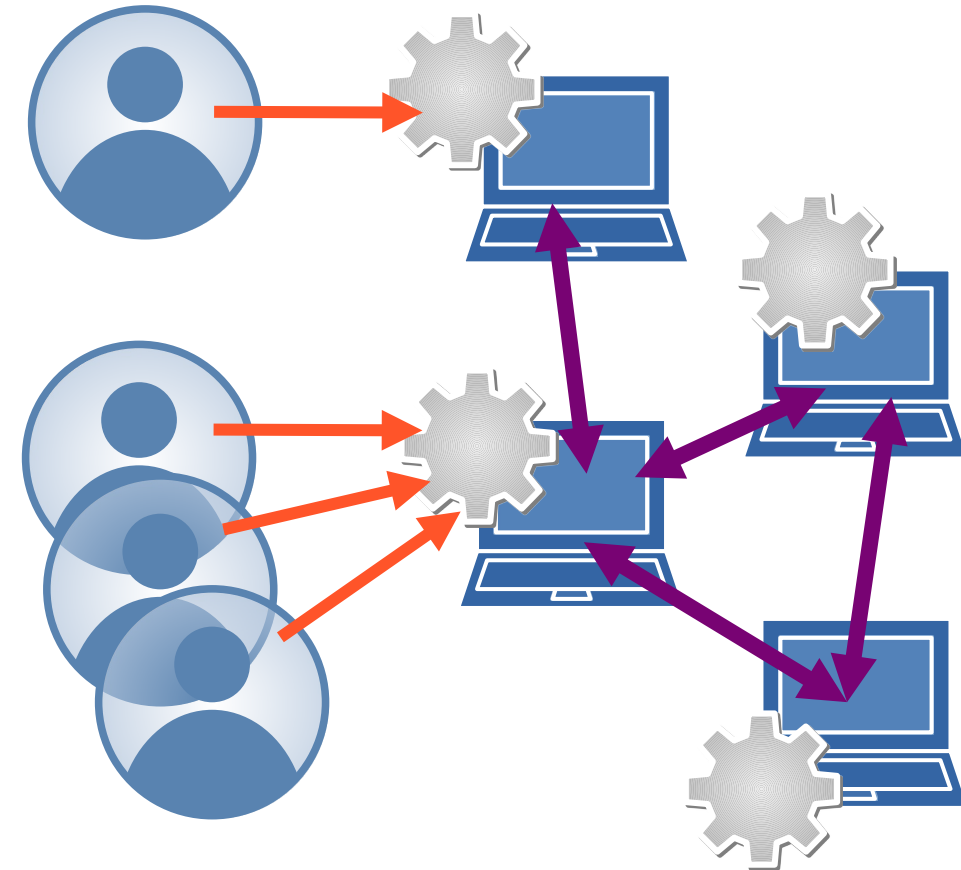
Linux

- Cela implique souvent une conception modulaire basée sur le modèle « client-serveur ».
- Cela implique aussi une communication entre systèmes, nécessitant des modules spécialisés.

Différents types de systèmes d'exploitation, selon leur architecture : systèmes centralisé

- L'ensemble du système est entièrement présent sur la machine considérée.
- Les machines éventuellement **reliées** sont vues comme des entités étrangères disposant elles aussi d'un système centralisé.
- Le système ne gère que les ressources de la machine sur laquelle il est présent.

UNIX, même si les applications réseaux (X11, FTP, Mail...) se sont développées.

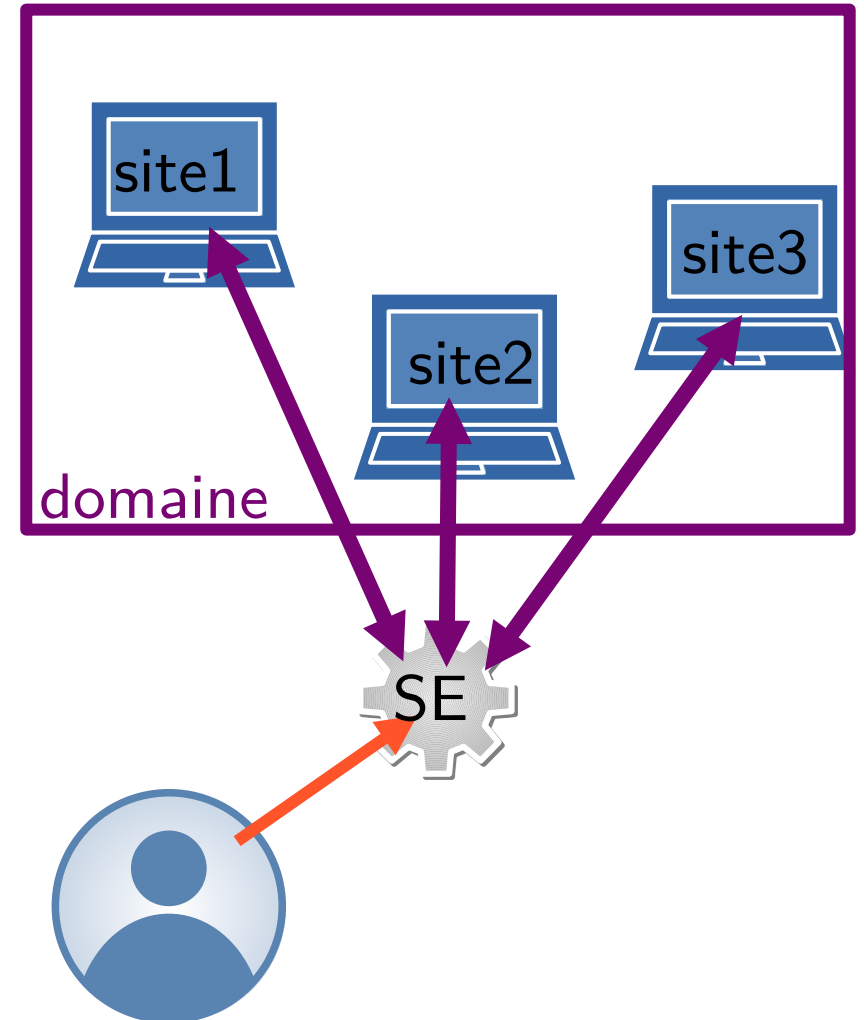


Différents types de systèmes d'exploitation,

selon leur architecture : système répartis = distributed systems

- Le SE contrôle un ensemble (**domaine**) de machines (**site**).
- L'utilisateur.ice n'a pas à se soucier de la localisation exacte des ressources. Quand iel lance un programme, iel n'a pas à connaître le nom de la machine qui l'exécutera. Le SE apparaît à ses yeux comme **une machine virtuelle monoprocesseur** même lorsque cela n'est pas le cas.
- Ces systèmes exploitent au mieux les capacités de parallélisme d'un domaine, et ils offrent des solutions aux problèmes de la **résistance aux pannes**.

Mach, Amoeba, Andrew, Athena, Locus, ...



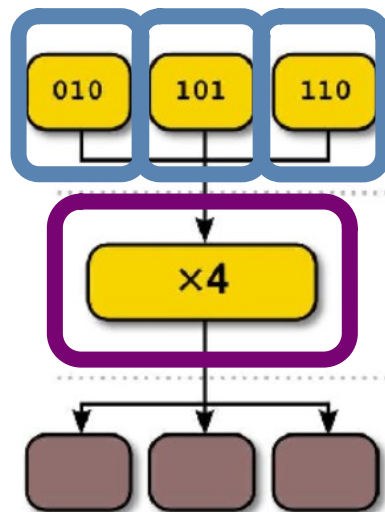
Différents types de systèmes d'exploitation, selon l'architecture matérielle qui les supporte

Architecture monoprocesseur : (pseudo)-parallélisme possible grâce à la **commutation** rapide entre les différents processus (pour donner l'illusion d'un parallélisme)

Architecture multiprocesseur (parallélisme natif) : grande variété d'architectures [Flynn, 1972] :

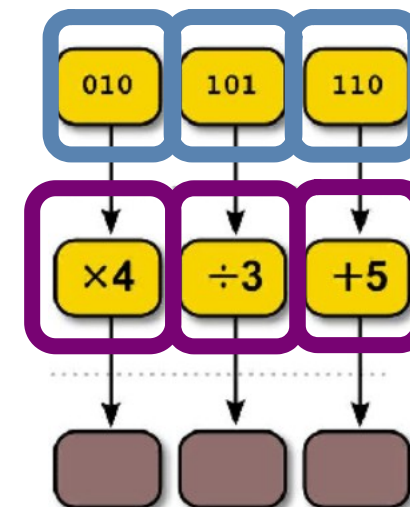
SIMD (**Single Instruction Multiple Data**)

Tous les processeurs exécutent la même instruction, mais sur des données différentes.



MIMD (**Multiple Instructions Multiple Data**) :

Chaque processeur est complètement indépendant des autres et exécute des instructions différentes sur des données différentes.






Architecture : appellations alternatives

- **Architecture fortement couplée** : Ce sont principalement des architectures à mémoire commune.
- **Architecture faiblement couplée** : Ce sont des architectures où chaque processeur possède sa propre mémoire locale ; c'est le cas d'un réseau d'ordi.
- **Architecture mixte** : Ce sont des architectures à différents niveaux de mémoire (commune et privée).


Remarque : Il n'y a pas de système universel pour cette multitude d'architectures. Les constructeurs de supercalculateurs ont toujours développé leurs propres systèmes. Aujourd'hui, compte tenu de la complexité croissante des systèmes d'exploitation et du coût inhérent, la tendance est à l'harmonisation notamment via le développement de systèmes polyvalents.



Différents types de systèmes d'exploitation, selon leur relation au temps : systèmes temps-réel

- Généralement dans les **systèmes embarqués** (satellites, sondes, avions, trains...).
- Exécution des programmes soumise à des **contraintes temporelles** (absolues ou relatives) : les résultats de l'exécution d'un programme n'est **plus valide au delà d'un certain temps** connu et déterminé à l'avance.
- Système temps réel **strict** : aucun dépassement de contrainte n'est toléré
- Système temps réel **souple** : s'accommode des dépassements de contraintes dans certaines limites

Linux-RT, RTX, Windows CE, Embedded Linux, Symbian OS, Palm OS et VxWorks



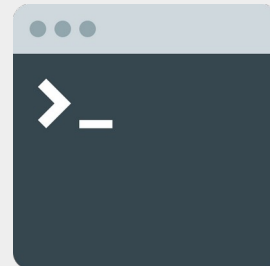
Différents types de systèmes d'exploitation :

5 générations successives

Quand	Nom	Description
années 50'	traitements par lots (batch)	calculs les uns après les autres.
années 60'	multiprogrammation	exécution simultanée de plusieurs programmes visant l'utilisation efficace de la puissance de calcul du processeur.
1970	temps partagé	techniques avancées de multiprogrammation pour donner l'impression à chaque utilisateur.ice du système qu'il est le seul.e à utiliser l'ordi
milieu 70'	temp réel	avec garantie que toute opération se termine dans un délai donné
années 90'	distribués	utilisation des ressources de plusieurs ordinateurs à la fois



Généralités sur UNIX et ses dérivés



Historique 1969 - 1979 :

les premiers pas universitaires

- **Été 1969** : **Ken Thompson**, chercheur aux **BELL** Laboratories, écrit la **version expérimentale d'UNIX** : système de fichiers exploité dans un environnement **mono-utilisateur, multi-tâche**, le tout étant écrit **en assembleur** sur un ordi de récupération **PDP 7** de 1964.
 - 1ère justification officielle : traitement de texte pour secrétariat (écriture de brevets)
 - Puis : étude des principes de programmation, de réseaux et de langages.



Le PDP 7 Oslo

<https://en.wikipedia.org/wiki/PDP-7>

Historique 1969 - 1979 :

les premiers pas universitaires

- **Eté 1969** : **Ken Thompson**, chercheur aux **BELL** Laboratories, écrit la **version expérimentale d'UNIX** : système de fichiers exploité dans un environnement **mono-utilisateur, multi-tâche**, le tout étant écrit **en assembleur** sur un ordi de récupération **PDP 7** de 1964.
 - 1ère justification officielle : traitement de texte pour secrétariat (écriture de brevets)
 - Puis : étude des principes de programmation, de réseaux et de langages.
- **Eté 1973** : réécriture du noyau et des utilitaires d'UNIX **en C** (langage compilé implémenté en **1972** par **Dennis Ritchie** à partir du langage interprété B, écrit par Ken Thompson).
- **En 1974** distribution d'UNIX aux **Universités** (**Berkeley et Columbia** notamment). Il se compose alors :
 - d'un système de fichiers modulaire et simple,
 - d'une interface unifiée vers les périphériques par l'intermédiaire du système de fichiers,
 - du multi-tâche
 - et d'un interprète de commandes (**shell**) flexible et interchangeable.

Historique 1979 - 1984 :

les premiers pas commerciaux

1979, avec la version 7, UNIX se développe commercialement :

- Par des sociétés privées comme **Microport** (1985), **Xenix-Microsoft** (1980) ... qui achetèrent les sources et le droit de diffuser des binaires.
- Des **UNIX-like** apparaissent ; le noyau est entièrement réécrit.
- L'**université de Berkeley** fait un portage sur les ordi **VAX** (ordi avec mémoire virtuelle) : BSD UNIX 32V.
- **AT&T** vend la version 7 sur les ordinateurs de la gamme **PDP 11**.

⇒ Se développent plusieurs produits différents, tous marketés « UNIX », et pas compatibles entre eux... Chaque industriel ajoute des fonctionnalités et pas grand monde partage les infos.

Historique 1984 - 1993 ... :

« UNIX war » et la standardisation

1984 un groupe de vendeurs (dont Siemens, Phillips et Ericson) met en place **X/Open**, chargé de définir un standard UNIX pour permettre la **portabilité**.

1987 **AT&T** et **SunMicrosystems** pactisent pour mettre en place un système UNIX unifié (system V x BSD)

1988 : Contre-attaque d'autres industriels pro-BSD (**OSF/1** via **Open Software Foundation**).
Contre-contre-attaque d'**AT&T** (**System V** via **UNIX international**).

D'autres standards sont développés en parallèle (**POSIX**), d'autres OS aussi (Windows)

Mai 1993 : des membres des 2 camps annoncent l'initiative **COSE** (Common Open Software Environment) = accord pour le développement d'applications dans un environnement commun.

1994 : fusion avec X/Open → the **Open Group**,
aujourd'hui, le **Open Group** est le seul à décerner le label UNIX (**The Single UNIX Specification**)

Au final : guerres techniques et culturels qui ont fini par passer au second plan, en faveur d'autres OS (Windows, macOS (dérivé de BSD), Linux).



La normalisation d'UNIX

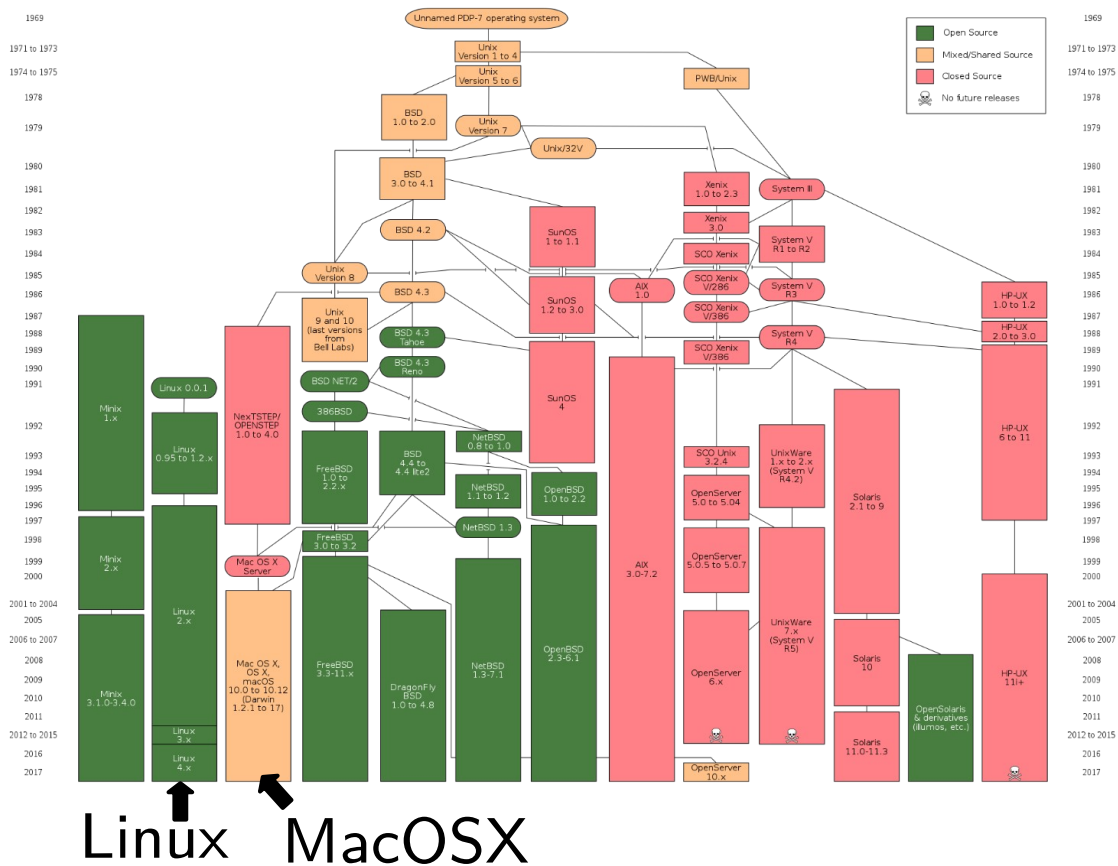
Historique :

- **1985** : Le **SVID** (System V Interface Definition) d'AT&T, qui définit l'interface d'application du Système V.2 et non pas son implémentation.

Toujours utiles :

- Pour les programmeur.euses et les system designers : les standards nous donnent un cadre qui rendent nos codes portables
- Pour les utilisateur.ices : les standards nous donnent un cadre qui rendent les outils facile à utiliser, « ça va marcher ».
 - **POSIX** (Portable Operating System Interface X), définit l'interface avec le système. Interface labellisée par l'ANSI (American National Standard Institute) et l'ISO (International Standard Organisation). **[Linux, macOS]**
 - les **Spec1170**, devenue Single UNIX Specification (**SUS**) de l'Open Group. **[macOS]**

Arbre généalogique de la famille des UNIX



Unix est l'ancêtre de bien des systèmes d'exploitation, notamment **Linux**, **BSD**, **Solaris**, **AIX**, **MacOSX**.



Caractéristiques générales du noyau UNIX

- **Multi-tâche / multi-utilisateur**
 - Plusieurs utilisateur.ices peuvent travailler en même temps ; chacun.e peut effectuer une ou plusieurs tâches en même temps.
 - Une tâche ou un processus = programme s'exécutant dans un environnement spécifique.
 - Les tâches sont protégées ; certaines peuvent communiquer, c-à-d échanger ou partager des données, se synchroniser dans leur exécution ou le partage de ressources. Certaines tâches peuvent être « temps réel ».
- **Système de fichiers arborescent**
 - Arborescence unique de fichiers, même avec plusieurs périphériques (disques) de stockage.
- **Entrée/Sorties compatible fichiers, périphériques et processus**
 - Les périphériques sont manipulés comme des fichiers ordinaires.
 - Les canaux de communication entre les processus (pipe) s'utilisent avec les mêmes appels systèmes que ceux destinés à la manipulation des fichiers.

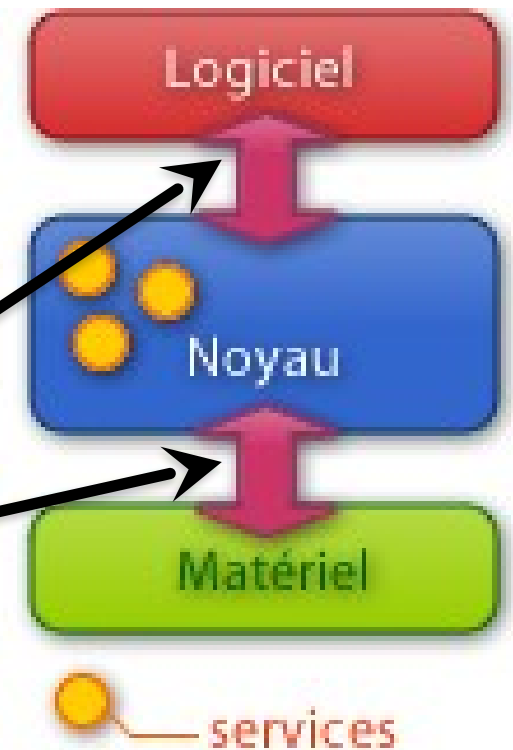


Les qualités du système UNIX

- Code source facile à lire et à modifier ; disponible commercialement.
- Interface utilisateur simple et très puissante (mais « non-conviviale » ?).
- Le système est construit sur un petit nombre de primitives de base ; de nombreuses combinaisons possibles entre programmes. (modularité)
- Les fichiers ne sont pas structurés au niveau des données, ce qui favorise une utilisation simple.
- Toutes les interfaces avec les périphériques sont unifiées (système de fichier).
- Le programmeur n'a jamais à se soucier de l'architecture de la machine sur laquelle iel travaille.
- C'est un système disponible sur de nombreuses machines, allant du super-calculateur au micro-ordinateur (PC), en passant par les smartphones.
- Les utilitaires et programmes proposés en standard sont très nombreux.

Caractéristiques du noyau UNIX

- UNIX comprend un **noyau (kernel)** et des **utilitaires (services)**.
- Irremplaçable par l'utilisateur, le noyau gère les processus, les ressources (mémoires, périphériques ...) et les fichiers.
- Tout autre traitement doit être pris en charge par des **utilitaires** ; c'est le cas de l'*interprète de commande* (sh, csh, ksh, tcsh ...).
 - Interfaces du noyau
 - L'**interface** entre le **noyau** UNIX et les **programmes utilisateurs** est assurée par un ensemble d'**appels systèmes**.
 - L'**interface** entre le **noyau** UNIX et les **périphériques** est assurée par les **gestionnaires de périphériques (devices driver)**



https://fr.wikipedia.org/wiki/Noyau_de_syst%C3%A8me_d%27exploitation.

UNIX : interface



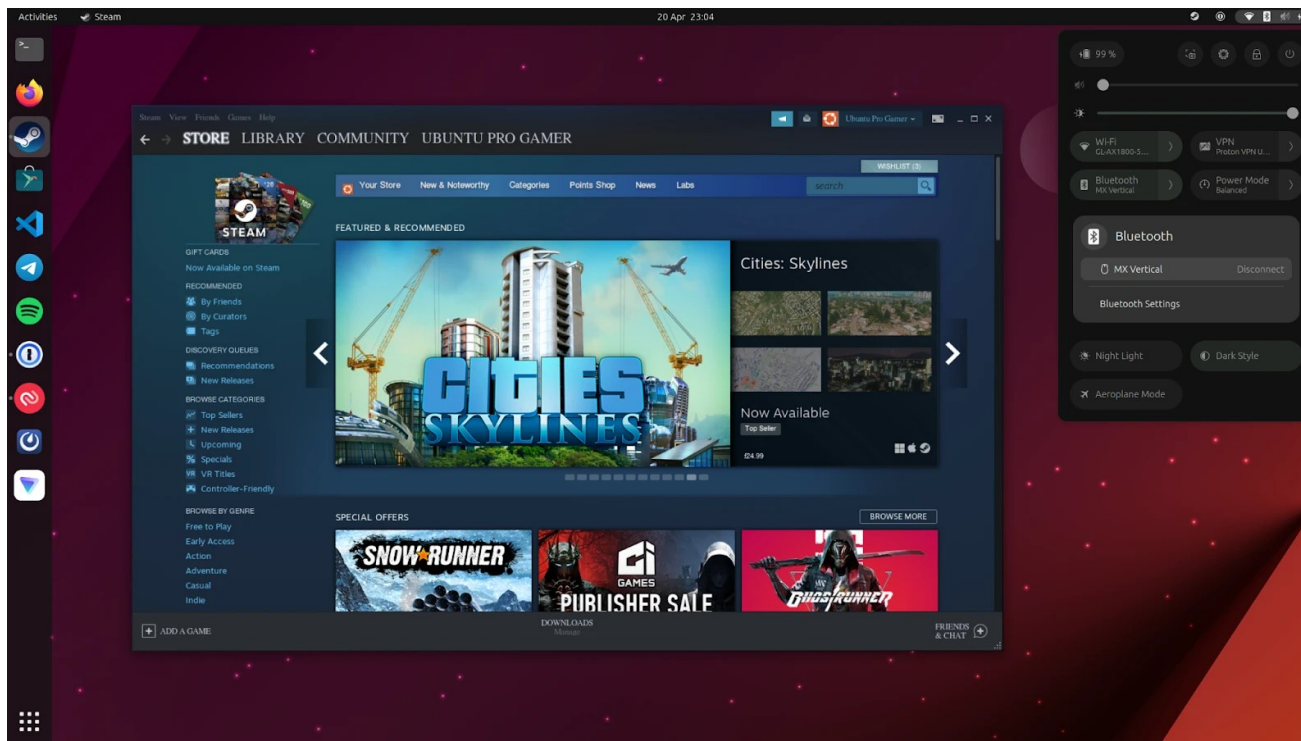
Bureau Unix vers le début des années 1990.

http://www-ens.iro.umontreal.ca/~gottif/bdeb/inf32/c1_fichiers/image049.jpg

Deux environnements très différents sous Unix

- l'environnement graphique (des boutons et des fenêtres)
- l'environnement console / terminal, en ligne de commande

Et il existe plusieurs environnements console : les **Shells**.



```
jfa@jfa-VirtualBox:~$ cd  
jfa@jfa-VirtualBox:~$ pwd  
/home/jfa  
jfa@jfa-VirtualBox:~$
```



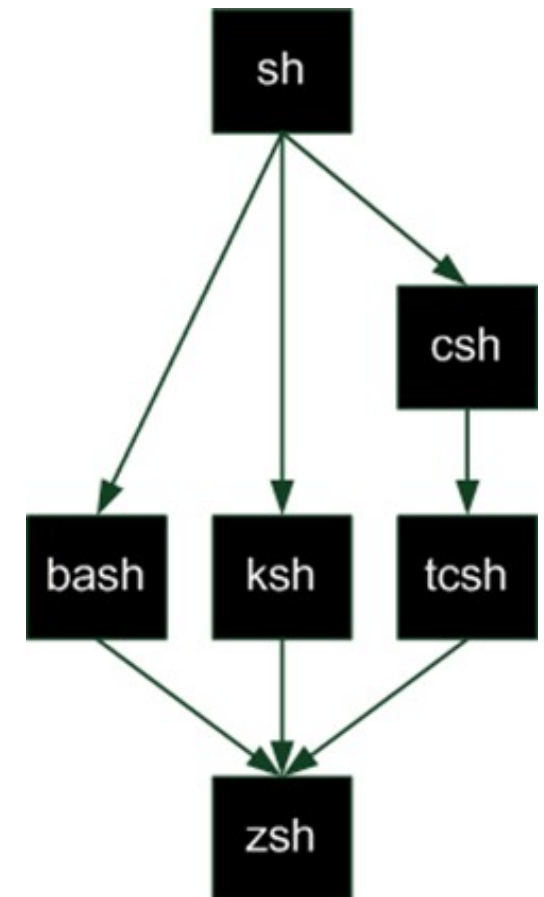

Qu'est-ce qu'un Shell Unix ?

Un **Shell Unix** est un **interpréteur de commandes** destiné aux systèmes d'exploitation Unix (et de type Unix) qui permet d'accéder aux fonctionnalités internes du système d'exploitation. Il se présente sous la forme d'une **interface en ligne de commande** accessible depuis la **console** ou un **terminal**. L'utilisateur lance des commandes sous forme d'une **entrée texte exécutée** ensuite **par le shell**.

Dans les différents systèmes d'exploitation Microsoft Windows, le programme analogue est par exemple **powershell**.

Les principaux Shells Unix

- **Sh : Bourne Shell.** L'ancêtre de tous les Shells. Sa syntaxe des commandes est proche de celle des premiers UNIX
- **Bash : Bourne Again Shell.** Une amélioration du Bourne Shell augmenté de la plupart des fonctionnalités avancées du C shell, un script Bourne shell sera correctement interprété avec un Bash, disponible par défaut sous GNU/Linux et Mac OS X.
- **ksh : Korn Shell.** Un Shell puissant assez présent sur les Unix propriétaires, mais aussi disponible en version libre, compatible avec Bash.
- **csh : C Shell.** Un Shell utilisant une syntaxe proche du langage C.
- **tcsh : Tenex C Shell.** C'est une extension et amélioration du C shell d'origine.
- **zsh : Z Shell.** Shell assez récent reprenant les meilleures idées de Bash, ksh et tcsh



Terminal / TTY / Console / Shell

- Terminal = extrémité
- Téléscripneur = télétype = TTY
- Console = « pupitre de commandes »
- émulateur de terminal = console virtuelle

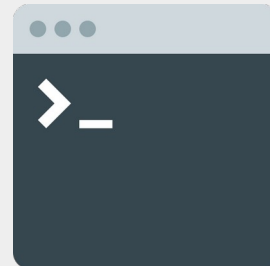


```
(jammy-base r5828)nanis@C302L-G24P07: ~  
┌───(jammy-base r5828)nanis@C302L-G24P07: ~  
(base) (jammy-base r5828)nanis@C302L-G24P07:~$ ls  
baseconfig Desktop miniconda3 ownCloud  
baseconfig-gitlab Documents Modèles Public  
bluetooth-pods.sh Images Musique signal-desktop-keyring.gpg  
(base) (jammy-base r5828)nanis@C302L-G24P07:~$ pwd  
/home/nanis  
(base) (jammy-base r5828)nanis@C302L-G24P07:~$
```



Généralités sur UNIX et ses dérivés

Dérivé n°1 : Linux





Historique 1991 - ... :

LINUX, le renouveau d'UNIX

1991 : **Linus B. Torvalds** (étudiant de 21 ans à l'univ. D'Helsinki, Finlande) étudie MINIX (un OS UNIX-like écrit par A. Tannenbaum)

Août 1991 : 1ère version de LINUX 0.01. C'est une réécriture de MINIX, avec des ajouts de nouvelles fonctionnalités et la diffusion des sources sur « Internet »

→ une **version instable**

Mars 1994 : **1ère version stable**, 176 250 lignes de code.

Janvier 2004 : v.2.6.0 qui respecte la norme POSIX (code source portable) et le code source est gratuit.

Novembre 2013 : v.3.12.0 écrite en C et en assembleur, sous licence GNU GPL 2.

2022 : 27,8 millions lignes de code (projet libre qui en contient le plus).



Linux

Post de Linus Torvalds pour partager son OS :

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID:
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
```

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Source : l'archive Usenet (l'ancêtre d'internet)

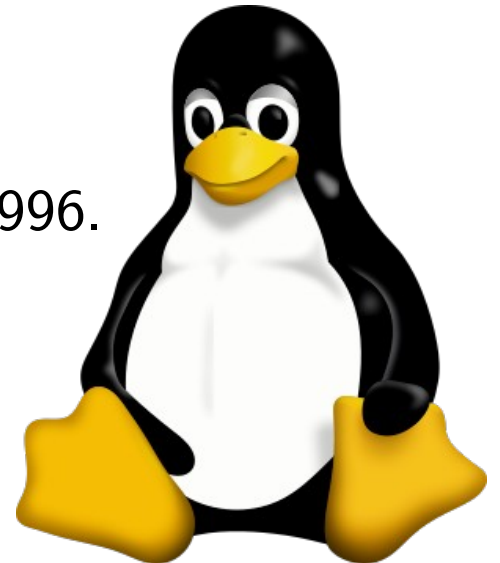
LINUX

Accronyme récursif :

LINUX = LINUX Is Not UniX = LINUX Is Not UniX = LINUX Is Not UniX = LINUX Is Not UniX = LINUX Is Not UniX = LINUX Is Not UniX = LINUX Is Not UniX = LINUX Is Not UniX = LINUX Is Not UniX = LINUX Is Not UniX = LINUX Is Not UniX = LINUX Is Not UniX = LINUX Is Not UniX = LINUX Is Not UniX = LINUX Is Not UniX = ...

Mascotte :

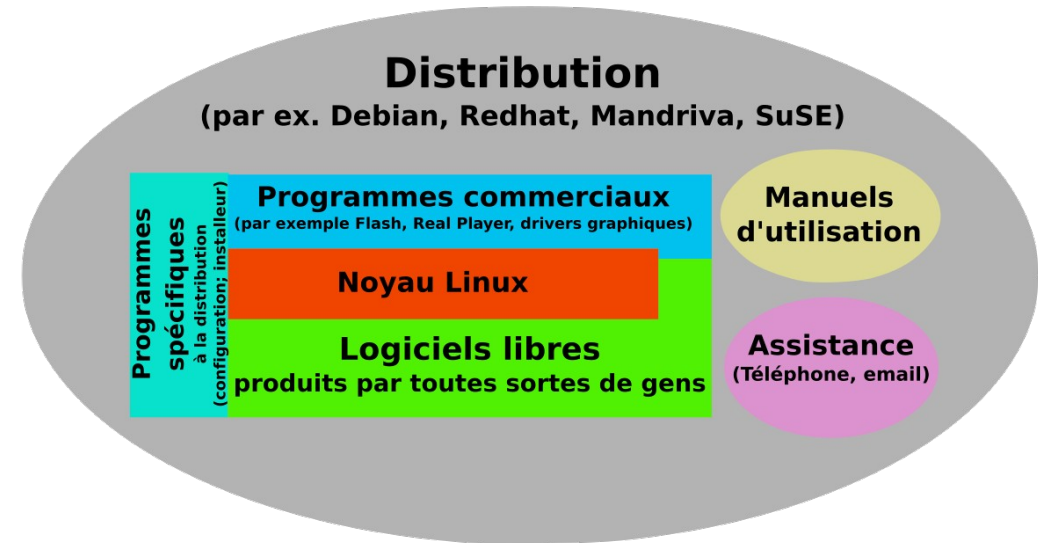
le pingouin manchot TUX (« Torvalds Unix »), crée par Larry Ewing en 1996.



Distributions LINUX (ou « distro »)

Définition :

Ensemble cohérent de logiciels assemblé autour d'un **noyau Linux** (des pilotes, les bibliothèques, les utilitaires, ... et une surcouche custom de logiciels).



« Linux » vs « GNU » vs « GNU/Linux »

À l'origine :

- À sa création (dans les années 90) le **noyaux Linux** n'était pas directement utilisable : un noyau n'est pas un système d'exploitation complet.
- En parallèle (depuis 1983), le **projet GNU (Richard Stallman)** travaillait à la mise en place d'un système d'exploitation compatible UNIX, et basé sur la **philosophie GNU** (logiciel libre). Il avait plein de composants logiciels mais il lui manquait un noyau.

Donc, selon le contexte :

- « **GNU** » se rapporte aux logiciels du projet GNU, à un système d'exploitation dont le noyaux est très probablement Linux (d'où le fait que le SE soit parfois appelé GNU/Linux), mais aussi à une licence de code (GNU GPL)...
- « **Linux** » se rapporte à un noyau, ou à un système d'exploitation dont le **noyaux est Linux**, et qui peut (ou pas) avoir sa couche logicielles issue (tout, ou partie) du projet GNU.



Distributions LINUX

- **Slackware** : la plus vieille distribution encore en activité (1993)
- **Fedora** : entreprise RedHat
- **S.u.S.E** : grande robustesse
- **Mandrake** : *Caennaise* basée sur **RedHat**
- **Caldera** : inclut des produits commerciaux
- **Gentoo** : système entièrement compilé à partir des sources, gestionnaire de paquetage Portage
- **Trinux** : fonctionne uniquement en mémoire, outils d'audit des réseaux
- **TurboLinux** : version en cluster, payante, pour gros serveurs
- **Knoppix** : très populaire, sans disque dur
- **Arch** : se veut simple (principe KISS – Keep it Simple Stupid)
- **Ubuntu** : entreprise Canonical, très populaire, basée sur **Debian**
- **Debian** : non commerciale et de grande qualité (genre vraiment).

et bien d'autres... (dont **Android** sur nos smartphones).

Arbre des distributions GNU/Linux sur http://fr.wikipedia.org/wiki/Distribution_Linux .

Distributions LINUX

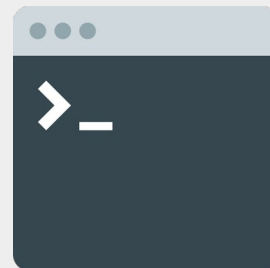


<https://linux.developpez.com/actu/137043/Quelles-sont-vos-distributions-Linux-preferees-Et-pour-elles-utilisations-Merci-de-partager-votre-experience-avec-les-distributions-Linux/>



Généralités sur UNIX et ses dérivés

Dérivé n°2 : MacOS



Mac OS (renommé OS X puis MacOS)

- Le système d'exploitation des ordinateurs Macintosh d'Apple → intimement **lié au matériel**.
- **Partiellement propriétaire** : certaines parties de FreeBSD (une saveur d'Unix) ont été réutilisées par Apple dans ce système d'exploitation.



1984 - 1988



Macintosh

1988 - 1995



Mac OS

1995 - 2002



Mac OS X

2001 - 2003



Mac OS X

2003 - 2012

OS X

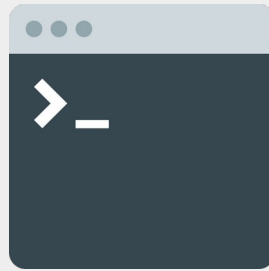
2012 - 2016

macOS

2016 - 2017

macOS

2017 - now



Windows : la vache à lait de Microsoft

Windows

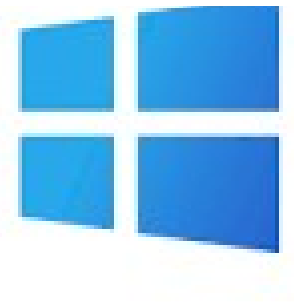
- Propriétaire, développé par Microsoft



```
Écran de démarrage MS-DOS 6.22.  
Démarrage de MS-DOS...  
Vérification de la mémoire étendue par HIMEM...  
Vérification terminée.  
C:\>C:\DOS\SMARTDRV.EXE /X  
Fonction MODE PREPARE pour la page de codes terminée  
Fonction MODE SELECT pour la page de codes terminée  
C:\>_
```

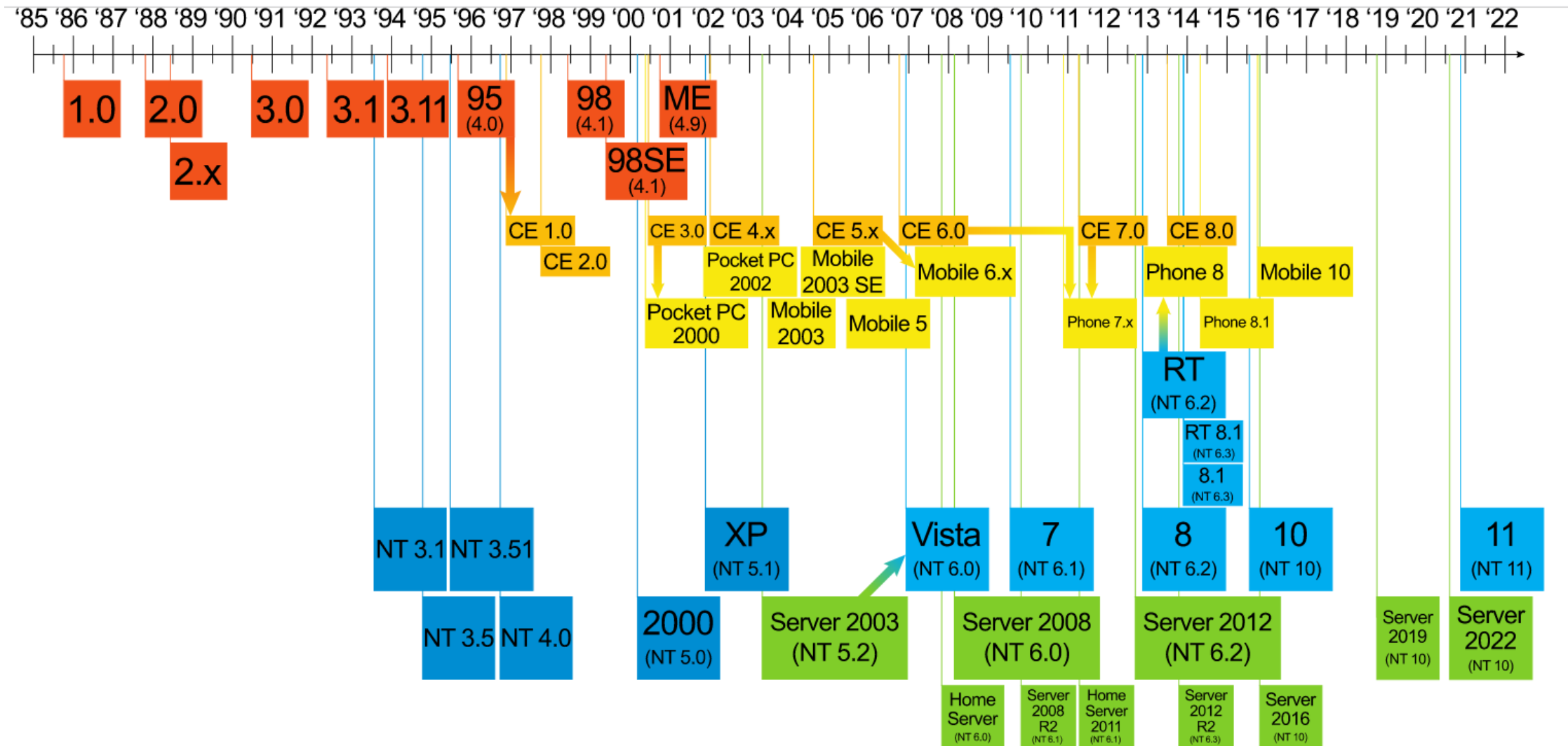
Historique :

- 1981 : MS-DOS « Microsoft Disk Operating System » :
Système monotâche, mono-utilisateur, et pas de mode protégé.
Interface en ligne de commande (langage CMD).
- 1985 : Windows v1 = interface graphique pour MS-DOS
- 1993 : Nouveau kernel Windows NT (« New technology »).
Système d'exploitation multitâche préemptif, multi-utilisateur, multiprocesseur.
Langage : powershell.



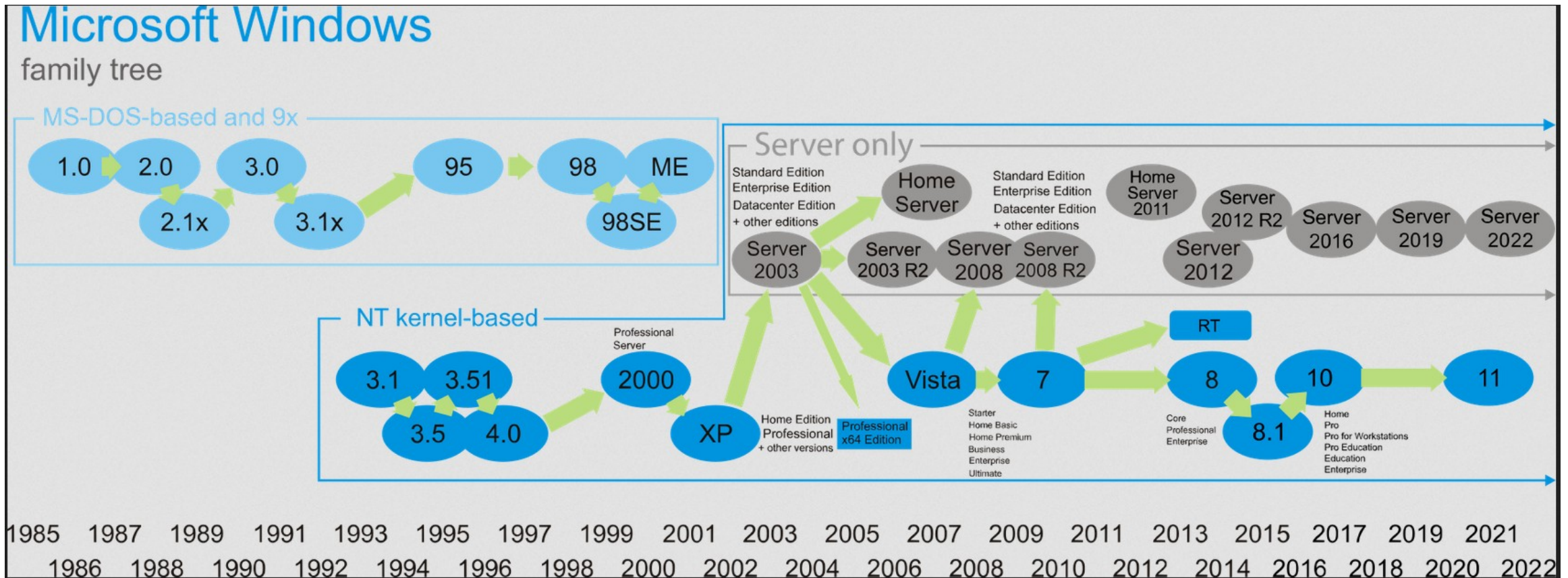
Sous-système Windows pour Linux (WSL) : plus besoin de dual boot.

Windows : l'arbre familial, versions



https://upload.wikimedia.org/wikipedia/commons/1/17/Suite_des_versions_de_Windows.svg

Windows : l'arbre familial, noyau



https://upload.wikimedia.org/wikipedia/commons/thumb/0/0e/Windows_Family_Tree.svg/1889px-Windows_Family_Tree.svg.png

Le quasi-monopole de Microsoft Windows

- « vente-liée » : Windows est (quasi) systématiquement installé quand on achète un ordinateur
- « racketiciel » (AFUL : Association francophone des utilisateurs de logiciels libres)







Recap

- Il existe de nombreux OS, certains étant adaptés à des cas très spécifiques.
- Plusieurs OS (MacOS, Linux, Android) ont un ancêtre commun : UNIX, et partagent donc une organisation proche.
- Les protagonistes des histoires de tous ces OS ne sont pas si vieux (par ex, **Ken Thompson** travaille chez **Google** depuis 2006 et est impliqué dans le développement du langage **Go**).



Teaser

TD 1 :

Première utilisation de Linux, via l'interface en lignes de commandes. :))))))

CM3 et suivants :

Reprise des notions vues en TD : utilisateur, fichiers, droits, plein de lignes de commandes



« ça marche pô »

- Vérifiez qu'il n'y a pas une faute de frappe (/!\ les commandes sont sensibles à la case).
- Vérifiez que vous avez les bons arguments, le bon nombre
- Regardez les indications du terminal (le terminal est votre ami)
- Préparez-vous à poser votre question à son.a voisin.e / le.a prof / internet :
 - Qu'est ce que voulais faire ?
 - Quelles sont mes entrées ?
 - Quelles sont les sorties attendues ?