



MOOC  
**Maîtriser le shell Bash**

Document compagnon

---

Séquence de bienvenue

**Présentation du cours**

---

Version 1.4



# Licence Creative Commons CC BY-NC-SA 4.0 International

## Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 4.0 International (CC BY-NC-SA 4.0)

**Avertissement** Ce résumé n'indique que certaines des dispositions clé de la licence. Ce n'est pas une licence, il n'a pas de valeur juridique. Vous devez lire attentivement tous les termes et conditions de la licence avant d'utiliser le matériel licencié.



Creative Commons n'est pas un cabinet d'avocat et n'est pas un service de conseil juridique. Distribuer, afficher et faire un lien vers le résumé ou la licence ne constitue pas une relation client-avocat ou tout autre type de relation entre vous et Creative Commons.

**Clause : C'est un résumé (et non pas un substitut) de la licence.**

<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode> Vous êtes autorisé à :

- **Partager** - copier, distribuer et communiquer le matériel par tous moyens et sous tous formats,
- **Adapter** - remixer, transformer et créer à partir du matériel,
- pour une utilisation non commerciale.

L'Offrant ne peut retirer les autorisations concédées par la licence tant que vous appliquez les termes de cette licence.

**Selon les conditions suivantes :**

 **Attribution (BY)** - Vous devez créditer l'oeuvre, intégrer un lien vers la licence et indiquer si des modifications ont été effectuées à l'oeuvre. Vous devez indiquer ces informations par tous les moyens raisonnables, sans toutefois suggérer que l'Offrant vous soutient ou soutient la façon dont vous avez utilisé son Oeuvre.

 **Pas d'Utilisation Commerciale (NC)** - Vous n'êtes pas autorisé à faire un usage commercial de cette Oeuvre, tout ou partie du matériel la composant.

 **Partage dans les mêmes Conditions (SA)** - Dans le cas où vous effectuez un remix, que vous transformez, ou créez à partir du matériel composant l'Oeuvre originale, vous devez diffuser l'Oeuvre modifiée dans les même conditions, c'est à dire avec la même licence avec laquelle l'Oeuvre originale a été diffusée.

**Pas de restrictions complémentaires** - Vous n'êtes pas autorisé à appliquer des conditions légales ou des **mesures techniques** qui restreindraient légalement autrui à utiliser l'Oeuvre dans les conditions décrites par la licence.

**Notes** : Vous n'êtes pas dans l'obligation de respecter la licence pour les éléments ou matériel appartenant au domaine public ou dans le cas où l'utilisation que vous souhaitez faire est couverte par une **exception**.

Aucune garantie n'est donnée. Il se peut que la licence ne vous donne pas toutes les permissions nécessaires pour votre utilisation. Par exemple, certains droits comme les **droits moraux, le droit des données personnelles et le droit à l'image** sont susceptibles de limiter votre utilisation.

Les informations détaillées sont disponibles aux URL suivantes :

- <http://creativecommons.org/licenses/by-sa/4.0/deed.fr>
- [http://fr.wikipedia.org/wiki/Creative\\_Commons](http://fr.wikipedia.org/wiki/Creative_Commons)



## Les auteurs



Pascal ANELLI est enseignant-chercheur à l'Université de la Réunion. Il enseigne l'informatique depuis plus de 20 ans. Il utilise les outils d'Unix et en particulier le Bash dans le cadre de ses travaux de recherche. Il a formé des dizaines d'étudiants à ce langage.



Régis GIRARD est maître de conférences à l'Université de La Réunion. Il a enseigné Unix et le shell pendant plusieurs années en Licence d'Informatique.



Xavier NICOLAY est ingénieur de recherche à l'Université de La Réunion. Il a été ingénieur systèmes & réseaux pendant 20 ans puis Directeur Informatique. Il intervient également comme enseignant dans de nombreuses formations universitaires.



Denis PAYET est maître de conférences à l'Université de La Réunion, il enseigne dans le domaine du génie logiciel et de la programmation en intervenant sur les différentes formations de l'université : DUT, Licence, Master et dans le cycle Ingénieur.



Tahiry RAZAFINDRALAMBO est maître de conférences à l'Université de La Réunion il enseigne l'informatique et notamment l'automatisation des tâches d'administration sous Linux et Unix. Il utilise le Bash dans la majorité de ses travaux et l'enseigne aux étudiants.



Pierre-Ugo TOURNOUX est maître de conférences à l'Université de La Réunion, il enseigne l'informatique et notamment l'administration système et serveur. Le Bash est la pierre angulaire de la plupart de ses enseignements.



# Table des activités

<b>0.1</b>	<b>Un cours orienté pratique</b>	<b>3</b>
1	Motivations à l'utilisation du shell . . . . .	3
2	Pourquoi connaître le shell ? . . . . .	3
3	Objectif du cours . . . . .	4
4	Contenu du cours . . . . .	4
5	Organisation . . . . .	4
6	A qui s'adresse ce cours ? . . . . .	6
7	Pré-requis . . . . .	6
<b>0.2</b>	<b>Comment le cours est-il structuré ?</b>	<b>7</b>
<b>0.3</b>	<b>Qu'est ce qu'un shell ?</b>	<b>9</b>
1	Introduction . . . . .	9
2	Le système d'exploitation de type Unix . . . . .	10
3	Le shell . . . . .	12
4	Conclusion . . . . .	14
<b>0.4</b>	<b>Guide d'utilisation de la console Weblinux</b>	<b>15</b>
1	Présentation de la Weblinux . . . . .	15
2	L'accès à la Weblinux . . . . .	16
3	Tutoriel pour l'usage de la Weblinux . . . . .	17
4	Les limites de la Weblinux . . . . .	18
5	En cas de problème . . . . .	19
6	Conclusion . . . . .	19



## Activité 0.1

# Un cours orienté pratique

## 1 Motivations à l'utilisation du shell

Le shell *Bash* est désormais omniprésent car il a été intégré à Windows 10 depuis une mise à jour récente. Ainsi, dorénavant, les 3 grands systèmes d'exploitation pour ordinateur (Windows, MacOS et Linux) partagent cet outil puissant et incontournable pour tout informaticien, que ce soit à l'échelle du poste de travail, d'un serveur et même d'un smartphone ou d'une tablette. L'essor des terminaux Android permet en effet également d'intégrer la dimension mobile au spectre d'utilisation du shell. On peut aussi noter que l'engouement des nano-ordinateurs comme les Raspberry PI et plus généralement de l'instrumentation connectée (ou Internet des objets) fonctionne presque exclusivement sous une distribution Linux. L'utilisation du shell est donc largement transversale et sa connaissance est un passage obligé pour tous les informaticiens.

## 2 Pourquoi connaître le shell ?

La science informatique revêt plusieurs aspects, un des plus fondamentaux porte sur le langage pour écrire des programmes. Dans la multitude des langages de programmation, le langage de programmation shell occupe une place à part. Et pour bien marquer cette différence, on parle de script et non de programme. En effet, en plus d'être un langage de programmation, le shell est aussi un interpréteur de commandes accessible depuis la console (aussi appelé terminal). Le shell offre alors une interface homme-machine pour décrire, sous la forme de lignes de commande, les actions à effectuer au sein du système d'exploitation. Cette interface textuelle nécessite de la pratique pour être maîtrisée et elle est souvent ressentie comme complexe à apprendre. Par la flexibilité et la richesse qu'elle propose, elle est indispensable pour celui qui souhaite administrer et configurer un ordinateur personnel ou un serveur sous Linux.

La programmation de script shell répond à la philosophie Unix classique consistant à diviser les tâches complexes en sous-tâches plus simples, puis à chaîner des composants et des utilitaires. Ces utilitaires sous forme de commandes sont comme une boîte à outils disponible dans les systèmes de type Unix. Beaucoup considèrent que c'est la meilleure approche, ou tout au moins plus agréable pour la résolution d'un problème que d'avoir recours à écrire un programme complet avec un langage puissant et généraliste. Cela impose cependant une modification du mode de réflexion, qui doit être adapté aux utilitaires disponibles.

### 3 Objectif du cours

En utilisant le shell *Bash*, vous verrez comment la console est complémentaire de l'interface graphique et par la richesse du shell *Bash*, vous découvrirez comment améliorer votre expérience utilisateur sur votre ordinateur. Ce cours a pour objectif de vous rendre capable d'utiliser une console pour les opérations courantes sur votre ordinateur. L'expression de vos actions s'effectue au moyen d'un langage de commande. Nous vous proposons de vous familiariser avec le shell *Bash*. Ce shell est la version la plus courante. Un langage quel qu'il soit s'apprend en le pratiquant - c'est pourquoi, nous adoptons une approche orientée vers la pratique.

A l'issue de ce cours vous serez capable de lire, d'écrire et de comprendre des commandes dans le contexte du shell *Bash*. Mais au delà de l'interaction classique avec le shell *Bash*, ce cours vous propose d'acquérir les compétences nécessaires pour exploiter toute sa puissance dans des scripts. Vous aurez acquis les connaissances suffisantes pour pouvoir interagir avec votre ordinateur sans passer par une interface graphique. De plus, vous maîtriserez les fondamentaux d'un langage de script pour en apprendre d'autres facilement par vous même.

### 4 Contenu du cours

Une connaissance pratique de la programmation de scripts shell est indispensable à quiconque souhaitant administrer un système informatique. Les scripts shell peuvent servir à mettre en place des procédures pour le système mais également constituer des programmes utilitaires. Le script shell va vous permettre d'écrire des prototypes d'application, de programmer des tâches courantes ou d'automatiser des enchaînements de programmes.

L'apprentissage des scripts n'est pas compliqué puisque les scripts shell peuvent se construire progressivement par petit morceaux par le principe même des langages interprétés. La syntaxe d'un script est simple même si elle peut paraître austère. Elle est identique à celle utilisée en ligne de commande. Cette syntaxe ne demande que de connaître quelques règles et des opérateurs spécifiques. La plupart des scripts courts fonctionnent la première fois, et la mise au point des scripts les plus importants reste simple.

Le cours couvre tous les aspects du shell et de son langage de programmation. Vous commencerez par étudier pendant les 3 premières séquences :

- les éléments d'un système d'exploitation ;
- la syntaxe spécifique du shell ;
- les utilitaires du système d'exploitation.

La dernière séquence sera consacrée à la présentation des structures de programmation pour la réalisation de scripts shell.

### 5 Organisation

L'approche pédagogique de ce cours se veut orientée sur l'utilisation courante du shell et sur la présentation des principales commandes d'un système de type Unix. Il peut être tentant de présenter l'ensemble des commandes et des constructions disponibles par un shell. Vu le nombre, cela peut vite devenir un catalogue. Nous avons préféré nous concentrer sur l'essentiel, à savoir sur la présentation

#### Langages interprétés

Dans ces langages, le code informatique appelé code source, que vous écrivez, est traité par un logiciel : l'interpréteur. Le rôle de ce dernier va consister à lire et à exécuter les lignes du programme une par une. A chaque lancement du programme à interpréter, l'interpréteur effectue cette traduction du code source en un code exécutable par l'ordinateur.

des concepts sous-tendant les commandes présentées. Nous avons recours à de nombreux exemples pour illustrer les éléments du langage .

Notre démarche de présentation est la suivante :

- Après un exposé des concepts illustrés par des commandes à l'aide de la vidéo, un support de cours reprend plus en détail les éléments de langage exposés par la vidéo. Ce support de cours que nous appellerons *document compagnon* constitue un complément indispensable à la vidéo.
- Une console identique à celle utilisée par l'enseignant est proposée pour refaire les exemples.
- Des exercices sous la forme de challenges sont à réaliser afin de pratiquer les notions abordées et de permettre une acquisition immédiate. Ces travaux pratiques que vous aurez à faire le sont dans une console Linux de votre navigateur. Cet environnement Linux simplifié et commun à tous les apprenants, vous permet de faire vos premières manipulations sans risque. A la fin du cours, vous pourrez vous lancer sans crainte dans un système complet.
- Des quiz sont à remplir pour valider les compétences que vous acquerez au fur et à mesure de votre progression.
- Un forum de discussion sera ouvert avec un fil de discussion relatif à chaque thème abordé. N'hésitez pas à aller sur le forum pour poser des questions, faire des remarques, obtenir des informations ou des éclaircissements supplémentaires sur le cours. La communauté du MOOC est là pour vous aider.

Ce cours est échelonné sur 6 semaines dont la sixième semaine de révision et d'entraînement. Chaque semaine présente une nouvelle séquence. Après la séquence de bienvenue, vous aurez 4 séquences thématiques composées chacune d'une introduction, de 5 activités et d'une conclusion. Le contenu de chaque séquence est développé dans un document compagnon disponible sous la forme d'un fascicule mis à disposition à l'ouverture de la séquence.

Par ailleurs, et dans le cadre de l'animation de votre MOOC, un enseignant vous proposera une conférence en ligne sur une démonstration. Vous pouvez suivre directement en ligne cette conférence, ou avoir accès à l'enregistrement qui sera accessible à partir d'un lien qui vous sera communiqué. Cette conférence sera annoncée dans l'info-cours.

Enfin, les apprenants, ayant obtenu un score global supérieur à 60% sur les évaluations, obtiendront une attestation de suivi avec succès attribuée par FUN.

## 5.1 Comment passer les évaluations ?

Le quiz de chaque activité porte exclusivement sur la vidéo associée à l'activité. Elle sert à l'apprenant à valider sa compréhension du thème présenté à partir de la vidéo. La vidéo donne une vision générale. Il est souvent nécessaire d'avoir consulté le document compagnon pour avoir une compréhension plus complète du thème présenté. Et de manière générale, les challenges pratiques s'appuient sur des compétences acquises avec la lecture du document compagnon.

C'est après avoir effectué le quiz de la vidéo, consulté le document compagnon et réalisé les challenges pratiques de la séquence que vous pouvez passer le devoir de fin de séquence. Ainsi votre progression sera évaluée à l'aide des 3 types de travaux :

- Les quiz portant sur la vidéo d'une activité, valant 1% chacun. Il y en a 20, donc les quiz comptent pour 20% de la note finale.
- Les challenges pratiques d'une activité, valant 3% chacun. Il y a en tout 20 challenges répartis en 4 séquences. Ainsi l'ensemble des challenges d'une séquence valent pour 15% de la note. Et tous Les challenges comptent pour 60% de la note finale.
- Les devoirs de fin de séquence, valant 5% chacun. Il y en a 4, donc les devoirs comptent pour 20% de la note finale.

Une attestation de suivi avec succès sera émise par FUN et attribuée aux apprenants obtenant une note supérieure à 60% à l'ensemble des évaluations du cours effectuées avant le 26 mars 2018.

## 6 A qui s'adresse ce cours ?

Vous êtes étudiants en informatique, autodidactes en informatique, ou plus généralement une personne s'initiant ou se perfectionnant à un système d'exploitation de type Unix (dont le *Bash* permet le pilotage). Ce cours s'adresse à vous, informaticiens en devenir ou spécialistes confirmés. Vous avez en commun de vouloir découvrir et maîtriser le langage de l'interaction avec un système de type Unix.

## 7 Pré-requis

Le pré-requis pour suivre ce cours est de posséder des notions simples de programmation. Les structures et les principes de la programmation doivent être connus afin de comprendre comment ils sont décrits par le shell. Ce cours s'adresse donc à des apprenants ayant des connaissances minimales de programmation.

Cependant, nous sommes convaincus qu'une forte motivation et un travail sérieux peuvent remplacer ces pré-requis.

La section "Testez vos connaissances" vous permet de vérifier si votre niveau de connaissance est suffisant pour commencer ce cours dans les meilleures conditions.

## Activité 0.2

# Comment le cours est-il structuré ?

Le MOOC Maîtriser le shell *Bash* vous donne les bases pour interagir avec un ordinateur à l'aide d'un langage de commande. Au moyen des vidéos pédagogiques et des travaux pratiques, vous allez acquérir les compétences nécessaires pour exprimer vos actions par un langage adapté pour le dialogue avec l'ordinateur et au-delà, pour programmer des scripts pour automatiser les tâches courantes. Vous serez ainsi capable d'exploiter toute la puissance d'expression et la flexibilité du shell.

Ce cours est structuré en 4 séquences thématiques comportant chacune une introduction, 5 activités et une conclusion. Préalablement à ces séquences, le cours commence par une séquence de bienvenue. La structuration de ce cours est la suivante :

1. Séquence de bienvenue présentée par Pascal Anelli & Denis Payet. Cette semaine traite de :
  - Présentation du cours
  - Qu'est-ce qu'un shell ?
  - Les challenges pratiques
  - Pour naviguer sur FUN
  - Faisons connaissance
  - Mieux vous connaître
  - Testez vos connaissances
  - Réussir votre MOOC
2. Séquence 1 "Découvrez votre système d'exploitation" présentée par Pierre-Ugo Tournoux
  - Qu'est ce que la ligne de commande ?
  - Trouver de l'aide
  - Gérer les répertoires et les fichiers
  - Les utilisateurs et leurs droits
  - Traiter un fichier de texte
3. Séquence 2 "Interagissez avec le Bash" présentée par Pascal Anelli & Régis Girard
  - Aide à l'interaction
  - Abréviations pour le nom des fichiers
  - Constructions syntaxiques
  - Contrôler l'exécution des commandes
  - Entrées et sorties des processus
4. Séquence 3 "Maîtrisez votre système d'exploitation par le Bash" présentée par Tahiry Razafindralambo
  - Contrôler son environnement
  - Filtres simples
  - Filtres puissants
  - Effectuer des calculs numériques
  - Archiver et compresser des données

5. Séquence 4 "Automatisez vos travaux" présentée par Denis Payet & Régis Girard
- Éléments d'un script shell
  - Expressions et conditions
  - Structures conditionnelles
  - Structures itératives
  - Structures de routines

A l'issue de ce cours, le shell *Bash* va faire partie de votre quotidien.

## Activité 0.3

# Qu'est ce qu'un shell ?

## 1 Introduction

### 1.1 C'est quoi ?

Habituellement, quand vous êtes devant votre ordinateur, vous communiquez avec lui en utilisant des modalités d'interaction graphique. Vous avez en effet à votre disposition une souris avec laquelle vous faites bouger un curseur à l'écran pour sélectionner des items et agir sur eux à l'aide de menus. Vous pouvez agir de cette façon parce que le système d'exploitation (SE ou en anglais OS pour *Operating System*) de votre ordinateur dispose d'une interface graphique qui affiche les éléments de votre ordinateur, et vos différents programmes, dans des fenêtres contenant des boutons et des menus.

Tous les systèmes d'exploitation modernes disposent de ce type d'interface. C'est effectivement plus facile pour l'utilisateur humain de visualiser les choses ainsi, et d'indiquer par des gestes au système ce qu'il doit faire. C'est plus facile pour l'humain, mais ce n'est pas aussi facile pour le système lui-même, car le système ne fonctionne pas en se représentant les choses visuellement comme un humain le fait. Aussi, l'interface graphique est une surcouche logicielle qui est chargée de traduire les informations du système sous forme d'images. Cette interface se charge en retour d'interpréter les actions de l'utilisateur effectuées à l'aide de la souris et du clavier en actions compréhensibles par le système interne.

L'interface graphique est ainsi une sorte d'interprète entre l'utilisateur et le système d'exploitation. Bien que l'interface graphique facilite les choses, ça ne les rend pas pour autant forcément efficaces. En effet, cet interprète ne laisse à l'utilisateur que les actions que le concepteur de l'interface a bien voulu lui laisser. Le champ des possibles est forcément limité. De plus, cet interprète consomme des ressources de l'ordinateur qui ne sont pas disponibles ou accessibles comme c'est le cas pour les accès depuis un ordinateur distant. Alors, quand on maîtrise l'interprétation des informations que l'on recherche, ou l'action que l'on souhaite déclencher, il est souvent plus efficace de se passer de cet intermédiaire et d'opérer plus directement avec le système. La question est donc : ce moyen de dialogue plus direct existe-t-il ? Et la réponse est : oui, c'est **le shell**.

#### Programme application et logiciel

Les termes : programme, application et logiciel sont des synonymes. Mais il existe cependant de petites nuances qui font qu'on est parfois amené à choisir l'un d'eux plutôt qu'un autre : le mot **programme** met l'accent sur les traitements informatiques réalisés, le terme **application** insiste sur le fait que l'utilisateur final est le principal bénéficiaire, alors que **logiciel** est souvent employé pour marquer une distinction par rapport à la couche matérielle.

## 1.2 L'origine du shell

Historiquement, les premiers systèmes d'exploitation ne disposaient pas d'interface graphique. D'ailleurs, à cette époque, la souris n'existait même pas. On interagissait donc avec le système essentiellement par échange de texte. L'écran servait à recevoir les informations fournies par le système sous forme de phrase, d'affichage de valeur ou de tableau textuel. Et l'utilisateur utilisait le clavier pour saisir les ordres qu'il voulait soumettre au système. C'est cette interface d'échange textuel qu'on appelle le shell.

Parler au passé du shell ? En fait, non, le shell existe encore. Il règne en maître depuis toujours sur la quasi-totalité des serveurs web que vous visitez, des ordinateurs des systèmes informatiques des entreprises et des administrations, ou encore sur les serveurs qui fournissent les services que vous utilisez quotidiennement sur Internet (Cloud, Twitter, etc.).

## 1.3 Pourquoi s'intéresser au shell ?

Le shell n'a jamais vraiment disparu, il a simplement été de moins en moins visible par le grand public. Essentiellement parce que pour le grand public l'accent a été mis sur la facilité d'usage plus que sur l'efficacité et la puissance de cet usage. Mais les choses ont déjà commencé à changer, car de plus en plus d'utilisateurs veulent aller plus loin. Ils sont de plus en plus exigeants et de plus en plus compétents, car comme vous, ce sont des passionnés qui cherchent à contrôler toujours mieux leur ordinateur. A tel point que l'on constate aujourd'hui que le shell est en train de faire son grand retour : notamment avec les objets connectés et les Raspberry Pi, mais aussi sur des systèmes d'exploitation propriétaires tels que Windows 10 dont la dernière mise à jour propose un accès plus immédiat au shell.

Pour ces passionnés, et donc pour vous, dans de nombreuses situations les interfaces graphiques vont très vite devenir un véritable frein. Vous aurez donc besoin d'un moyen d'échange plus intime avec le système d'exploitation. Vous aurez besoin du shell. C'est pourquoi nous vous proposons de faire la connaissance de cet outil, et avec toute l'équipe de ce MOOC nous éprouvons une grande fierté à vous guider dans cette aventure.

Dans la suite de cette activité introductive, nous allons présenter quelques notions liées au fonctionnement d'un ordinateur. Nous commencerons par définir le rôle du système d'exploitation et en particulier celui de type Unix. Nous reviendrons ensuite de façon plus technique sur les fonctions du shell. Nous expliquerons pourquoi la version *Bash* du shell est celle retenue dans ce MOOC. Puis nous terminerons cette activité en justifiant l'utilité d'un langage pour une interface shell.

# 2 Le système d'exploitation de type Unix

## 2.1 Qu'est-ce qu'un système d'exploitation

Un ordinateur est en ensemble d'éléments électroniques et mécaniques animé par un calculateur central appelé micro-processeur. Sur cet ensemble se greffent des composants plus ou moins "externes", dits périphériques, qui sont des dispositifs dédiés aux entrées et sorties de données vers et depuis le micro-processeur. Certains de ces périphériques sont en charge de l'interaction avec l'utilisateur comme par exemple le haut parleur, le clavier ou encore l'écran. Pour chacun de ces types de périphériques, il y a de nombreux modèles avec pour chacun d'eux des spécificités techniques propres.

L'utilisation de ces équipements depuis un programme informatique demanderait au concepteur de ce programme (le programmeur) d'écrire du code qui va interagir directement avec les équipements. Mais c'est là un travail complexe, car il faudrait prendre en compte toutes les diversités techniques

pour un même type d'équipement. Car la portabilité d'un tel code (le fait d'assurer son bon fonctionnement pour tous les cas de figure) obligerait à prendre en compte, dans le développement, tous les périphériques qui pourraient être rencontrés. Ce serait là une tâche beaucoup trop longue et compliquée, et le résultat serait un programme dont le code serait volumineux avec le risque d'un nombre important de bugs. Aussi, la solution proposée à cette problématique est ce qu'on appelle le système d'exploitation.

Le système d'exploitation est un logiciel système qui joue un rôle d'intermédiaire entre les programmes et les éléments matériels de l'ordinateur. C'est le premier programme exécuté au démarrage de la machine, et c'est le seul qui reste en permanence en exécution. On le nomme également noyau (*kernel*).

Le rôle du noyau est de masquer l'hétérogénéité des composants en offrant des fonctionnalités de manipulation uniformes, quelque soit le dispositif réellement connecté à l'ordinateur. Il décharge ainsi les programmes de la gestion spécifique de chaque matériel. Le noyau gère notamment les entrées sorties (I/O : *Inputs/Outputs*), que ce soit vers ou depuis les périphériques de stockage, le clavier ou encore l'écran. C'est aussi un gestionnaire de ressources qui s'occupe de distribuer la mémoire, le temps de calcul CPU, les accès aux périphériques, etc. de manière équitable et efficace entre les différents programmes qui s'exécutent en même temps. La vision simplifiée que l'on peut avoir d'un ordinateur peut se représenter par la figure 0.3.1. L'utilisateur accède à des applications qui utilisent des ressources matérielles qui sont gérées et contrôlées par le noyau. Rien ne peut se faire sans le concours du noyau. C'est la raison pour laquelle l'ordinateur se bloque lorsqu'il y a une erreur dans le noyau.

#### Logiciel système

Un logiciel système est un groupe de programmes et de bibliothèques logicielles qui forment un support permettant de créer et de faire fonctionner d'autres logiciels qualifiés de "logiciel applicatifs".

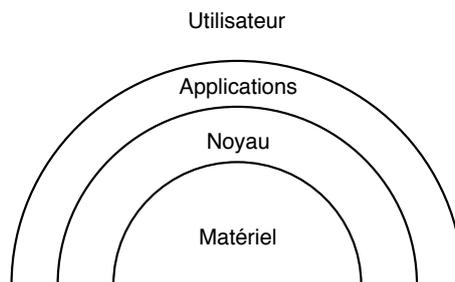


FIGURE 0.3.1: Organisation d'un ordinateur.

## 2.2 Un système d'exploitation nommé Unix

Le système d'exploitation Unix est un noyau très complet. En plus de prendre en charge la gestion de la mémoire, celle des entrées/sorties et de la répartition des temps de calcul entre les programmes, il incorpore également des programmes utilitaires pour effectuer des tâches courantes comme la lecture des fichiers ou leur édition. Ces programmes peuvent être sollicités à la demande, ou bien encore s'exécuter en permanence en arrière plan (on parle alors de service) de sorte à toujours être disponible instantanément. Cet ensemble de constituants est accompagné de fichiers de configuration spécifiques qui définissent les nombreux réglages de ces différents utilitaires et services. C'est cet ensemble composé du noyau et des programmes utilitaires qu'on appelle le système Unix.

#### Arrière plan

On dit qu'un programme s'exécute en arrière plan lorsqu'il n'a pas besoin d'être associé à une session. Autrement dit, il n'a pas besoin d'un écran et d'un clavier pour fonctionner. Nous reviendrons sur cette notion dans la suite de ce cours.

Le système d'exploitation Unix peut se représenter sous la forme de couches comme le montre la figure 0.3.2. Au sommet de ces couches se trouve l'utilisateur qui va utiliser l'ordinateur pour atteindre des

buts. Pour cela, il va interagir avec des programmes. Ces programmes vont interpréter les actions qu'il réalise sur les périphériques d'entrées comme le clavier et retourner des états sur les périphériques de sorties comme l'écran. Ces programmes peuvent être :

- des commandes : l'ensemble des programmes utilitaires fourni avec le système,
- le shell : un programme spécifique pour une interaction en mode texte pour accéder aux commandes et aux fonctions du noyau.
- des applications spécifiques aux besoins de l'utilisateur (interface graphique, traitement de texte, navigateur web, etc.).

Nous allons maintenant détailler le rôle du shell dans l'interaction avec l'utilisateur.

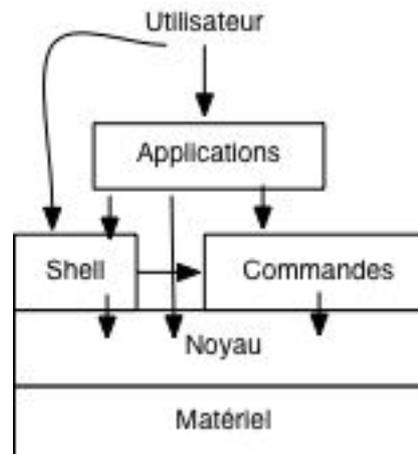


FIGURE 0.3.2: Représentation simplifiée du système d'exploitation Unix.

### 3 Le shell

Le shell ou coquille en français est la couche logicielle pour accéder aux fonctions du noyau et pour accéder aux commandes du système d'exploitation. La philosophie des systèmes d'exploitation de type Unix est de privilégier une diversité de petits programmes utilitaires spécialisés et optimisés, plutôt que des programmes volumineux capables de "tout faire". Mais "tout faire" est devenu possible à l'aide du shell. En effet, le shell donne à l'utilisateur les moyens de manipuler tous les programmes, et même de les assembler pour d'accomplir des tâches complexes. Il en résulte un procédé bien plus flexible et efficace, et c'est ce qui explique que le shell occupe encore aujourd'hui une place importante dans le paysage informatique moderne.

#### 3.1 Une coquille à plusieurs fonctions

Le shell se présente sous la forme d'une interface en ligne de commande accessible depuis une console, qu'on appelle aussi un terminal. L'utilisateur exécute des commandes sous la forme d'une ligne de texte dont le shell se charge d'interpréter la syntaxe. Le shell est ainsi une application qui sert d'interface entre l'utilisateur et le système d'exploitation Unix.

En fait, le shell exécute les commandes que l'utilisateur saisit dans la console, c'est ce qu'on appelle le mode interactif. Mais il peut aussi traiter une liste de commandes pré-enregistrées dans un fichier ; on parle alors de mode script, ou plus généralement de script shell. Un script shell peut être utilisé pour créer des applications en combinant les fonctions du noyau, les utilitaires et les applications. Et comme si cela ne suffisait pas, le shell comporte des structures de programmation qui apporte une flexibilité supplémentaires aux scripts. Il devient alors un langage de programmation assez complet. Les scripts

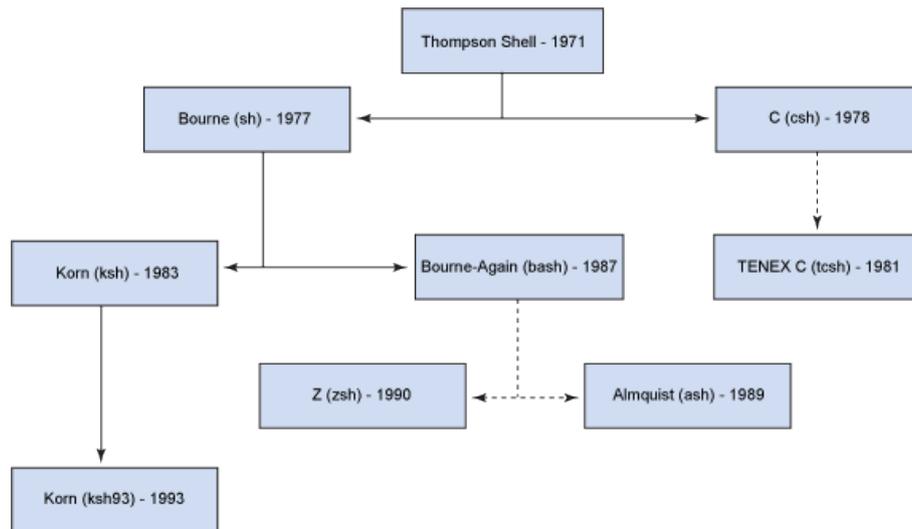


FIGURE 0.3.3: Généalogie des shells.

shell sont particulièrement adaptés aux tâches administratives du système et à d'autres tâches répétitives ne nécessitant pas d'avoir recours à des développements avec des langages de programmation de haut niveau.

Le shell est donc une interface par laquelle l'utilisateur interagit avec l'ordinateur. D'un point de vue fonctionnel, il joue ainsi le même rôle que l'interface graphique d'un système Windows, Linux, ou Mac OS. Mais, comme nous allons le voir dans ce MOOC, malgré le fait qu'il soit moins raffiné visuellement, sa puissance ne connaît aucune limite !

Pour terminer, notons, qu'un terminal et une console peuvent être confondus fonctionnellement, ils proposent tous les deux d'effectuer une session en shell. Et, de ce point de vue, on ne peut rien faire de plus ou de moins qu'on soit dans une console ou un terminal. En pratique, la console est l'équipement physique d'un terminal sans mode graphique. C'est une sorte de Minitel (pour ceux qui ont connu). A ce titre, il n'a pas de barre de défilement sur le coté au contraire d'un terminal. Un terminal prend de nos jours la forme d'une application en mode fenêtre dans un environnement graphique pour émuler une console.

## 3.2 Le shell *Bash*

Même s'il joue le rôle aussi spécifique que celui que nous venons de voir, il n'en reste pas moins que du point de vue du système Unix, le shell n'est autre qu'un programme utilitaire parmi d'autres. Par conséquent n'importe qui peut créer son propre shell du moment que celui-ci est conforme au rôle attendu d'un shell. Cela a mené à l'apparition une douzaine de shells concurrents dans les années 90.

Les shells se différencient par leur facilité d'utilisation, leur programmabilité ainsi que par leur licence d'utilisation. Ceux qui sont conformes à la norme POSIX présentent le plus d'avantages, notamment en termes de portabilité. Car ils partagent un même ensemble de syntaxe, ce qui assure notamment que des commandes ou des scripts développées dans l'un des shells respectant cette norme pourra fonctionner à l'identique dans tous les autres shells conformes à POSIX.



### Norme POSIX

POSIX est une famille de normes techniques définie depuis 1988 par l'Institute of Electrical and Electronics Engineers (IEEE), et formellement désignée par IEEE 1003. Ces normes ont émergé d'un projet de standardisation des interfaces de programmation des logiciels destinés à fonctionner sur les variantes du système d'exploitation Unix. (source : wikipedia)

Le shell *Bash* a été conçu comme une mise en œuvre *open source* d'un shell conforme à POSIX. Il inclut également des facilités appréciées par les programmeurs de scripts et les utilisateurs en ligne de commande. Il donc devenu le shell par défaut des systèmes de type Unix et il est même disponible sous Windows 10. C'est pour cette raison que ce cours se concentre sur le shell *Bash*. Ce que vous apprendrez dans ce cours pourra donc être appliqué à tous les ordinateurs, et même aux smartphones ou aux tablettes fonctionnant sous Android.

## 4 Conclusion

Le fait d'utiliser un shell revient à effectuer une interaction avec l'ordinateur par un terminal. C'est à dire entrer des commandes à l'aide du clavier. De nos jours, où tous les ordinateurs ont la puissance nécessaire pour offrir une interface graphique, cela peut sembler archaïque d'utiliser un terminal. En fait il n'en est rien, car l'interaction par un langage textuel offre une richesse et une flexibilité sans limite pour décrire des actions.

Pour s'en convaincre on peut faire l'analogie suivante avec le langage humain : l'interface graphique est l'équivalent d'un petit livre "guide du voyageur", qui contient des phrases toutes faites pour une langue étrangère. Mais connaître la langue étrangère reste le moyen le plus efficace pour faire des phrases adaptées à tous ses besoins de communication, présents et futurs, donc même les besoins auxquels on ne s'attendait pas initialement et qui ne sont pas non plus prévus dans notre petit guide...

Eh bien, maîtriser le shell, c'est en quelque sorte maîtriser un langage pour communiquer avec un ordinateur. Si vous souhaitez être efficace avec l'outil informatique, vous savez maintenant ce que vous avez à faire.

### Open source

Un logiciel open source est un programme informatique dont le code source est distribué sous une licence permettant à quiconque de lire, modifier ou redistribuer ce logiciel. La diffusion et la modification du logiciel sont libres.

## Activité 0.4

# Guide d'utilisation de la console Weblinux

Dans ce cours, nous vous proposons d'utiliser une console pour compléter votre apprentissage du shell par la pratique. Cette console s'exécute dans votre navigateur web. Pour rendre cela possible, le navigateur web charge un système Linux qui lance ensuite directement une console avec le shell *Bash*. Ce système d'exploitation dans le navigateur n'est pas bien entendu aussi complet qu'un système Linux installé nativement sur un ordinateur. Cependant ce système que l'on appelle Weblinux reprend les principales caractéristiques d'un système d'exploitation de type Unix. L'objectif est de vous donner accès à un shell *Bash* et ce quelque soit le système d'exploitation de votre ordinateur. Avec le shell *Bash* disponible depuis la Weblinux, vous avez les moyens de reproduire facilement les manipulations mentionnées dans les vidéos ou les exercices. Par la suite, vous pourrez vous lancer, sans hésiter, sur un vrai système.

Ce document se propose de vous expliquer le fonctionnement et l'usage de la Weblinux. La section 1 présente cet outil. Le guide utilisateur de la console Weblinux est décrit dans la section 3. Malgré sa ressemblance à un système Linux, Weblinux souffre de limitations rappelées dans la section 4. Enfin, il se peut que vous rencontriez des problèmes. La section 5 peut vous aider dans ce cas là.

## 1 Présentation de la Weblinux

Ce cours vous offre un environnement d'apprentissage commun sans nécessité d'installation sur votre ordinateur. La Weblinux se télécharge et fonctionne sur n'importe quel navigateur web. Cette solution reste très légère et économique. En effet, il n'est pas nécessaire de procéder à une installation de Linux sur votre ordinateur. Weblinux vous donne l'équivalent d'un ordinateur virtuel Linux simplifié.

Une fois téléchargée, la Weblinux démarre et ouvre une console dans une fenêtre de votre navigateur. Notez qu'une fois chargée dans votre navigateur, cette Weblinux s'exécute localement dans votre navigateur et reste interne à votre ordinateur. La console de la Weblinux se présente selon la figure 0.4.1.

Afin de satisfaire la configuration d'un maximum d'utilisateurs, la résolution minimale de la fenêtre du navigateur de la Weblinux est de 800x600 pixels. La taille de l'écran de la console est quant à elle fixée à 26 lignes et 90 caractères.

Le démarrage de la Weblinux consiste à exécuter un noyau Linux dans  **votre**  navigateur. C'est pour cette raison qu'une liste de messages s'affiche sur la console et qu'un certain temps est pris avant que vous ne puissiez utiliser la console. Enfin, un système de fichiers est constitué dans la mémoire du navigateur. Il faut savoir que les fichiers que vous pourriez créer ne sont pas conservés d'un démarrage à un autre de la Weblinux. Il vous faut les sauvegarder sur votre ordinateur. Nous vous expliquerons cette manipulation dans la section 3.

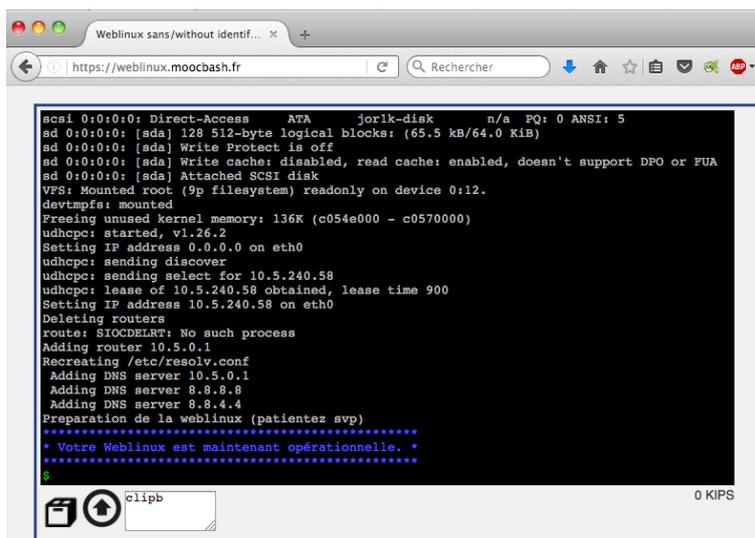


FIGURE 0.4.1: Console de la Weblinux dans une fenêtre d'un navigateur web.

Une fois chargée, la Weblinux reste en grande partie dans le cache de votre navigateur. Les invocations successives de Weblinux font que le système Weblinux se relance rapidement. D'où l'expression "économique" en bande passante.

## 2 L'accès à la Weblinux

Il existe plusieurs moyens pour accéder à la Weblinux, et nous avons mis à votre disposition 2 Weblinux différentes. Par défaut, la Weblinux ne requiert pas d'authentification et vous permet d'accéder directement au shell *Bash*. La seconde Weblinux s'utilise dans le cadre de l'activité 1.4. Elle fonctionne avec authentification de l'utilisateur, ce qui la rend plus proche de l'accès courant à un système d'exploitation de type Unix.

Concrètement, vous trouverez le lien pour démarrer la Weblinux à partir du menu général (bandeau supérieur) du cours sous l'onglet Weblinux. Vous pouvez aussi indiquer le lien de la Weblinux explicitement dans votre navigateur :

- <http://weblinux.moocbash.fr> pour la console sans authentification,
- <http://login.moocbash.fr> pour la console avec authentification.

Quelque soit la Weblinux retenue, vous serez identifié en tant qu'utilisateur *alice*. Nous reviendrons sur cette notion d'identifiant utilisateur dans l'activité 1.4.

Dès l'ouverture de la nouvelle fenêtre, et dès le premier téléchargement, le navigateur récupère le noyau de la Weblinux ainsi que quelques fichiers d'initialisation. Le temps de démarrage de la Weblinux est lié au temps de transfert et au temps d'exécution. Ces temps dépendent respectivement du débit de votre connexion Internet et de la puissance de votre ordinateur. A partir du second téléchargement, le code de la Weblinux est conservé dans le cache de votre navigateur. Le temps de transfert diminue ce qui a pour conséquence de rendre le redémarrage beaucoup plus rapide. Lorsque un caractère '\$' vert apparaît, cela vous informe que la Weblinux est désormais opérationnelle et que c'est à vous de jouer.

### 3 Tutoriel pour l'usage de la Weblinux

#### 3.1 Les boutons de l'interface de la Weblinux

Dans la partie inférieure de la fenêtre de la Weblinux, vous trouverez 3 boutons comme le montre la figure 0.4.2. Ces boutons sont, de gauche à droite :



FIGURE 0.4.2: Boutons de la fenêtre Weblinux.

1. une petite boîte ; ce bouton sert à sauvegarder le contenu du répertoire de l'utilisateur *alice*. Le contenu de la Weblinux est transféré vers votre ordinateur. Ce bouton trouve son utilité pour sauvegarder votre travail et récupérer l'ensemble des fichiers que vous avez créés.  
Attention : Pour des questions de performance, nous avons exclu de la sauvegarde les répertoires suivants : Movies, Books, Music, Pictures et imdb.
2. la flèche vers le haut ; ce bouton représente la fonction d'envoyer (*upload*). Il sert à transférer un fichier depuis votre ordinateur vers Weblinux. Vous vous en servirez pour restaurer votre travail (que vous avez préalablement sauvegardé).
3. un rectangle blanc ; ce rectangle représente un presse-papier (*clipboard*). Le texte issu du presse-papier de votre ordinateur sera collé comme une saisie dans la console de la Weblinux. Pour faire cette opération de copier-coller, vous devez tout d'abord copier le texte à partir d'une fenêtre de votre ordinateur. Puis, vous devez cliquer sur le rectangle blanc de la fenêtre de la Weblinux et à l'aide d'un clic secondaire, lorsque le menu apparaît, choisir l'action *coller* comme indiquée par la figure 0.4.3.

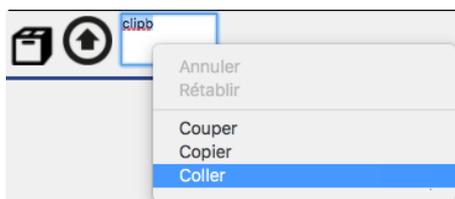


FIGURE 0.4.3: Coller dans la fenêtre Weblinux.

A droite de ces boutons, apparaît l'usage des touches de fonctions de votre clavier, tel des voyants d'activité. L'usage des touches `Shift`, `Control` et `Alt` est affiché.

#### 3.2 Entrer des caractères spécifiques à un système de type Unix

Au fil des activités de ce cours, vous allez découvrir des caractères qui ont des significations spéciales et particulières à Unix. Ces caractères ne sont pas toujours indiqués explicitement par des touches de votre clavier. Nous vous présentons ici comment les obtenir malgré tout.

##### Clavier Apple en français

Certains caractères usuels des systèmes d'exploitation de type Unix tels que le *pipe* (`|`) ou le *tilde* (`~`) s'obtiennent avec des combinaisons de touches. Le tableau ci-dessous précise la combinaison de touches pour ces caractères.

Caractères	Combinaisons de touches
~	 + 
	 +  + 
[	 +  + 
]	 +  + 
{	 + 
}	 + 
\	 +  + 

La correspondance des touches réalisée par Chrome, Safari ou Opéra impose que le caractère '|' se fasse avec la combinaison de touches  +  + . Alors qu'avec Firefox et comme c'est le cas dans un terminal, le caractère '|' s'obtient avec la combinaison  +  + .

### Clavier PC en français

Les caractères *pipe* ('|'), *tilde* ('~') ou *backquote* ('`') sont obtenus avec l'usage de la touche  située à droite de la barre d'espacement combinée avec une autre touche comme le montre le tableau ci-dessous :

Caractères	Combinaisons de touches
~	 + 
	 + 
[	 + 
{	 + 
]	 + 
}	 + 
\	 + 
^	 + 
`	 + 

## 4 Les limites de la Weblinux

Il existe quelques différences entre la Weblinux et une distribution Linux « complète ». La première différence porte sur les commandes. Toutes les commandes d'un système de type Unix ne sont pas disponibles sur le système Weblinux. De plus, pour alléger le chargement de la Weblinux, nous avons supprimé de nombreuses commandes inutiles. Nous avons retenu les commandes utilisées durant ce MOOC.

La seconde différence porte sur le manuel en ligne appelé couramment le "*man*". Nous avons décidé de mettre à disposition le manuel des commandes Linux, en français<sup>1</sup>. Cette aide en ligne explique, en français le plus souvent, le comportement et la syntaxe des commandes disponibles dans un système Linux classique. Nous reviendrons en détail sur l'aide en ligne lors de la première séquence. Mais attention, dans le cadre de l'utilisation de la Weblinux, l'usage de certaines commandes peut différer légèrement par rapport à la description faite dans le manuel. Pour retrouver l'usage réel des commandes disponibles, il faut faire suivre le nom de la commande par l'option `--help`. Par exemple, si vous tapez `mkdir --help`, vous obtiendrez l'aide de la commande `mkdir`. Notez que cette aide succincte est en anglais.

---

1. extrait de la distribution Ubuntu 16.04

Enfin le système de fichiers est émulé en Javascript et créé au moment du téléchargement de la Weblinux dans le navigateur. De ce fait, tous les fichiers portent la même date et il y a parfois un bug lors de la suppression de fichiers. De facto, la manipulation des dates de fichiers ne fonctionnent pas comme sur une distribution "normale".

## 5 En cas de problème

Si la Weblinux vous semble bloquée ou qu'elle est bien bloquée, nous vous suggérons les solutions suivantes :

- Tout d'abord vérifier que la fenêtre de la Weblinux est bien la fenêtre courante ou active. Plus précisément que la partie console est la zone active. Pour cela cliquez sur la console ou laissez le pointeur de la souris au dessus de la console en fonction de votre système de gestion de l'interface graphique que vous utilisez.
- Soyez aussi patient : votre navigateur est peut-être très occupé et l'exécution du système Weblinux très ralentie.
- Si malgré toute votre patience rien ne bouge, qu'il ne vous n'est plus possible de saisir des caractères, que le clavier semble inopérant, alors vous pouvez essayer d'interrompre l'exécution en cours par la combinaison de touches `CTRL` + `c` (appui simultané sur ces 2 touches).
- La situation n'est toujours pas redevenue à la normale? Il faut alors avoir recours au grand moyen : il faut redémarrer la Weblinux. Pour cela, il faut recharger la Weblinux soit en actualisant sa fenêtre soit en la fermant, puis en l'ouvrant à nouveau. Il va sans dire que cette solution entraîne la perte de tout le travail effectué lors de cette session. Pour se prémunir de ce désagrément, il peut être pertinent d'effectuer l'édition des fichiers sur son ordinateur puis de les envoyer ensuite sur la Weblinux pour la mise au point et leur exécution.
- Il faut savoir que la Weblinux "pèse" 95 Mo dans sa totalité. En fait, le noyau a une taille 2.2Mo et les principaux fichiers initiaux 3Mo. Ensuite chaque nouvelle commande est chargée dynamiquement. Ainsi, lors du chargement de la Weblinux, la première fois alimente le cache du navigateur. Le redémarrage de la Weblinux par le rechargement revient à recharger la page web de présentation et quelques scripts en Javascript pour moins de 1 Mo et à utiliser le cache du navigateur. Pour se retrouver dans la situation initiale ou pour charger la dernière version de Weblinux, il faut préalablement vider le cache du navigateur. La procédure de vidage du cache du navigateur est propre à chaque navigateur. Vous trouverez sur Internet ou dans l'aide de votre navigateur les manipulations à effectuer.

Quoiqu'il en soit, nous avons ouvert un forum dédié à tous les problèmes que vous pourrez rencontrer dans l'utilisation de la Weblinux. N'hésitez pas à nous signaler les éventuels défauts.

## 6 Conclusion

L'outil proposé vous permet d'exécuter un Linux dans votre navigateur. Nous espérons qu'il répondra à vos attentes et qu'il vous aidera dans l'apprentissage du shell *Bash*. Vous pouvez maintenant utiliser le *Bash* via la Weblinux et saisir des commandes. Mais n'anticipons pas, tout cela vous sera expliqué à partir de la première séquence.

Si vous avez encore des questions, n'hésitez pas à consulter le forum dédié à l'utilisation de la Weblinux mis à disposition spécifiquement pour cela.

## Annexe A : Spécifications techniques

Le noyau Linux est exécuté dans le navigateur, des scripts Javascript émulent les périphériques (carte réseau, périphérique d'entrée ou de sortie, carte graphique, clavier, etc...). La Weblinux est construite sur la base d'un émulateur Javascript pour processeur OpenRISC 1000 (abrégé en jOR1k). Cet outil est développé principalement par M. Sebastian Macke dans le cadre général d'un projet d'émulateur de microprocesseur et disponible gratuitement à cette adresse :

<https://s-macke.github.io/jor1k/demos/main.html>

Nous le remercions d'ailleurs pour avoir rendu disponible le code de cet outil en *open source*.