

OSlow : Un OS pour recycler les anciennes machines

Thomas B. @teajay@mastodon.social

Judicaël D. @judicael@upload.fedithese.net

Arthur L. @arthurlutz@mamot.fr

Alexandre G. @Sacha@upload.fedithese.net

RÉSUMÉ

Aujourd'hui un nombre croissant d'équipements numériques sont mis au rebus alors qu'ils sont encore opérationnels. Afin de prolonger leur durée de vie, notre point de départ est de s'interroger sur quel système d'exploitation minimaliste (avec quelles applications) choisir pour faire fonctionner les anciens ordinateurs en général et les anciens smartphones en particulier. Nous exposons notre parcours de réflexion et montrons comment l'application d'une démarche de low-technisation nous amène à redéfinir l'objet du projet initial. En partant du choix d'un système d'exploitation, nous arrivons à l'élaboration d'une encyclopédie des savoir techniques gérée comme un commun pour prolonger l'utilisabilité de dispositif numériques.

1. Sujet initial

Quel OS minimaliste (avec quelles applications) pour faire fonctionner les anciens ordinateurs en général et les anciens smartphones en particulier. Définir les spectres fonctionnels visés : ordinateurs (machines non connectées, clients, serveurs), transformation en périphériques (webcam, micro, GPS...)

2. Point de départ

Le point d'entrée choisi pour ce sujet semble un peu problématique du point de vue de la démarche lowtech. Dans l'approche lowtech, partir d'un besoin identifié pour construire une solution ajustée est essentiel. Ici, nous sommes dans un autre cas de figure. Nous avons un équipement technique "superflu" (en tout cas du point de vue de l'imaginaire collectif présent) et nous voulons lui trouver une place sur le "marché".

En caricaturant un peu, nous pourrions dire : "Génial, nous avons inventé le vieil ordinateur ! Mais à quoi ça pourrait servir ?".

Au delà de cette critique, nous pouvons supposer que ces équipements sont utiles et demandés, afin de satisfaire cette demande, nous cherchons un système d'exploitation pour permettre l'utilisation des outils numériques.

3. État de l'art

L'ensemble de la bibliographie est à retrouver sur [Zotero](#) .

De cette recherche ressort qu'il y a très peu voir aucune recherche académique sur le sujet de la prolongation de la durée de vie d'équipements électroniques par le maintien du support logiciel. Il existe un corpus important autour de la gestion du cycle de vie des équipements numériques (switch, serveurs, stations de travail...), mais dont l'objectif est la gestion financière des équipements visant à équilibrer au

mieux les comptes bancaires des industries, notamment en garantissant le moins de pannes possibles qui causeraient des dommages aux activités et revenus de l'entreprise. Une part importante de "rebus" informatiques sont produites par ce genre de stratégies préventives qui disqualifie des équipements opérationnels et rend nécessaire l'acquisition de nouveau matériel.

Nous avons également trouvé plusieurs études sur les mesures d'impacts énergétiques des équipements informatiques. Une grande partie se focalisent sur la consommation des centres de calcul. Ces recherches ont produit plusieurs outils pour mesurer la consommation d'énergie à l'usage. D'autres mettent en évidence le lien entre la consommation d'énergie et la qualité des logiciels, notamment au travers de la comparaison de l'efficacité énergétique des solutions produites par différents langages de programmations à des problèmes algorithmiques.

Finalement, aucun lien direct n'est visible entre les études trouvées et l'usage d'OS alternatifs pour prolonger l'utilité des dispositifs électroniques.

Certains fournisseurs d'OS ou de matériel ont un planning d'obsolescence programmé disponible pour leurs équipements. La durée de support maximale varie selon les fabricants et peut atteindre jusqu'à 10 ans au maximum. Au-delà certains communs permettent de prolonger l'utilisabilité des équipements.

Les références trouvées au sujet des OS adaptés aux équipements anciens, sont toutes issues de projets open source qui maintiennent dans leur base de code le support pour les vieilles plateformes.

4. Idéation

4.1. Recensement d'usages d'outils numériques

- fonctionnel
 - rédaction de documents
 - lecture
 - archives / stockage
 - mesure du temps / montre
 - repérage (gps / cartes)
 - connectivité (appareils réseau)
 - comptabilité
- support aux lowtechs
 - rédaction de doc technique
 - partage des savoirs
 - assistance conception (CAO)
 - assistance fabrication
- commerce
 - Bourses d'échanges
 - Gestion d'entreprise/association
 - Paiement à distance (Duniter)
- culture
 - audio / vidéo
 - écoute
 - création
 - diffusion
 - ludique/apprentissage

- jeux vidéos
- communication à distance
- clients de messagerie
- webconf
- participation à des salons virtuels
- Santé
 - sport
 - valeur nutritionnelles des aliments
 - calcul des besoins énergétiques par repas
 - méditation
 - suivi de cycle menstruel / fertilité

L'ensemble de ces usages font références a des usages courants actuels des outils numériques. Nombre de ces usages sont à mettre en perspective dans leurs contextes. Certains usages pourraient être remis en question par l'application d'une démarche Lownum/lowtech mais au niveau de la sélection d'un système d'exploitation il est difficile d'opérer ces choix, car le périmètre de "responsabilité" de l'OS n'englobe pas tous les usages. Des logiciels additionnels permettent d'augmenter le champ d'application de l'outil numérique.

Ce constat nous mène à poser la définition suivante d'un OS dans notre cadre : Un OS est un ensemble de composants logiciels permettant à des applications de fonctionner sur des plateformes matérielles diverses.

En général un OS est constitué :

- d'un noyau dont la fonction est d'offrir une interface logicielle unifiée aux composants matériels du système numérique
- un ensemble de bibliothèques supports permettant l'interopérabilité des systèmes numériques :
 - systèmes fichiers
 - algorithmes de compression/décompression
 - protocoles de communication
 - simplification de l'usage des fonctions du noyau dans la couche applicative
 - composant de bases d'interfaces hommes machines
 - ...
- un ensemble d'outils/services pour :
 - accéder aux données
 - organiser les données
 - installer/désinstaller des applications
 - paramétrer le système
 - diagnostiquer le système
 - interagir avec d'autres systèmes numériques

5. Ouvrir l'imaginaire / maquettage

Afin de mieux cerner les besoins il nous apparait nécessaire de construire du contexte à partir de la rédaction de scénario d'usages. Au vu de ce que nous savons de la situation écologique actuelle et de l'état des stocks de ressources naturels disponibles pour la fabrication d'équipement numérique et leurs usages nous formulons ces hypothèses pour guider nos scénarios :

- Les équipements doivent fonctionner avec des contraintes d'accessibilité à l'énergie

- La connectivité à un réseau n'est pas garantie ou intermittente
- Plusieurs utilisateurs peuvent partager l'usage d'un équipement

5.1. L'encyclopédie

Aujourd'hui, le soleil était au rendez-vous. Les batteries du netbook du centre devraient être presque pleines. Adrien se réjouit de pouvoir continuer à consulter les articles de l'encyclopédie concernant la statique des fluides. Adrien a besoin d'approfondir sa compréhension pour aider à la régulation des niveaux des réservoirs d'eau. En fin de journée, il se rend au centre informatique et démarre l'ordinateur "encyclopédie". Bien que ce soit un vieux coucou, son processeur démarre au quart de tour. En 20 secondes, l'ordinateur est allumé et il peut se logger sur la machine.

Zut, petite déception, la batterie n'est pas au max. C'est Flavie qui est passée avant lui et lui a laissé un message. 5 minutes pour répondre à Flavie et c'est parti pour l'étude. Il ouvre Klivix et reprend sa lecture là où il l'avait laissée hier.

C'est le bip d'indication du niveau de charge du netbook qui sort Adrien de sa lecture. Vite, il enregistre sa progression et laisse un message à Lucien, il aura finalement besoin des tuyaux dont il lui avait parlé.

C'est fini pour aujourd'hui, la batterie est vide. Finalement, ça a du bon qu'elle n'ait pas été chargée à bloc, il va pouvoir prendre le temps de faire le tour du lac au lieu de rentrer directement !

5.2. Gestion des stocks

Martine a trouvé une vieille machine peu encombrante qu'elle a installée sur son comptoir. Elle a préféré choisir cet outil robuste et simple d'utilisation pour faire fonctionner sa boutique. Elle y enregistre tous les articles et peut suivre facilement l'état de son stock non alloué. Martine ne vend presque que à 95 % sur commande. C'est une façon pour elle d'éviter le gaspillage. Elle commande en général une ou deux pièces en plus au cas où, mais jamais plus.

Ses clients viennent déposer leur commande sur le petit smartphone posé juste derrière la vitrine. Lors de leur passage, ils peuvent charger le catalogue de Martine et le mettre à jour pour le consulter en rentrant chez eux et reformuler leur prochaine commande. De toute façon, ils savent qu'il faudra quelques jours pour que leur commande arrive, donc inutile de s'éterniser dans la boutique, surtout si Martine est en train de conseiller un autre client :)

Tous les jours, elle relève les nouvelles demandes et organise au mieux les commandes auprès de ses fournisseurs. Avec ce système, elle n'a pas besoin d'avoir un site internet compliqué et payer une redevance. En plus, cela a l'avantage de limiter ses envois de colis à un périmètre local.

5.3. On met les voiles

Clink, clink, clink... la corde tinte en tapant le mat lorsque Méline hisse le pavillon de VakitaCorp. C'est le jour du grand départ. L'équipage a chargé et arrimé la cargaison du voilier cargo. Cette fois ils pourront sortir du port et du détroit sereinement grâce au sonar de récup que le capitaine a fait adapter au Hackerspace de Brest. Il leur a fallu trouver un vieux modèle de téléphone bien spécifique pour faire le traitement numérique nécessaire et l'interfacer comme une webcam à l'ordinateur principal.

Et oui, Méline est obligé de faire de la récup, il a investi toutes ses économies pour le bateau, et vu le prix des équipements électroniques neufs, c'était le seul moyen d'avoir accès à ces équipements. Méline n'était pas super à l'aise avec tous ces outils, mais la formatrice du centre de réaffectation électronique lui a appris à se servir de l'ensemble des fonctions de l'ordinateur :

- consulter les cartes de navigations
- connecter l'ordinateur au téléphone portable (de récup) pour voir sa position sur la carte
- consulter les prévisions météo
- consulter la bibliothèque embarquée, pour toutes les docs techniques, les recettes, etc.

- remplir le carnet de bord
- envoyer des e-mails

Il utilise aussi le téléphone de la cabine pour prendre des photos avant de faire une réparation, comme ça le soir il peut insérer des images dans le carnet de bord qu'il écrit sur l'ordinateur. C'est le plus simple pour avoir un bon suivi de l'état du bateau.

D'ailleurs hier lorsqu'il a vérifié les branchements des panneaux solaires du pont, il a bien pris le temps de faire une photo des connecteurs, au cas où il aurait besoin de s'en souvenir loin de la côte...

Hier, Ludivine lui a apporté sa collection de musique pour accompagner au moins la première partie de leur voyage. Elle avait un téléphone ultra moderne et le seul moyen qu'ils ont réussi à trouver pour transférer les fichiers, c'est avec un câble USB !

Espérons que dans les ports d'escales ce sera plus simple et qu'ils auront accès à Internet via du wifi. Méline en aura besoin pour transmettre son carnet de bord par e-mail, comme convenu avec les propriétaires de la cargaison. Ce sera aussi plus rapide de rapatrier les prévisions météo à 30 jours pour toute la prochaine étape. Si jamais ce n'était pas possible, il se débrouillera avec la connexion de donnée du téléphone de la cabine. Il a trouvé un opérateur téléphonique qui propose ce service dans tous les pays pour un prix acceptable, mais c'est en dernier recours.

Dès qu'ils seront sortis du détroit, ils vérifieront le logiciel de pilotage automatique installé sur l'ordinateur. Si tout va bien, ils auront des nuits plus faciles. Le réveil servira seulement à annoncer les quarts.

6. L'atelier de programmation educPop

Hayat et Mia se sont retrouvées, une dizaine d'années après la fin de leurs études, lors d'un festival du logiciel libre à Lyon. Elles se rendent alors compte qu'elles habitent à une dizaine de kilomètres à peine l'une de l'autre, dans deux villages dromois, et décident d'initier ensemble un atelier d'éducation populaire dédié aux questions numériques dans leur territoire d'adoption. Au fil des mois, elles ont réussi à agréger autour d'elles un groupe d'une demi-douzaine de volontaires pour construire brique par brique les trois principales activités de leur association :

- un atelier d'initialisation à la programmation à destination de collégien-ne-s
- une rencontre trimestrielle de dé-windowsisation organisée dans la salle des fêtes de la commune d'Hayat, et à destination du grand public
- une permanence bi-mensuelle d'aide à la réalisation de projets numériques, qui est consacré chaque année à des groupes de quelques personnes, plus initiées, porteuses de projets très spécifiques ; il s'agit cette année du club de danse et de musiques médiévales, qui a délégué trois de ses membres auprès de l'association pour mettre en place un site de podcasts pour partager leurs découvertes et pratiques au-delà des limites de l'intercommunalité

Assez vite, quand il s'est agi de rendre opérationnels les terminaux mis à disposition par les collectivités locales ou rachetés à prix modique auprès d'entreprises du bassin, le problème de l'obsolescence des matériels récupérés a conduit nos deux comparses à privilégier des systèmes d'exploitation légers, libres et modulaires. Dans le local de l'association, le parc "recyclé" de l'association fait la part belle à tous types de postes :

- une dizaine de postes fixes, de tailles et âges très divers, sont dédiés aux cours de programmation, et tournent sous la version de bureau de l'OS OSlow : un gestionnaire de fenêtres minimaliste, un IDE open-source et la suite éducative [KTurtle](#) pour l'édition de code, un navigateur web léger et un simple émulateur de terminal pour tout le reste
- deux tours de PCs plus puissantes tournent sous la version serveur de l'OS OSlow et n'embarquent pas d'interface graphique ; ils permettent de faire tourner l'infrastructure de l'association et fournissent en prime quelques services tels qu'un repo git pour le partage et le versionnage de code et une instance Mobilizon pour l'organisation interne des événements
- un ordinateur portable, enfin, embarque la même suite logicielle que les postes fixes, et sert d'appareil-témoin en même temps que d'outil de travail aux bénévoles de l'association lorsqu'ils tiennent des stands dans divers salons du monde du libre et de l'éducation militante

Le caractère modulaire et minimaliste du micro-noyau d'OSlow a permis d'utiliser cet OS comme base de travail sur des projets spécifiques des sessions précédentes impliquant de construire sur la base de micro-contrôleurs (ex : Arduino) des dispositifs tels qu'un **voltmètre**, un **oscilloscope** ou encore un **lecteur de cartes à puces**. Cette année encore, les porteurs du projet de site de podcasts ont pu vérifier la pertinence du choix d'OSlow pour mettre en place les outils nécessaires à leur chaîne de production (prise de son, montage audio, développement et administration web) sans changer d'ordinateur inutilement.

7. Évaluation

7.1. Analyse

Quelles hypothèses / présupposés se cachent derrière l'idée d'un OS permettant de maintenir fonctionnels des matériels jugés obsolètes ?

C'est à partir de cette questions nous avons élaborés la carte mentale, présenté ci-dessous.

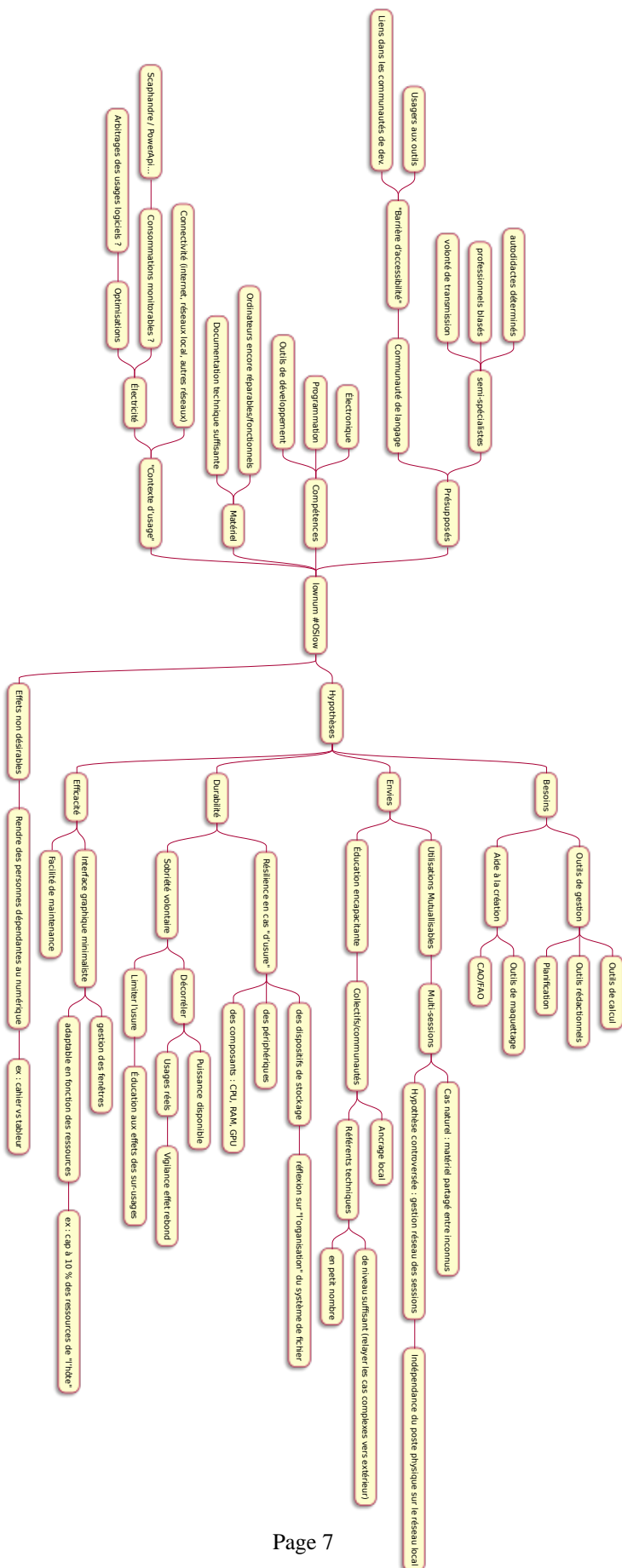
En premier lieu nous avons identifié la question de l'utilité sociale. Les usages que nous avons actuellement sont chronophages et sont possibles par la disponibilité de temps pour l'usage du numérique. Est-ce que cette disponibilité sera toujours aussi grande ? De plus, la maîtrise des outils informatiques demande un temps d'apprentissage important et dans le contexte actuel une réactualisation des connaissances permanente, que ce soit pour les utilisateurs ou les concepteurs. Implicitement nous supposons que le temps nécessaire sera alloué à cette tâche.

Au-delà du partage des connaissances il peut aussi être envisageable de partager des équipements entre usagés. OSlow doit permettre de mutualiser de l'espace de stockage, du temps de calcul, et des périphériques tels que des imprimantes, scanners, webcam, etc.

La question de l'accès aux composants matériels nécessaires pour l'utilisation et la réparation d'outils numériques est également posée. Sans composants pour réparer un matériel, il sera mis au rebut. Cette question est en lien fort avec la couche de l'OS, car c'est elle qui est responsable de l'adaptation aux composants physiques de l'outil numérique. Si des éléments sont substituables, tels que dans les plateformes "desktop" avec des cartes d'extensions, la durée de vie d'un équipement sera plus facile à prolonger. OSlow doit pouvoir s'adapter aux substitutions de matériel, ceci sans faire un usage démesuré de ressources (CPU, RAM, stockage) à chaque usage. Par exemple l'utilisation de modules sur la plateforme Linux est indispensable pour satisfaire ce critère. Un noyau monolithique serait contreproductif à cet égard. Cependant, l'usage de modules coûte en termes de temps de configuration et en espace utilisée pour le stockage des modules dont seulement une partie est nécessaire pour une plateforme donnée.

La flexibilité et l'adaptabilité s'appuie sur l'accessibilité à plusieurs types de ressources :

- le stockage pour le matériel et les logiciels
- des réseaux de diffusion du matériel et du logiciel (revendeurs, louers, Internet, DVD, clefs USB...)
- des réseaux de construction et de diffusion des connaissances au niveau utilisateur et concepteur :
- Education générale et populaire
- Internet
- Livres
- Cercle social
- les documentations techniques pour le développement et le support de OSlow concernant les technologies matérielles et logicielles
- le temps et les compétences pour la maintenance des composants logiciels
- les outils matériels et logiciels nécessaires pour le développement et la maintenance
- l'énergie nécessaire pour faire fonctionner les équipements et réseaux de diffusions



Etant donné l'étendue du champ technique du numérique il est indispensable que OSlow s'appuie sur des ressources existantes qui permettent de mutualiser au maximum les efforts de maintenance des logiciels.

Comme nous l'avons vu avec la variété des scénarios, il est impossible de prévoir quel serait le champ d'usage d'OSlow, tant du point de vue matériel que fonctionnel. Le matériel pourrait être un ordinateur, un smartphone, un vieux routeur ou un autre dispositif. Le champ d'application influe également sur les périphériques et leurs protocoles associés à mettre en oeuvre, ce qui demande une souplesse d'extension et d'appropriation des usages.

7.2. Expérimentation

Afin d'évaluer la faisabilité de choisir un OS pour un équipement, nous avons procédé à l'évaluation de plusieurs distributions de Linux sur un petit ordinateur "obsolète". Il s'agit d'un ASUS EeeBook X205TA :

- Intel Atom Z3700, 1.33 Ghz 4 Coeurs, 2 Mo de Cache
- 2 Go de RAM DDR3L (1333 Mhz)
- 32 Go de Flash embarqué
- Intel HD Graphics
- Wifi 4 (802.11n) intégré
- Système audio SonicMaster
- Écran 11.6", 1366 × 768
- Batterie 38 Wh

La batterie sur cet ordinateur étant hors service elle a été déconnectée et il a donc fonctionné uniquement sur secteur.

Au vu des recherches documentaires, nous avons envisagé d'évaluer la consommation d'énergie de l'OS, une fois démarré, dans un état stationnaire de repos. En effet l'énergie consommée agrège de manière synthétique plusieurs paramètres de performances :

- utilisation du CPU
- utilisation de la RAM
- utilisation des périphériques de stockage
- utilisation de l'affichage
- indirectement donne une indication sur la complexité et l'efficacité des couches logicielles mises en oeuvre

L'outil Powertop permet d'effectuer une estimation de la consommation d'énergie sur un laps de temps donné. Cet outil développé par Intel, utilise les fonctions de suivi de consommation intégrés au processeur. Il ne s'agit pas dans notre cas d'avoir des mesures exactes mais des ordres de comparaison. Malheureusement sur plusieurs distributions la fonctionnalité du noyau Linux nécessaires à la mise en oeuvre de powertop n'était pas disponible.

Nous avons donc réorienté nos critères de comparaison :

- Espace disque utilisé pour l'installation
- Utilisation de la RAM après démarrage
- Utilisation du CPU après démarrage
- Temps de démarrage du système

7.2.1. Préparation du média d'installation

De manière à pouvoir installer différentes distributions de Linux à partir des fichiers ISO, nous avons préparé une clef USB avec l'outil MultibootUSB. Il s'est avéré impossible de démarrer à partir de cette clef, car la plateforme utilisée requiert l'UEFI. La clef a été modifiée en réinstallant GRUB avec le support de l'UEFI. Nouvel échec !

Après de laborieuses recherches, un post sur un forum datant de 2017 a permis de trouver la solution : le système de démarrage de cet ordinateur requiert l'UEFI 32bit. L'installation que nous avons effectué précédemment était avec une version 64bit de l'UEFI.

Finalement, nous avons préparé la clef d'installation avec l'outil [Ventoy](#) qui supporte tous les systèmes de démarrage.

7.2.2. Évaluation des distributions PuppyLinux

Le système de distribution PuppyLinux est conçu pour être utilisable sur des vieux ordinateurs. C'est un système de distribution qui s'appuie sur d'autres distributions en les adaptant spécifiquement pour des vieux appareils. Des outils frugaux et fonctionnels sont fournis pour les tâches courantes. Ces distributions n'ont pas forcément vocation à être installées sur un ordinateur mais visent plutôt à être utilisées à partir d'un média de stockage flash (en général une Clef USB). Cependant une option d'installation est disponible.

Il s'est avéré que le support de l'UEFI n'est pas disponible sur ces distributions ce qui compromet le test avec une installation sur le PC de test.

7.2.3. Évaluation de version de Antix Linux

La distribution Antix Linux est basé sur une distribution debian, adaptée pour des ordinateurs anciens. Il existe plusieurs variantes que nous avons comparées :

- 32Bit/64Bit
- Système d'initialisation runit ou init.d

Cette distribution a pu être installée sur le support interne, cependant il a fallu désactiver le montage automatique des volumes reconnus pour que l'installation puisse être finalisée.

Une fois la distribution installée, ni le wifi, ni le son fonctionnait. Des posts sur divers forum de distributions Linux ont permis de rendre le wifi opérationnel. Le son n'a pas pu être mis en service car le chipset son n'est pas supporté dans la version du noyau Linux utilisé par Antix.

Une fois installé l'utilisabilité n'est pas très satisfaisante, les temps de réponse de la souris et du clavier laissent à désirer. Il existe probablement une solution... mais il faut la trouver. Par faute de temps, seul la variante utilisant runit comme système d'initialisation a été évaluée.

Pour la version 32bit :

Temps de démarrage	22 s
Utilisation du disque après installation	3.002 GB / 11%
Utilisation RAM (MiB)	76Mb / 1862.3Mb soit 4.08%
Utilisation CPU	1.6%

Pour la version 64bit:

Temps de démarrage	21 s
Utilisation du disque après installation	2.752 GB / 11%
Utilisation RAM (MiB)	108Mb / 1932.3Mb soit 5.59%
Utilisation CPU	1.6%

7.2.4. Évaluation de Void Linux

La distribution Void Linux est une distribution générique de Linux, elle n'a pas été conçue pour des anciens ordinateurs. Cette distribution a pu être installée sans encombre, le wifi était opérationnel sans paramétrage additionnel. Le système de son a nécessité des adaptations, qui ont demandé un investissement en temps important. Des connaissances avancées sur l'architecture logiciel de Linux et de ses composants systèmes ont été nécessaires pour trouver les informations pertinentes.

L'utilisabilité est très satisfaisante, pour toutes les taches simples aucune différence n'est perceptible par rapport à un ordinateur plus puissant ou récent.

Temps de démarrage	20 s
Utilisation du disque après installation	2.1 GB / 8%
Utilisation RAM (MiB)	197Mb / 1854.5Mb soit 10%
Utilisation CPU	3.6%

7.2.5. Évaluation de Debian 11

La distribution debian est une distribution Linux d'une grande maturité et grande stabilité, elle propose un nombre important de variantes avec différents environnements de bureaux. Pour être comparables avec les autres distributions, la variante avec l'environnement de bureau xfce a été testée. Cette distribution n'est pas spécifiquement dédiée aux équipements anciens.

Temps de démarrage	13 s
Utilisation du disque après installation	10.195 GB / 38%
Utilisation RAM (MiB)	495Mb / 1911Mb soit 26%
Utilisation CPU	0.6%

7.2.6. Comparaison des distributions

Critère	Antix 32bit	Antix 64bit	Void	Debian 11
Temps de démarrage	22 s	21 s	20 s	13 s
Utilisation du disque après installation	11%	11%	8%	38%
Utilisation RAM	4%	5%	10%	26%
Utilisation CPU	1.6%	1.6%	3.6%	0.6%

Bien que les mesures effectués soit très approximatives, nous voyons clairement que chaque distribution utilise une part plus ou moins grande des ressources matériels disponibles. Une distribution de Linux stable et bien rodée ne veut pas forcément dire frugale. Dans l'ensemble les logiciels ont tendances à être écrits pour utiliser l'ensemble des capacités de calculs disponibles des nouveaux équipements. Ceci conduit à produire des logiciels toujours plus demandeurs en ressources et qui risquent de rendre obsolète des équipements encore fonctionnels lors de leurs déploiement. La disponibilité de logiciels et de systèmes d'exploitation open-source permet d'ajuster dans de multiples contextes le logiciel aux ressources disponibles. Cependant le choix d'un système d'exploitation ou d'une distribution qui corresponde à tous les contextes semble difficile.

7.3. Critères de convivialité

Pour continuer l'évaluation du projet OSlow, il nous semble important de prendre en compte l'aspect convivial au sens d'Illich.

7.3.1. Définition

Ivan Illich propose une définition de la convivialité :

L'outil est convivial dans la mesure où chacun peut l'utiliser, sans difficulté, aussi souvent ou aussi rarement qu'il le désire, à des fins qu'il détermine lui-même. L'usage que chacun en fait n'empêche pas sur la liberté d'autrui d'en faire autant.

S'appuyant sur les travaux d'Illich, l'internationale convivialiste a étendu cette définition dans "Le second manifeste convivialiste". Nous résumerons pour notre propos les 6 axes de la convivialité en ces termes :

- compatibilité avec la sphère du vivant (n'accapare pas de ressources excessives et ne compromet pas l'épanouissement des êtres vivants)
- non-évicition : l'usage de l'outil par une personne ne fait pas obstacle à l'usage d'un même outil par une autre
- non-capture : l'outil ne place pas l'utilisateur en position de ne plus pouvoir se passer de lui (de manière "réelle" ou perçue)
- autonomie de l'utilisateur : l'outil ne se donne pas à lui-même ses propres fins mais reste subordonné aux buts poursuivis par l'utilisateur
- accessibilité matérielle et cognitive : question des barrières économiques et cognitives à l'appropriation de l'outil
- sobriété : tant dans les usages que dans l'utilisation des ressources

7.3.2. Problématisation

La définition de l'outil convivial proposée par Illich se caractérise comme accordant la primauté à des subjectivités humaines autonomes. On peut mieux le saisir en la mettant en regard d'approches alternatives du rôle de l'outil logiciel :

- conception marchande : le logiciel comme source de maximisation d'un profit économique, qui doit d'une part générer du revenu, et surtout inscrire ses utilisateurs dans un certain rapport de dépendance permettant de pérenniser le processus de génération de revenu
- conception technocratique : le logiciel comme moyen de canaliser les pratiques sociales des utilisateurs en les médiant par des outils et procédures normalisés, par exemple à des fins d'amélioration de la mesure et de la prévisibilité desdites pratiques

L'approche lownum / conviviale vise à rendre accessible et utilisable l'outil informatique de manière non prescriptive (non obligatoire pour effectuer une tâche), laissant libre choix des options techniques à l'utilisateur en lui permettant de prendre en compte, l'accessibilité des ressources matérielles, des compétences et des savoirs dans un environnement "immédiat".

Au vu de ce qui précède, il nous paraît infaissable et contreproductif de définir et prescrire une solution unique pour tous les matériels et usages du numérique. Ce qui apparaît dans notre cheminement est que l'élément central pour prolonger l'usage d'outils numériques est d'organiser l'accès aux ressources, tant matériel que culturels. Il s'agit notamment de préserver les connaissances et compétences techniques associées au numérique, sans quoi l'utilité des dispositifs sera nulle. Ceci représente un effort colossal, déjà en partie réalisé par de nombreux projets open source, tels que Linux et ses distributions. La mutualisation entre ces projets a fait ses preuves et remplit déjà l'objectif de prolonger la durée de vie d'équipements.

Cependant le caractère morcelé des projets rend difficile la recherche d'information pour chaque type de matériel et également incertaine la pérennité des informations sur le matériel et la compatibilité logicielle. De plus l'accessibilité au réseau Internet est absolument nécessaire pour avoir accès au savoir partagé.

8. Reformulation du projet OSlow

Nous choisissons donc de reformuler le projet OSlow, afin de proposer :

- une approche non prescriptive d'OS
- une mutualisation des savoirs et leur pérennisation
- une souplesse pour s'adapter aux ressources locales

OSlow est une plateforme distribuée qui permet d'organiser les ressources et savoir nécessaires pour la sélection et la mise en oeuvre de systèmes d'exploitation adaptés aux ressources matérielles disponibles et aux besoins identifiés. À partir d'un besoin ou d'un équipement vous trouverez des options d'usage d'OS éprouvées et partagées par d'autres usagers.

8.1. Nature des contributions

Les contributions visent à évaluer et décrire les options d'usages à partir d'un matériel. Idéalement une recherche d'un modèle d'ordinateur doit permettre de trouver des indications sur quel OS est utilisable et comment le mettre en oeuvre.

Pour être utilisables les contributions devraient contenir un corpus commun d'information :

- Grille d'évaluation de la convivialité de l'OS
- Documentation matériel
- Procédures d'installation
- Procédures de recherches/solutions d'erreurs
- Hyperliens vers ressources pertinentes déjà existantes et mobilisables
- Profils d'usage (associations besoins / outils):
- quelles utilisations du numérique ?
- quelles utilisations d'outils non-numériques ?

8.2. Hébergement distribué

Les contributions alimentent dans un premier temps l'instance OSslow sur laquelle l'utilisateur est directement inscrit. Un mécanisme de synchronisation des contributions permet dans un second temps la diffusion des contributions à d'autres instances d'OSslow. Le travail est collaboratif et nécessite des processus de revue et d'intégration des contributions de manière à maintenir la qualité de l'information proposée.

8.3. Grille d'analyse du niveau de convivialité d'un système d'exploitation

Ci-dessous quelques critères qui nous semblent pertinent d'évaluer et de documenter pour évaluer la convivialité d'un OS.

8.3.1. Compatibilité avec les ressources disponibles

- Compatibilité hardware
- Compatibilité firmware
- Compatibilité avec le pool de compétences accessibles localement
- Documentation pour le matériel et les logiciels :
- disponibilité
- qualité

8.3.2. Utilisabilité

- Interfaces proposées
 - clavier
 - souris
 - textuelle
 - graphique
 - synthèse vocale
- Adaptabilité du système à mes capacités physiques (ex : cas d'un utilisateur non-voyant)
- Type d'utilisateur ciblé par le pré-paramétrage : novice | avancé ?
 - choix proposé par le logiciel d'installation de la distribution ?
- Niveau d'accompagnement dans la prise en main

- Écran de démarrage pour les premiers pas ?
- Accès hors-ligne et "intuitif" à la documentation logicielle
- Installation et utilisation complète possibles en hors-ligne

8.3.3. Adaptabilité

- Possibilité de choisir les logiciels pour les tâches à effectuer
- Simplicité d'adaptation du paramétrage des interfaces aux préférences de l'utilisateur
- Simplicité d'adaptation aux usages : environnement bureau / serveur / embarqué
- Cas des OS nomades : compatibilité avec un large panel de postes de travail
- Portabilité de son environnement de travail d'un équipement vers un autre
- Inter-compatibilité avec les systèmes tiers (dont OS propriétaires : Windows, OSX)

8.3.4. Non-capture

- logiciel libre : code ouvert, droit à la modification, droit au partage
- communauté ouverte
- inter-compatibilité (migrations possibles vers systèmes tiers)
- absence de produits/services imposés
- non conditionnement des usages à l'autorisation explicite ou implicite à une organisation d'exploiter les données d'usage
- conception non addictive

8.3.5. Non-éviction

- licence non limitée à un équipement ou à un utilisateur nommé
- OS non monopolisateur des ressources réseau ou périphériques de l'environnement (bande passante, accès aux imprimantes partagées, etc.)
- Facilité de contribution (vs barrières à l'entrée dans la communauté des contributeurs)

8.3.6. Efficience

- rapport service rendu / ressources (index synthétique : énergie)
 - Disponibilité des outils de mesure des consos énergétiques
- Temps de démarrage acceptables : max 1 minute
- Usage des ressources pour l'OS proportionnées au matériel (après boot sans applications):
 - 10 % RAM disponible max
 - 20 % RAM GPU / Interne si pas de GPU
 - 3 % CPU max
 - GPU facultative
- Espace de stockage occupé par l'OS seul < min(10 %, 4 Go)
- Efficacité de la gestion de la bande passante pour les mises à jour logicielles
 - Taille des binaires proposés au téléchargement
 - Mutualisation des ressources nécessaires à la compilation de l'OS et des paquets (centralisée vs distribuée)
 - Possibilité de mise à jour hors ligne à partir de paquets pré-téléchargés

9. Conclusion

Au lieu d'avoir identifié un OS, nous arrivons à la définition d'un outil associé à un commun de savoir technique (semblable à Wikipedia). Ce point d'arrivée que nous touchons est inattendu par rapport à l'énoncé initial du projet. Pour nous cette issue montre la pertinence et la plus-value du processus de "low-technicisation" dont la partie critique et évaluation est centrale. Le fait de mettre cette phase en amont de la phase de réalisation et même de prototypage donne plus de souplesse pour réorienter une solution avant d'être engagé dans un effort de réalisation, point d'entrée dans le processus d'écrit par la "théorie de l'engagement". Le choix de ne pas seulement se référer dans l'évaluation à des critères techniques, mais également à des critères philosophiques et politiques étend les opportunités d'identifier les besoins sociaux profonds. Le convivialisme étendue (au sens d'Illich) nous semble précieux dans cette démarche.

10. Annexes

Pour apporter un aperçu de notre cheminement, voici les notes prises lors de nos échanges.

10.1. Échange du 13/05/2022

Participants :

- Judicaël
- Thomas
- Alexandre

Judicaël :

Découverte du cours à la journée du logiciel libre, Dev pour des logiciels d'épicerie bio (propriétaire) *Thomas :*

Arrivé sur le cours suite à une réflexion sur les techniques qu'il développe. Quitte l'ingénierie définitivement au mois de juin. Le cours est un petit peu un moyen de redonner de la valeur à la technique. *Alexandre :*

chercheur indépendant, en cours de travail sur un projet de recherche ; domaine de recherche : low-technicisation, avec des projets de recherche sur la low-tech et ses composantes sociales / philosophies technocritiques (dont, mais pas seulement, Illich); pas mal d'activités en parallèle ; envie d'aller vers un débouché assez concret au projet ; utiliser les connaissances techniques et la programmation informatique pour mieux réutiliser les très nombreux devices numériques en "jachère" dans nos sociétés

- adresse à Thomas : que penses-tu de réutiliser tes compétences techniques dans des collectifs alternatifs dans le secteur numérique ?

Thomas :

Question de départ de Thomas : est-ce que la dimension technique est vraiment la clé par laquelle appréhender nos questionnements ? Ne risque-t-on pas, en inventant des solutions, de créer des usages sur-numéraires ? Exemple issu de sa vie pro : la demande d'une connection usb pour lave-vaisselle.

- en termes positifs : comment accompagner à des usages plus sobres / raisonnés ? #education_populaire #emmabuntus.org

Alexandre :

dans ses terrains d'observation des low-techs, constat d'une difficulté à passer à l'échelle / industrialiser -> en réaction, accompagnement sur la phase de conception, avec un travail réflexif sur les valeurs et les contraintes à en tirer (ressource à partager : une cartographie des valeurs)

judicael :

Sur la question des usages et des publics, dans mon cas je travaille pour un outil de gestion des bio-coop. On leur vend des logiciels propriétaires et ils nous font des retours. Mais au niveau du modèle économique, c'est viable uniquement parcequ'on travaille avec Biocoop qui a les moyens de payer. Cette question des gains de productivité des outils d'informatique me semble compliqué à résoudre. Il y a un décalage entre les besoins sociaux qui ont besoins d'être adressé et les outils qui sont destinés à de gros acteurs du

marché.

Thomas :

Pour rebondir, dans Illich il y a une bascule entre l'utilité de l'outil et l'aliénation de l'outil. Illich, dit qu'à partir d'un moment l'outil prend la main sur l'utilisateur qui devient sujet de l'outil. Ça rend très difficile ce qu'on fait car choisir un OS oldschool pour recycler les machines c'est possible mais pourquoi faire ?

Judicael :

Pour prendre la suite de ce que tu viens de dire et par rapport aux outils, cela permettrait de virtualiser un service déjà existant et donc de "forcer" les utilisateurs à acheter un PC et virtualiser le service.

Thomas :

cela pourrait être plus justifiable, si on virtualise le service non pas pour les utilisateurs mais pour les personnes qui fournissent le service.

Alexandre :

cours donné au CNAM sur les low-techs, parti-pris de ne pas commencer par la question des besoins -> volonté de ne pas commencer par une réflexion trop large/noyante ; scepticisme sur la problématique des besoins essentiels ; alternative : réflexion de Monnin sur les impacts des outils plutôt que sur la nécessité a priori des besoins En reliant à la question low-tech : si l'usage proposé en low-tech n'impacte que moi, c'est ok // problème parallèle : capture des capacités par des publics spécialisés d'informaticiens ; à nous d'imaginer et réfléchir dans leur contexte aux usages possibles de nos propositions.

Thomas :

l'exercice des scénarios va dans ce sens

Alexandre :

propose un focus sur l'autonomie et l'impact environnemental (avec en background : le coût d'extraction)

Thomas :

vigilance sur le ratio service rendu / coût de performance des outils (cf les navigateurs web modernes) -> axe implicite : minimalisme. Exemple : sur un petit PC avec un OS minimaliste, tout démarre bien jusqu'à l'ouverture du navigateur

Judicael :

Sur une distribution, il faudrait lister tous les outils qui permettraient d'avoir des usages minimaliste. Idée d'atelier d'éducation populaire pour qu'il puisse revoir leur rapport à l'outil (scratch). En pratique on utilise pas grande chose du potentiel.

Alexandre :

sur l'idée de s'emparer du potentiel des ordinateurs -> intéressé ; sur les usages, cf le pharcon : (remède | poison) Culture technique et communs : comment faire des communs sur des ressources numériques ? problématique de la tension entre communalisation et rétribution du travail Question : peut-on faire tourner comme serveur un device sur un OS minimaliste et pour exposer des services faits main ?

Thomas :

ça dépend d'où tu pars... ex d'un Alcatel sous Android, avec pour visée de le repasser sous Linux (PostMarketOS), mais exposition à un coût d'entrée important pour porter l'os sur le hardware ciblé Passé le niveau du hardware, dans le cadre de l'OS installé : les choses redeviennent plus accessibles à quiconque a été suffisamment formé à la programmation (faire le deuil des couches de simplification liées au GUI). Actuellement l'apprentissage de l'informatique n'est pas axé vers l'autonomie des individus. Initiation à l'informatique avec le langage Logo (alternative textuelle à Scratch) : + value -> démystifier l'usage du clavier, les interfaces de type console

Alexandre :

point de convergence == [ataraxie](#) L'objectif d'Alexandre : réaliser la transition sociale et écologique. Dans cet objectif : y a-t-il besoin d'expliquer le numérique ? Si oui, où sont les ressources pour le faire ?

(#education_populaire)

Thomas :

sur l'éducation populaire, souligne le besoin d'un dialogue dans des communautés de taille humaine entre des initiés et des utilisateurs qui ne le sont pas

Alexandre :

problématique de l'approche auto-didactique → par où commencer ?

Thomas :

Aussi il y a la rapidité de l'obsolescence des connaissances. On est tout le temps sur une base mouvante et il faut se réactualiser toujours pour rester maître de l'outil. On devrait se poser. Dans les outils de bases de Linux, il y a un corpus de fonctions qui servent un seul but et qui fonctionnent encore.

Alexandre :

rediriger le projet vers des solutions durables/statiques, avec des langages. Prendre position sur le projet en choisissant des solutions informatiques qui ne deviendront pas obsolètes.

Judicael :

Il y a des langages qui ne bougent pas beaucoup comme le C++, le problème c'est qu'il a un niveau d'abstraction assez faible et du coup qui coûte cher pour développer un nouvel outil ; Il y a une tension entre la charge mentale humaine et l'abstraction.

Thomas :

La magie des outils UNIX, c'est de penser des ensembles de fonctions suffisamment simple pour résoudre un sous problème et ainsi permettre, par agencements de bloc, de résoudre un problème complexe. Mais c'est difficile de délimiter la fonction. Sur la question des langages de programmation pas forcément pertinent, mais il faut identifier les outils et fonctionnalités proposés.

Judicael :

Du coup, on revient au travail de conception : définir des fonctionnalités, penser les limites du cas d'usages. C'est la profondeur de la réflexion de la conception qui va guider l'usage du produit (?)

Thomas :

En découpant les différents scénarios et en regardant ce qui est commun, on va pouvoir déterminer quels blocs sont intéressants.

Alexandre :

les scénarios → permettre de faire émerger des fonctionnalités communes ; mais ces fonctionnalités comme telles, devront être subdivisées pour permettre l'identification des blocs que l'on recherche. Gros atout pour autonomiser les gens, via l'apprentissage / la prise en main des blocs dont les fonctionnalités nous intéressent

Thomas :

ok, mais à condition de bien investir dans l'accompagnement

Alexandre :

deux dimensions → transmission humaine (éducation pop') | autoportage / documentation capacitante

10.2. Rencontre du 17/05

Participants :

- Judicaël
- Thomas
- Arthur

Auto-présentation d'Arthur Dev orienté DevOps, libriste (18 ans d'XP dans une SLL), intérêt pour les questions environnementales et énergétiques, pour les low-tech (#LowTechLab). Antérieurement :

participation au Mooc Numérique responsable. Tropicisme perso très technophile, mais apprécie le pas de côté permis par le MOOC Low-Num.

Judicael: Questions comment chercher des infos sur :

- ou trouver des docs sur le choix d'un OS
- performance énergétique micro kernel/kernel monolithique
- ergonomie dans le choix du desktop

Arthur: Voir si les outils qu'on connaît ont des références sur des papiers scientifiques.

- sur les consos énergétiques :
- PowerApi (suivi des consos machine, tant en datacenter qu'en poste desktop) <https://powerapi-ng.github.io/>
- Sources académiques : <https://powerapi-ng.github.io/#publications>
- Scaphandre <https://github.com/hubblo-org/scaphandre>

Thomas Rappel de la problématique posée : que faire d'intelligent avec du matériel obsolète ? Lien avec la conférence "Empreinte fantôme" : questions à se poser sur la pertinence des usages proposés et sur les éventuels impacts cachés ? À partir de la scénarisation des usages, on peut mieux percevoir les interactions avec le contexte et l'empreinte fantôme. Cible perçue de l'exercice : prépondérance de la dimension réflexive sur la démarche low-technicisation, davantage que d'éventuelles propositions techniques.

Judicael: prise de recul sur l'usage du numérique, est-ce juste de mettre le numérique partout ?

Arthur BetaGouv -> une approche de conception de services iso en numérique et en non-numérique. Mise en parallèle de services non numériques.

Judicael: Un chemin désirable : la décomplexification. Comment l'appliquer au numérique ? Retour vers des pratiques plus encapacitantes.

Thomas :

sur l'idée de réappropriation – la simplification apparente n'est pas ce qui permettra aux utilisateurs de s'approprier leurs outils -> démarche low-tech = plutôt, trouver les moyens de réorienter sa machine vers des usages souhaités + sur le degré de spécialisation : plutôt une approche de pool de personnes ressources à portée de main dans le cercle social des utilisateurs (bémol sur l'approche locale : jusqu'à un certain degré de profondeur, contre-exemple -> le noyau, qui demeure d'échelle plus globale, mais demeurant accessible via des communautés ouvertes).

Alexandre :

CollapseOS -> comment reconstruire l'informatique après l'effondrement ? Approche minimaliste de ce qu'est un ordinateur. Limite perçue néanmoins : désindustrialiser totalement l'informatique == une impasse. Trouver la juste mesure dans les savoirs à se réapproprier en priorité.

Thomas :

y voit un parallèle possible avec l'idée évoquée par Carnino de l'absence historique de points de ruptures abrupts -> pour CollapseOs : approche réductionniste peut-être extrême, MAIS, réexploitable pour imaginer des outils, qui sans être aussi minimalistes, représentent une forme de simplification. Couplage avec l'idée d'**efficience** : maximiser le service rendu en minimisant l'entrée d'énergie / d'effort.

Alexandre :

axe clé -> celui de la convivialité -> garder une dose d'accessibilité

Thomas :

rebond == présumé identifié dans la remarque d'Alexandre -> la question de l'accessibilité n'est pas détachable en fait du degré d'acculturation aux outils numériques de la population -> si tout le monde apprend les rudiments, l'accessibilité s'accroît (cf. le parallèle avec la lecture, qui n'est pas non plus un savoir faire "intuitif").

Sur la question des matériels à fabrication et maintenance non maîtrisés : où et quand trouver des alternatives ? Poser la question préalable de la durabilité et de la maintenabilité du matériel disponible pour

nos solutions.

Judicael: Plutôt que poser la question : "quels sont les bons usages ?", la question serait : "comment proposer des solutions qui permettent d'utiliser ces outils dans son contexte ?"

Faire des outils petits qui fonctionnent bien et qui sont combinables. Quel OS mettre à disposition d'un utilisateur des outils pour qu'il construise son outil ?

Monde adapté aux outils numériques, comment faire d'autres choix ?

Thomas :

sur les usages communicationnels du numérique, poser quand même la question de l'énorme empreinte fantôme → infrastructure matérielle, structure du système de production

Création du mind map des impacts.

10.3. Échange du 24/05

Participants :

- Judicaël
- Thomas

Judicaël (pour patienter)

Thème de la semaine : évaluation

- s'inspirer de l'ACV ?
- cycle de vie : fabrication, distribution, utilisation, fin de vie
- inputs / outputs
 - sub-set des 7 péchés capitaux du greenwashing :
- affirmations non étayées
- coûts cachés
- question plus générale : geler notre définition du projet OSlow et assumer de poser/investiguer/documenter les points aveugles et limites de cette idée ?

Thomas :

Une limite du parcours lownum ? L'hypothèse de départ est non low-tech : écarte l'idée de remettre radicalement en question le bien-fondé de la proposition de projets techniques.

Ce qui serait intéressant : pointer les critères pour admettre qu'un OS est un OS low-tech ? Et sur cette base-là : comment ça permet aux usagers de définir leurs propres outils.

⇒ d'abord une grille d'analyse documentée, et ensuite seulement voir si une proposition technique est fondée.

Low-tech : ne pas prescrire un système à un utilisateur dont on ne connaît rien.

Judicael :

proposer une grille de compatibilité avec la convivialité. Voir comment accompagner la réflexion sur cet axe.

Thomas

Lecture rapide du rapport de GreenIT sur l'impact environnemental du numérique. Citation sur l'évaluation énergétique des logiciels : bon indicateur d'efficacité → l'énergie dépensée pour **une tâche/un service** rendu = un indicateur synthétique des impacts → taille mémoire, efficacité des traitements, usage du réseau

Une piste corroborée par d'autres sources :

- benchmark langages

- étude conso énergétique des data centers

Autre aspect : mesurer l'énergie → mesure implicite de la complexité du programme // mesure de l'obsolescence induite par la non-frugalité.

Idée pratique :

- système de mesure pour comparaison
- conso au repos
- conso avec les différentes distros comparées

Récap' possible également à partir de ces données sur la marge d'usage laissée disponible par les différentes distro pour les ressources hardware (CPU, RAM)

Judicaël :

Quel serait le cas "témoin" dans ce benchmark ?

Thomas :

Outil Intel pour mesure des utilisations des composants hardware → données accessibles au noyau Linux
Phase de calibration Post-calibration : lancement du programme sur un scénario de référence pour mesure de la quantité d'énergie dépensée.

En pratique : installer cet outil sur les différentes distributions avec le paramétrage par défaut

Test réalisé : comparaison Firefox / browser minimal avec interpréteur HTML5 [https://mastodon-social/@teajay/108353483774151130](https://mastodon.social/@teajay/108353483774151130)

- pic firefox : crête à 5 W, conso = 108 J
- pic browser minimal : crête à 445 mW, conso = 0.07 J

Autres dimensions à évaluer : celles relatives à la convivialité

- cf question du midi : comment amener les utilisateurs à s'éduquer à des outils conviviaux, c'est-à-dire non pas intuitifs mais appropriables ?
- critères
- disponibilité d'infos : conduit à écarter les solutions propriétaires ?
- quid des solutions propriétaires mais documentées ?
- test proposé : capacité d'adapter l'OS à un device non prévu par l'éditeur ?
- quelles pratiques de collectes imposées des données personnelles ? => dans quelle mesure l'outil impose à l'utilisateur des pratiques indépendantes de (voire contraires à) sa volonté ? #mise en dépendance et pratiques des BigTech <https://replay.jres.org/w/kCEEhbvGS6Dzim2ibXnuJo>
- sur la question de la non-capture de l'utilisateur : comment sont associés entre eux les composants de l'OS pour permettre de sélectionner le sous-ensemble strictement pertinent pour mon usage/besoin ?
- coût cognitif d'assemblage néanmoins potentiellement élevé
- à étayer : les usages courants → consultation de pages, visionnage vidéo/audio, et part RÉSIDUELLE pour la création de contenus (logiciels, audiovisuels) <https://labo.societenumérique.gouv.fr/2020/03/03/les-français-et-le-numérique-quels-usages-courants/>
- retour sur coût énergétique des usages : trouver pour une page web donnée des indicateurs de fréquentation => y transposer les données du benchmark pour mettre à l'échelle les impacts énergétiques
- sur la frontière OS / user-space software : démarche similaire possible, mais maintenir la clarté de la distinction entre les deux

Objectif d'une approche frugale du software : comment ça nous aide à ne plus produire de nouveaux devices ? En termes d'ACV : tous les coûts déjà investis pour créer du matériel sont hors de portée et hérités → en revanche : l'ACV redevient pertinente pour évaluer plutôt les usages (la fabrication : hors-scope par définition ?) => point de vigilance : besoin d'étayer la possibilité de substituer du nouvel hardware par

du software frugal permettant effectivement le réemploi de hardware réputé obsolète + sur les coûts de développement vs usages : se trouver des indicateurs prouvant que les échelles sont vraiment inégales (os libres : puiser dans les métriques des outils de code versionning -> fréquence de commit, nombre de lignes + études existantes sur la valeur économique des ressources engagées dans la création de logiciels libres : voir <https://www.openhub.net/p/boost>, https://www.openhub.net/p/fuchsia-os/estimated_cost, https://www.openhub.net/p/linux/estimated_cost)