

FORTH: Un langage pour la frugalité ?

L. Chaloyard, M.P. Escudié, L. Morel, G. Salagnac, N. Stouls

Équipe Phénix, laboratoire CITI, INSA de Lyon

Contexte : l'urgence de la transition écologique

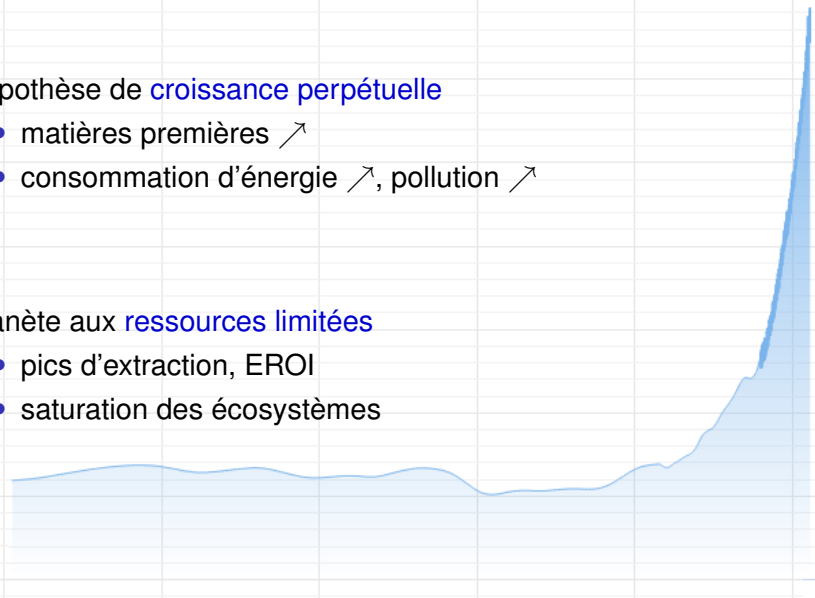
Hypothèse de **croissance perpétuelle**

- matières premières ↗
- consommation d'énergie ↗, pollution ↗

vs

Planète aux **ressources limitées**

- pics d'extraction, EROI
- saturation des écosystèmes



Positionnement

Comment préserver nos conditions de vie à moyenne échéance ?

Développement durable

- améliorer l'efficacité énergétique
- pivoter vers la croissance verte



et/ou



Frugalité

- viser la réduction des besoins
- jusqu'à atteindre l'équilibre

Mais que faire du numérique ?

Faut-il tout arrêter ?

Pourquoi et comment démanteler le numérique?

Romain COUILLET¹, Grégoire POISSONNIER²

¹ Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, 38000 Grenoble, France.

² Chercheur transdisciplinaire indépendant, Grenoble, France.
romain.couillet@univ-grenoble-alpes.fr

Résumé – Prenant appui sur cinquante ans de travaux en sciences sociales, sur l'accumulation des connaissances relatives à l'anéantissement biologique mondial ainsi que sur les analyses révélant l'illusion techno-solutionniste, cet article vise à démontrer l'urgence impérieuse d'un démantèlement effectif du monde numérique. Une grille d'évaluation et des pistes d'engagement individuel et collectif sont développées.

Abstract – Buttressed on fifty years of social science evidence, on the accumulated knowledge and analyses of the ongoing global biological collapse and on the techno-solutionism delusion, this article demonstrates the urgency of an effective dismantlement of the digital world. An evaluation grid and directions of personal and collective engagement measures are developed.

1 Le numérique à l'heure de l'anéantissement¹ biologique global

1.1 État des lieux planétaire

Le secrétaire général de l'ONU annonce en septembre 2018 qu'une "menace existentielle directe" pèse sur l'espèce humaine, potentiellement dès l'aube du 22^e siècle [1]. Guterres se fait ici le relais du nouveau consensus scientifique [2] : les réserves de pétrole à ce jour extractibles induiraient, dans un scénario d'émissions considérées par le GIEC comme "modérées", un réchauffement global de +2,9°C d'ici à 2100 (+3,8°C sur le territoire français) [3]. Dans ce scénario, en France métrapolitaine, les conditions de température et d'humidité deviennent invivables pour l'être humain plusieurs jours par an.²

Le réchauffement climatique n'est cependant pas le plus grave des dépassements planétaires (identifiés parmi 9 limites de stabilité du système-Terre). L'effondrement de la biodiversité, consécutive à la "première extermination de masse"³ [4] (82% de la biomasse sauvage mondiale s'est éteinte depuis 1970), l'introduction de nouvelles substances (l'océan de plastique couvre aujourd'hui 3× le territoire de la France) et la perturbation des cycles de l'azote et du phosphore (les engrais chimiques ont doublé la quantité de nitrates mondiales naturelles, eutrophisant ainsi les cours d'eau douce)⁴ ont atteint des points de non-retour autrement plus inquiétants.⁴

Cet emballage est la propriété de l'éditeur.

et d'artificialisation des milieux de la société de production-consummation occidentale. À titre d'exemple, un terrain de football de forêt disparaît toutes les deux secondes dans le monde (essentiellement dû à la consommation de viande de l'Occident); l'extraction fossile induit la production annuelle de 40 Gtons de CO₂ (chiffre en perpétuelle croissance); il faut aujourd'hui creuser, concasser et filtrer à l'arsenic et au cyanure une tonne de roches pour extraire un gramme d'or alors qu'aucun règlement mondial n'impose le traitement des déchets toxiques rejetés dans l'environnement; etc.

Dès lors, est-il trop tard pour éviter une sortie des dynamiques planétaires en vigueur pendant l'Holocène (de -10 000 à nos jours)? Oui, les dépassements ne sont plus réversibles, nous sommes entrés dans une nouvelle ère géologique très instable. Est-il possible d'éviter la disparition de l'humanité au 22^e siècle et avant cela d'assurer notre survie dans les décennies à venir? Les derniers rapports du GIEC suggèrent que oui, si une transformation sociétale majeure s'opère dès aujourd'hui. Contrairement à ce qui était estimé jusqu'alors, un arrêt hypothétique des émissions mondiales de gaz à effet de serre marquerait l'arrêt du réchauffement global et une lente baisse pourrait être observée en quelques décennies [6].

La "solution" à l'enjeu de la transition écologique est en réalité éminemment "simple" et visiblement exécutable en col

Mais que fa

Pourquoi et comment démanteler le numérique?

HÉRITAGE ET FERMETURE

UNE
DU
ÉCOLOGIE
DÉMANTÈLEMENT

Nous dépendons pour notre subsistance d'un « monde organisé », tramé par l'industrie et le management. Ce monde menace aujourd'hui de s'effondrer. Alors que les mouvements progressistes rêvent de monde commun, nous héritons contre notre gré de communs moins bucoliques, « négatifs », à l'image des fleuves et sols contaminés, des industries polluantes, des chaînes logistiques ou encore des technologies numériques. Que faire de ce lourd héritage dont dépendent à court terme des milliards de personnes, alors qu'il les condamne à moyen terme ? Nous n'avons pas d'autre choix que d'apprendre, en urgence, à destaurer, fermer et réaffecter ce patrimoine. Et ce, sans liquider les enjeux de justice et de démocratie. Contre le front de modernisation et son anthropologie du projet, de l'ouverture et de l'innovation, il reste à inventer un art de la fermeture et du démantèlement: une (anti) écologie qui met « les mains dans le cambouis ».

EMMANUEL BONNET,
DIEGO LANDIVAR,
ALEXANDRE MONNIN

Résumé – Prenant appui sur la biologie mondiale, nous proposons un démantèlement effectif

Abstract – Buttressed by world biology, we propose an effective dismantling of the digital grid and its

1 Le numérique

1.1 État des lieux

Le secrétaire général de l'ONU a récemment déclaré qu'une "menace existentielle" se fait ici le relais de la crise des réserves de pétrole. Le scénario d'émissions "réchauffement", un réchauffement sur le territoire tertiaire, les villes tropicales, les villes viennent invader.

Le réchauffement grave des six dernières années, diversité, consommation [4] (82% de la population en 1970), l'introduction de la technique couvre la perturbation chimiques organiques, eutrophication, points de non-retour.

Cet emballement

Mais que fa

Pourquoi et comment démanteler le numérique?

HÉRITAGE
ET FERMETURE
UNE

Résumé – Prenant app
biologique mondial ain
démantèlement effectif

Certains usages paraissent pourtant utiles

- Base de données, par ex catalogue, réservations
- Communication personnelle, par ex e-mail
- Discussion de groupe, par ex forums
- Partage de connaissance, par ex Wikipedia
- ...

▶ pourrait-on encore faire tout ça dans un monde frugal ?

1970), l'intro
tique couvre
perturbation
chimiques o
relles, eutro
points de no
Cet emb

DIEGO LANDIVAR,
ALEXANDRE MONNIN

ent que
aujourd-
un arrêt
de serre
e baisse
est en

Vers une informatique frugale ?

Y a-t-il de la place pour le numérique **dans un monde frugal** ?
i.e. après «la transition», après «avoir atterri»

Hypothèses : oui, à condition de

- s'autoriser à repenser les usages
 - évaluer utilité/coût vs implémentations alternatives
- réduire la **puissance des machines**
 - fréquence CPU ↘ capacité mémoire ↘
- donc réduire la taille et la **complexité du logiciel**
 - bonus : appropriation/compréhensibilité/explicabilité ↗↗

Deux cas d'usage distincts

- Utilitaire : faciliter une tâche «du monde réel»
 - pas forcément besoin d'un ordinateur universel
- Logiciel : pouvoir «tout reprogrammer» sur la machine
 - «**Comment reconstruire du petit avec seulement du petit ?**»

Problématique : programmation + frugalité = ?

Quelles **briques logicielles** (OS, langage...) sont indispensables pour construire un système autosuffisant ?

- Peut-on identifier une notion d'«**autosuffisance logicielle**» ?

Quelles sont les «**bonnes propriétés**» d'un langage de programmation vis-à-vis de la frugalité ?

- Comment distinguer les «vrais progrès» au milieu de la surenchère technologique ?

Notre démarche

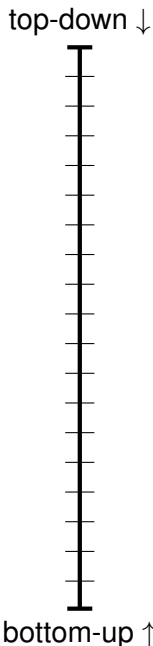
Approche soustractive

- partir d'un système existant (par ex Linux+GCC)
- identifier les briques superflues, les retirer
- recommencer aussi longtemps que possible

vs

Approche constructive

- repartir de zéro ou presque
- identifier les briques manquantes, les ajouter
- recommencer jusqu'à avoir un système utilisable



Notre démarche

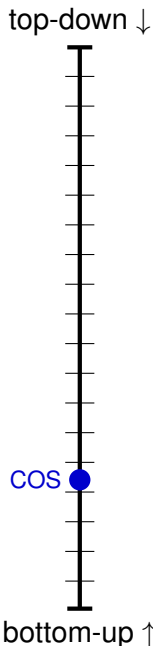
Approche soustractive

- partir d'un système existant (par ex Linux+GCC)
- identifier les briques superflues, les retirer
- recommencer aussi longtemps que possible

vs

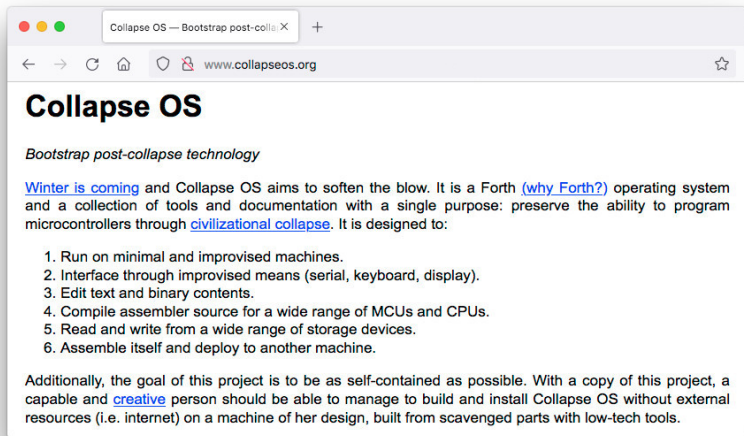
Approche constructive

- repartir de zéro ou presque
 - identifier les briques manquantes, les ajouter
 - recommencer jusqu'à avoir un système utilisable
- ▶ l'exposé d'aujourd'hui
- exploration de «CollapseOS» (et portage msp430)



The name of the game

développé depuis \approx 2021 par Virgil Dupras (Québec)



Rappel : nous ne sommes pas des survivalistes/collapsologues/etc
▶ simple étude de cas d'un logiciel frugal

Plan

Introduction

- Contexte et motivation

- Problématique et approche

FORTH : un exemple de langage frugal

- Historique

- Syntaxe et sémantique

- À retenir

Conclusions et perspectives

- Retour d'expérience

- Pistes de travaux futurs

Le langage FORTH

Historique

- créé par **Charles H. Moore** (NRAO) vers 1968
- (famille de) langage(s) très bas niveau mais portable(s)
- exécutif léger : interpréteur+compilateur+asm en qq Kio

Pas de grammaire

- seulement des **noms** séparés par des **espaces**
- tous caractères imprimables ok

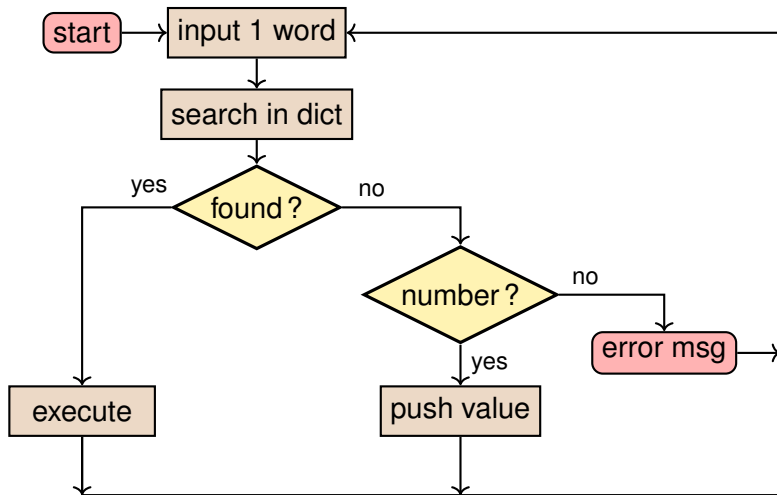
+	XOR	DUP	.	(."
---	-----	-----	---	---	----

 ...

Machine abstraite

- interpréteur **interactif** (Read-eval-print loop)
- «**le dictionnaire**» contient l'ensemble des **mots** connus
- **sémantique à pile** 2 3 4 + * → 14

Interpréteur top-level : un mot à la fois (aka outer interpreter)



Deux types de mots dans le dictionnaire

Mots primitifs

- implémentés en code machine

par ex `+`, `-`, `NOT`

Mots interprétés

- implémentés en termes d'autres mots forth

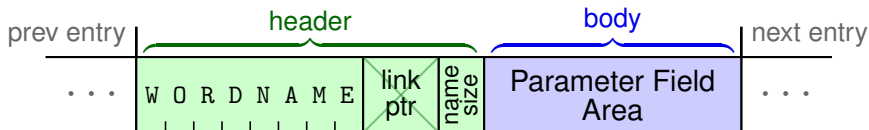
par ex `=` défini comme `-` puis `NOT`

- **définition** de nouveaux mots avec `:` et `;`

```
: = - NOT ;  
: SQUARE DUP * ;
```


Structure du dictionnaire

«Le dictionnaire» \triangleq une liste chaînée de mots



Exécuter un mot \triangleq sauter à la première adresse de son PFA

PFA d'un mot primitif \triangleq séquence d'instructions machine

```
32ca:    2b                                name=<+>
32cb:    c0 32 01                          prev=0x32c0 nsize=1
32ce:    <+>
32ce:    07 45                            mov  R5, R7
32d0:    35 41                            pop  R5
32d2:    05 57                            add  R7, R5
32d4:    30 40 1c 31                       br   #0x311c aka lblnext
```

Rq : registre R5 contient toujours le *Top-Of-Stack*

Structure d'un mot interprété

Exécuter un mot \triangleq sauter à la première adresse de son PFA

PFA d'un mot interprété \triangleq séquence de *execution tokens*

```
349a:    3d                                name=<=>
349b:    96 34 01                          prev=0x3496 nsize=1
349e: <=>
349e:    b0 12 32 31                       call #0x3132 aka lblxt
34a2:    dc 32                                -
34a4:    ec 32                                NOT
34a6:    50 31                                EXIT
```

Compiler un mot \triangleq émettre une séquence de tokens

execution token \triangleq pointeur sur un PFA

lblxt \triangleq adresse de l'«interpréteur de tokens»

- exécute successivement chacun des tokens du PFA
- *Return Stack* pour savoir où revenir lors du EXIT

Tous les mots compilés finissent par un *execution token* EXIT

```
3149:    45 58 49 54                name=<EXIT>
314d:    00 00 04                prev=0x0000 nsize=4
3150: <EXIT>
3150:    26 44                mov  0(R4), R6
3152:    24 83                decd R4
3154:    30 40 1c 31        br   #0x311c aka lblnext
```

registre R6 \triangleq *Virtual Machine Instruction Pointer*

registre R4 \triangleq *Return Stack Pointer*

Moteur d'exécution principal \triangleq DTC (*direct threaded code*)

lblnext:

```
311c:    67 46                mov  0(R6), R7
311e:    26 53                incd R6
3120:    00 47                br   R7
```

lblxt:

```
3132:    24 53                incd R4
3134:    84 46 00 00         mov  R6, 0(R4)
3138:    36 41                pop  R6
313a:    f0 3f                jmp  lblnext
```

registre R6 \triangleq *Virtual Machine Instruction Pointer*

registre R4 \triangleq *Return Stack Pointer*

jmp vers lblnext \neq call vers lblxt

Ajouter une entrée au dictionnaire

Définir de nouveaux mots avec `:` et `;` (eux-mêmes des mots)

```
: = - NOT ;  
: SQUARE DUP * ;
```

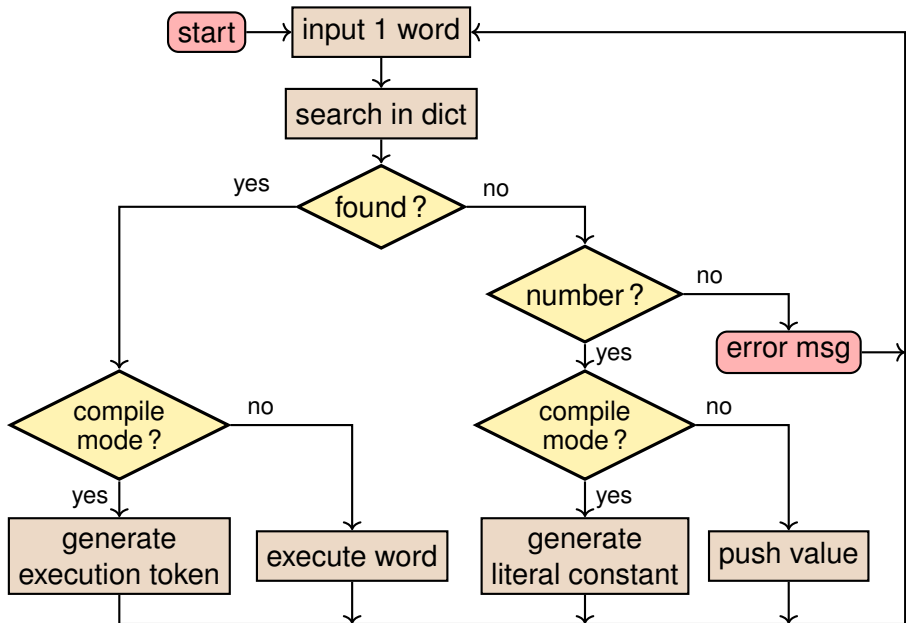
Intuition : le mot `:` entre en «mode compilation»

- 1) émet un en-tête avec nom, chaînage, préambule
 - le nom est lu (par `:` lui-même) sur l'entrée standard
- 2) les prochains mots lus seront *compilés* et non pas interprétés
 - compiler un nom $\hat{=}$ le chercher dans le dico puis émettre son XT à la suite du PFA de la *définition courante*

Intuition : le mot `;` repasse en «mode interprétation»

- émet le *execution token* de EXIT
- attention de ne pas oublier l'espace avant le point-virgule...

«*Interpretation semantics*» vs «*Compilation semantics*»



Exemple de définition : 1+ 1 + ;

```
350b:    31 2b                name=<1+>
350d:    00 35 02            prev=0x3500 nsize=2
3510: <1+>
3510:    b0 12 32 31        call #0x3132 aka lblxt
3514:    02 32 01            (b) value=0x01
3517:    ce 32                +
3519:    50 31                EXIT

31fc:    28 62 29            name=<(b)>
31ff:    ea 31 03            prev=0x31ea nsize=3
3202: <(b)>
3202:    05 12                push R5
3204:    65 46                mov.b 0(R6), R5
3206:    16 53                inc R6
3208:    30 40 1c 31        br #0x311c aka lblnext
```

Le token (b) occupe 3 octets : 2o de XT + 1o pour l'opérande immédiat

Certains mots sont spéciaux : les «mots immédiats»

Exemple

```
: MAX ( n n -- n ) 2DUP < IF SWAP THEN DROP ;
```

Certains mots ne respectent pas le «mode compilation»

- (début de commentaire
 - ignore l'entrée standard jusqu'à rencontrer une)
- f IF A THEN B n'exécute A que si f est non-nul
 - le mot-clé THEN est à lire comme un ENDIF
 - syntaxe complète f IF A ELSE A' THEN B
- ; sort du mode compilation
 - après avoir émis un execution token EXIT
- : implémente toute la compilation !

**Un «mot immédiat» sera toujours exécuté
(même en «mode compilation»)**

Exemple : IF-THEN

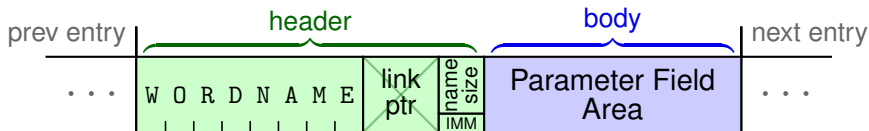
```
: MAX ( n n - n ) 2DUP < IF SWAP THEN DROP ;
```

```
4204:    4d 41 58          name=<MAX>
4207:    e4 41 03          prev=0x41e4 nsize=3
420a: <MAX>
420a:    b0 12 32 31      call #0x3132 aka lblxt
420e:    46 35             2DUP
4210:    fc 32             <
4212:    26 32 03          (?br) offset=+3 dest=0x4217
4215:    7a 33             SWAP
4217:    6c 33             DROP
4219:    50 31             EXIT
```

En mode compilation :

- **IF** émet un token (br?) avec offset factice
- **THEN** calcule et rectifie l'offset

Mots immédiats : implémentation



IMMEDIATE : Flag the latest defined word as immediate

```
: IMMEDIATE CURRENT 1- DUP C@ $80 OR SWAP C! ;
```

IMMEDIATE flag (1 bit) + *name size* (7 bits)

CURRENT $\hat{=}$ pointeur sur la dernière entrée du dictionnaire

HERE $\hat{=}$ pointeur sur la mémoire libre après le dictionnaire

Mots immédiats : exemples

Commentaires \triangleq entre parenthèses

```
: ( S" )" WAITW ; IMMEDIATE
```

```
: WAITW ( sa sl -- ) BEGIN 2DUP WORD S= UNTIL 2DROP ;
```

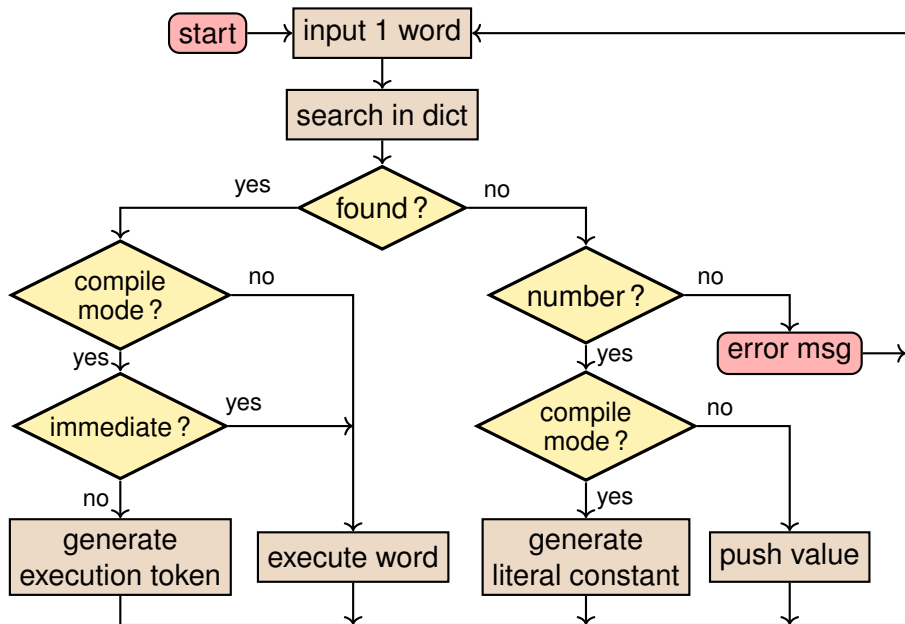
Flot de contrôle

```
: IF ( -- a / a: br cell addr )  
  COMPILE (?br) HERE 1 ALLOT ; IMMEDIATE
```

```
: THEN ( a -- / a: br cell addr )  
  DUP HERE SWAP - _bchk SWAP ( a-H a ) C! ; IMMEDIATE
```

```
: _bchk DUP $80 + $ff >  
  IF S" br ovfl" STYPE ABORT THEN ;
```

À retenir : Interpréteur+Compilateur



Si besoin : détail du «compilateur»

```
: : WORD TUCK MOVE, ( len )
CURRENT , C, | write prev value and size
HERE 'CURRENT ! | update CURRENT
[ lblxt LITN ] HERE CALLi! ALLOT
BEGIN
    WORD S" ;" S= IF
        COMPILE EXIT EXIT
    THEN
    CURWORD PARSE IF
        LITN
    ELSE
        CURWORD FIND IF
            DUP 1- C@ $80 AND ( imm? )
            IF EXECUTE ELSE , THEN
        ELSE
            (wnf)
        THEN
    THEN
AGAIN ;
```

En résumé

Forth

- (famille de) langages interprétés, sans grammaire
- frontière poreuse entre spécification et implémentation
- certains mots agissent directement sur l'état de l'interpréteur

COS

- un forth pour machine nue
- interpréteur + compilateur + éditeur \approx 1500 loc
- assembleur + drivers \approx 200 loc
- x-asm/x-comp : générer un binaire forth pour une autre ISA
 - z80, 8086, 6502, 6809
 - nous : portage vers msp430

Plan

Introduction

Contexte et motivation

Problématique et approche

FORTH : un exemple de langage frugal

Historique

Syntaxe et sémantique

À retenir

Conclusions et perspectives

Retour d'expérience

Pistes de travaux futurs

Problématique : programmation + frugalité = ?

Quelles **briques logicielles** (OS, langage...) sont indispensables pour construire un système autosuffisant ?

- Peut-on identifier une notion d'«**autosuffisance logicielle**» ?

Quelles sont les «**bonnes propriétés**» d'un langage de programmation vis-à-vis de la frugalité ?

- Comment distinguer les «vrais progrès» au milieu de la surenchère technologique ?

Briques matérielles

Quelles **briques logicielles** (OS, langage...) sont indispensables pour construire un système autosuffisant ?

- Processeur + mémoire
- Écran + clavier
- Stockage persistant

- Connexion réseau (?)

Briques logicielles

Quelles **briques logicielles** (OS, langage...) sont indispensables pour construire un système autosuffisant ?

Vraiment **indispensables** :

- générateur de code machine
 - ▶ y compris compilation croisée
- éditeur de texte
- shell interactif
- drivers

Quand même **bien pratiques** :

- émulateur
- débogueur

Bilan : difficultés rencontrées

Quelles sont les «**bonnes propriétés**» d'un langage de programmation vis-à-vis de la frugalité ?

Langage **sans typage** ni vraie **sémantique**

- ▶ programmation laborieuse, risque d'erreur

Exécution **machine nue** ▶ bug = reboot, mais sans dégâts

- ▶ difficultés de mise au point

Effort de **rétro-ingénierie** : 5+ kloc de python (+ émulateur msp430) pour savoir écrire 200 lignes de forth

- ▶ Développement croisé : linux, python, emacs, mspdebug...

Objectif initial \triangleq démo avec comm réseau, mais trop ambitieux

Bilan : leçons retirées

Quelles sont les «**bonnes propriétés**» d'un langage de programmation vis-à-vis de la frugalité ?

Circularité : langage X implémenté en X

- (et aussi en langage machine)

Programmation pour **machine nue**

- modèle de concurrence ? (nécessaire pour ISR)

Compilation ou interprétation ?

- facilité de programmation vs taille/complexité du logiciel
- messages d'erreur : compile-time ? runtime ? jamais ?

Perspectives de recherche

Continuer le travail de **portage + dev**

- driver flash (stockage persistant)
- communication «réseau»
- démonstrateur applicatif

Étudier des **forth plus sophistiqués** : typage, concurrence, debug, FFI...

Chercher d'autres **langages auto-suffisants** pour machine nue : BASIC ? C ? Lua ? Python ?

Mesurer le **bootstrap d'autres langages** (sur POSIX) pour comparer

Concevoir une **grille d'évaluation** des langages

Étudier les usages : que veut dire «**utiliser une petite machine**» ?

top-down ↓



bottom-up ↑

Merci de votre attention

Des questions ?

Transparents supplémentaires

Contexte : Taux de CO₂ atmosphérique

Contexte : les scénarios ADEME Transition(s) 2050

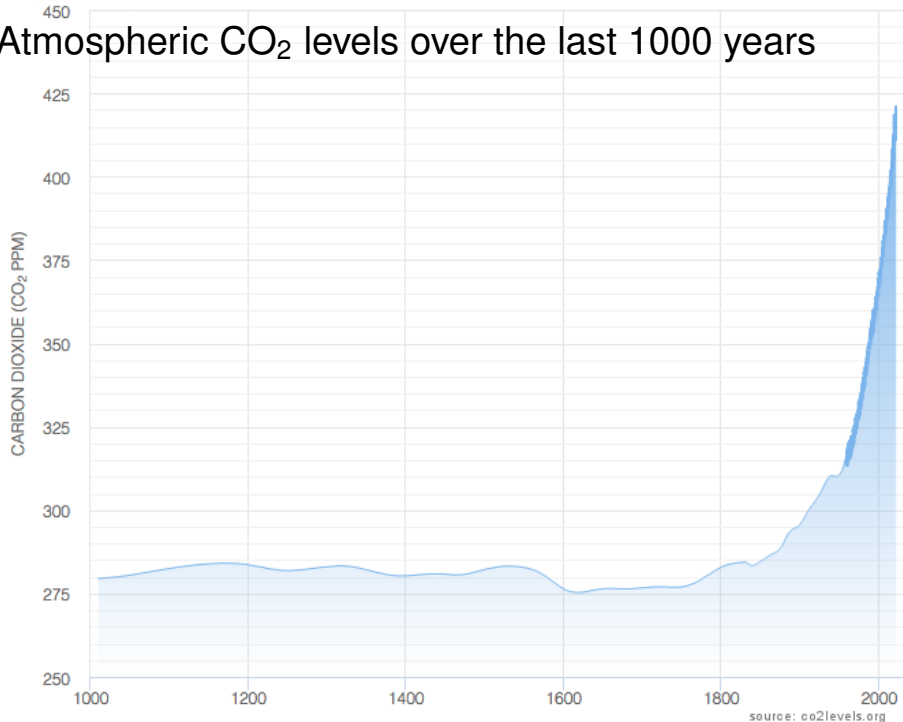
Forth : Principe d'une machine à pile

COS : Vue mémoire globale

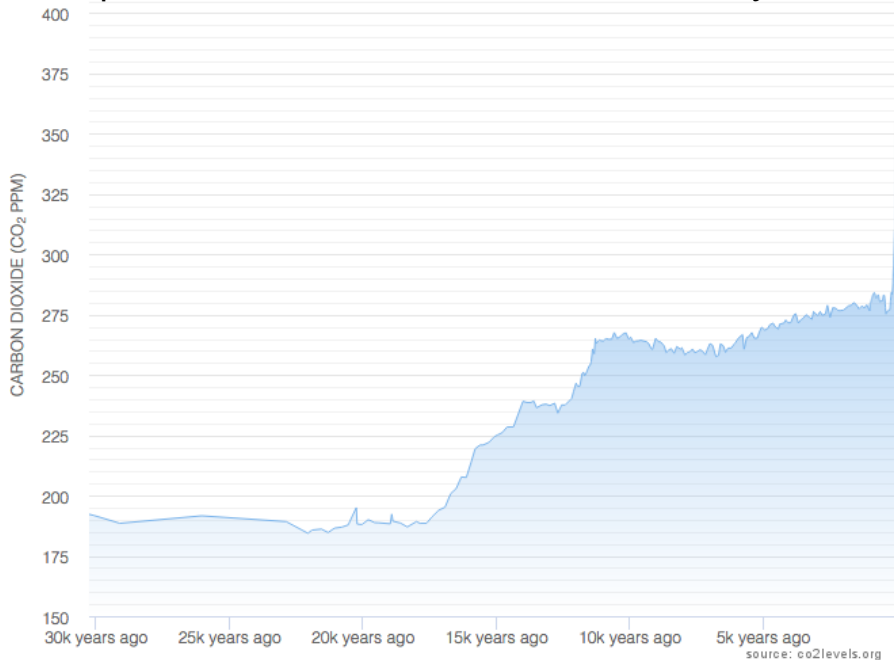
COS : compilation croisée

Perspectives de recherche

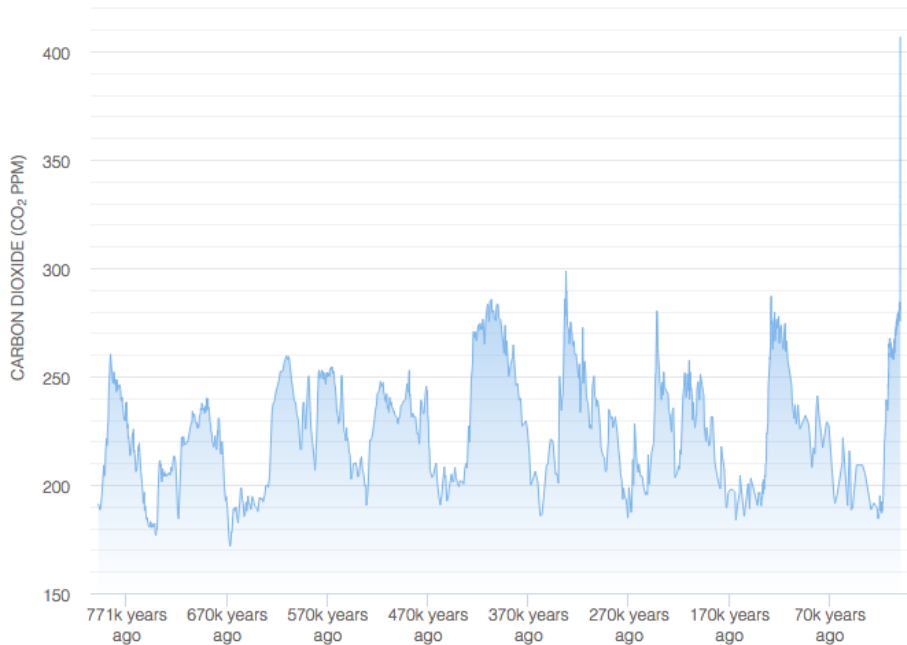
Atmospheric CO₂ levels over the last 1000 years



Atmospheric CO₂ levels over the last 30 000 years



Atmospheric CO₂ levels over the last 800 000 years



source: co2levels.org

ADEME – Rapport d'étude « Transition(s) 2050 »

S1 GÉNÉRATION FRUGALE

S2 COOPÉRATIONS TERRITORIALES

S3 TECHNOLOGIES VERTES

S4 PARI RÉPARATEUR

MODES DE VIE

ÉCONOMIE

Société

- Recherche de sens
- Frugalité choisie mais aussi contrainte
- Préférence pour le local
- Nature sanctuarisée



- Évolution soutenable des modes de vie
- Économie du partage
- Équité
- Préservation de la nature inscrite dans le droit

- Plus de nouvelles technologies que de sobriété
- Consommérisme « vert » au profit des populations solvables, société connectée
- Les services rendus par la nature sont optimisés

- Sauvegarde des modes de vie de consommation de masse
- La nature est une ressource à exploiter
- Confiance dans la capacité à réparer les dégâts causés aux écosystèmes

Alimentation

- Division par 3 de la consommation de viande
- Part du bio : 70 %



- Division par 2 de la consommation de viande
- Part du bio : 50 %



- Baisse de 30 % de la consommation de viande
- Part du bio : 30 %



- Consommation de viande quasi-stable (baisse de 10 %), complétée par des protéines de synthèse ou végétales



Habitat

- Rénovation massive et rapide
- Limitation forte de la construction neuve (transformation de logements vacants et résidences secondaires en résidences principales)

- Rénovation massive, évolutions graduelles mais profondes des modes de vie (cohabitation plus développée et adaptation de la taille des logements à celle des ménages)

- Déconstruction-reconstruction à grande échelle de logements
- Ensemble des logements rénovés mais de façon peu performante : la moitié seulement au niveau Bâtiment Basse Consommation (BBC)

- Maintien de la construction neuve
- La moitié des logements seulement est rénovée au niveau BBC
- Les équipements se multiplient, alliant innovations technologiques et efficacité énergétique

Mobilité des personnes

- Réduction forte de la mobilité
- Réduction d'un tiers des km parcourus par personne
- La moitié des trajets à pied ou à vélo



- Mobilité maîtrisée
- 17 % de km parcourus par personne
- Près de la moitié des trajets à pied ou à vélo



- Mobilités accompagnées par l'État pour les maîtriser : infrastructures, télé massif, covoiturage
- + 13 % de km parcourus par perso
- 30 % des trajets à pied ou à vélo



- Augmentation forte des mobilités
- + 28 % de km parcourus par personne
- Recherche de vitesse
- 20 % des trajets à pied ou à vélo



Technique

Rapport au progrès, numérique, R&D

- Innovation autant organisationnelle que technique
- Règne des low-tech, réutilisation et réparation
- Numerique collaboratif
- Consommation des data centers stable grâce à la stabilisation des flux

- Investissement massif (efficacité énergétique, EnR et infrastructures)
- Numerique au service du développement territorial
- Consommation des data centers stable grâce à la stabilisation des flux

- Ciblage sur les technologies les plus compétitives pour décarboner
- Numerique au service de l'optimisation
- Les data centers consomment 10 fois plus d'énergie qu'en 2020

- Innovations tout azimut
- Captage, stockage ou usage du carbone capté indispensable
- Internet des objets et intelligence artificielle omniprésents : les data centers consomment 15 fois plus d'énergie qu'en 2020



Gouvernance

Échelles de décision, coopération internationale

- Décision locale, faible coopération internationale
- Règlementation, interdiction et rationnement via des quotas

- Gouvernance partagée
- Fiscalité environnementale et redistribution
- Décisions nationales et coopération européenne



- Cadre de régulation minimale pour les acteurs privés
- État planificateur
- Fiscalité carbone ciblée

- Soutien de l'offre
- Coopération internationale forte et ciblée sur quelques filières clés
- Planification centralisée du système énergétique

Territoire

Rapport espaces ruraux – urbains, artificialisation

- Rôle important du territoire pour les ressources et l'action
- « Démétropolisation » en faveur des villes moyennes et des zones rurales

- Reconquête démographique des villes moyennes
- Coopération entre territoires
- Planification énergétique territoriale et politiques foncières

- Métropolisation, mise en concurrence des territoires, villes fonctionnelles



- Faible dimension territoriale, étalement urbain, agriculture intensive



Macro-économie

- Nouveaux indicateurs de prospérité (écarts de revenus, qualité de la vie...)
- Commerce international contracté



- Croissance qualitative, « réindustrialisation » de secteurs clés en lien avec territoires
- Commerce international régulé

- Croissance verte, innovation poussée par la technologie
- Spécialisation régionale
- Concurrence internationale et échanges mondialisés

- Croissance économique carbonée
- Fiscalité carbone minimaliste et ciblée
- Économie mondialisée

Industrie

- Production au plus près des besoins
- 70 % de l'acier, mais aussi de l'aluminium, du verre, du papier-carton et des plastiques viennent du recyclage



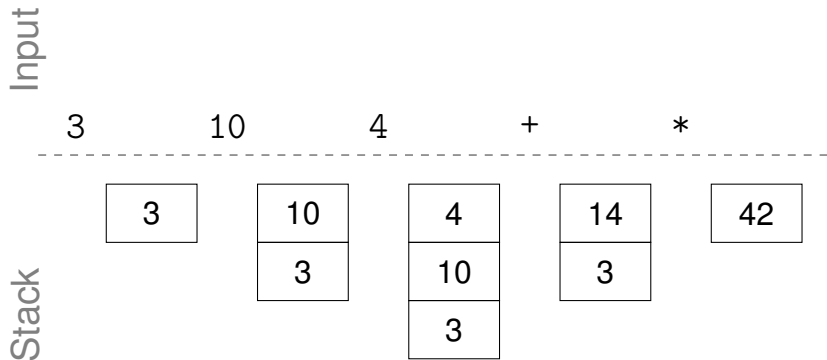
- Production en valeur plutôt qu'en volume
- Dynamisme des marchés locaux
- 80 % de l'acier, mais aussi de l'aluminium, du verre, du papier-carton et des plastiques viennent du recyclage

- Décarbonation de l'énergie
- 60 % de l'acier, mais aussi de l'aluminium, du verre, du papier-carton et des plastiques viennent du recyclage

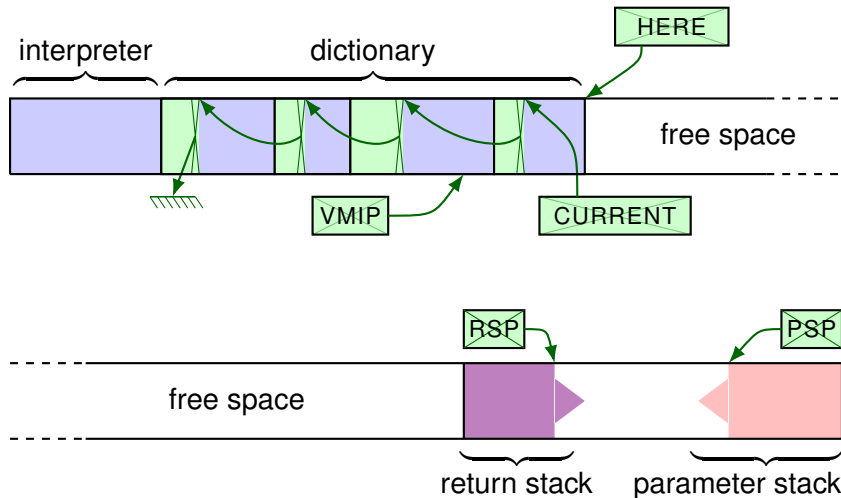


- Décarbonation de l'industrie reposant sur le captage et stockage géologique de CO₂
- 45 % de l'acier, mais aussi de l'aluminium, du verre, du papier-carton et des plastiques viennent du recyclage

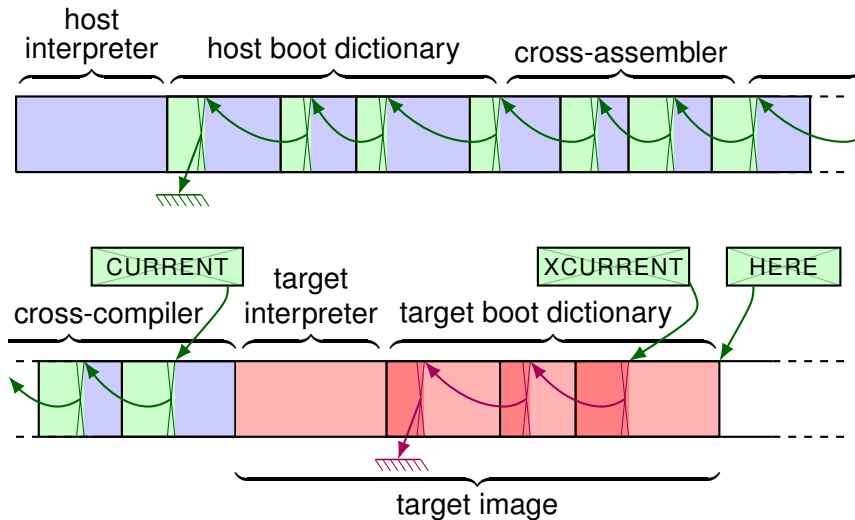
Machine à pile : principe



Vue mémoire globale



Compilation croisée



Portage COS vers msp430

Machine hôte

Assembleur MSP430

- \approx 60 lignes de forth
- \approx 1.2 kio : en-têtes + *execution tokens*

Image cible (\approx 5 kio)

Interpréteur : lblxt, lblnext...

- \approx 15 lignes de forth
- \approx 70 octets de code machine

Mots natifs (\approx 35) : `EXIT` `+` `@` `!` `(b)` `(n)` ...

- \approx 200 lignes de forth
- \approx 700 octets de binaire (en-têtes + code machine)

Mots interprétés (\approx 130) : `=` `1+` `FILL` `STYPE` `:` `IF` ...

- \approx 350 lignes de forth
- \approx 4kio de binaire (en-têtes + execution tokens)

Pilotes

- port série : \approx 30 lignes de forth, \approx 120 octets

WIP : Continuer travail autour de COS

Prétexte : un **démonstrateur texte+réseau** ala minitel

Objectif : Cerner le «bon» **modèle de programmation** pour du logiciel frugal

Auto-recompilation

- driver flash
- commande

LOAD

Communication «réseau» ?

- chiffrement
- concurrence

Étudier des forth plus sophistiqués

Travail de Lucas autour de DuskOS

- Forth + mini-C
- sales pitch : «lire un PDF»

Typage : par exemple via du *linting*?

Concurrence : main-vs-ISR, files FIFO ou mémoire partagée, threads ?

Debug : traçage d'exécution, points d'arrêt, introspection ?

Chercher d'autres langages

Existe-t-il d'autres langages visant explicitement la programmation machine nue et l'auto-suffisance logicielle ?

BASIC

- langage universel dans le grand public années 80/90
- facile à prendre en main ; quel modèle de programmation ?

C

- la *Lingua franca* d'aujourd'hui
- langage compilé, bas niveau

Python

- facile à apprendre, modèle d'exécution clair
- compatibilité avec la frugalité et l'auto-suffisance ?

WIP : quantifier le bootstrap de différents langages

Quelle «taille de machine» faut-il pour programmer en X ?

Combien de temps CPU et d'espace RAM consomme

- l'exécution d'un hello world ?
- la compilation d'un hello world par la toolchain ?
- la compilation de la toolchain par elle-même ?

Idée : comparer (en ordres de grandeur) différentes paires langage+toolchain : C, C++, Java, Python, Rust

Étudier les usages (1)

Que veut dire «utiliser une petite machine» ?

Exemples de **cas d'usage** :

- rédiger une lettre de motivation
- maintenir l'inventaire d'une collection (disques, livres)
- faire un bilan comptable

Exemples de **dates de référence** :

- 1975
- 1985
- 1995
- 2005
- 2015

Idée : Pour chaque paire usage+date, recréer (en émulation) les conditions d'époque et mesurer la conso CPU+RAM

Étudier les usages (2)

De quoi avons-nous vraiment «besoin» ? et/ou «envie» ?

Idée : mener une enquête (sociologique ?) auprès du grand public avec un questionnaire :

- Dans un monde frugal, quels usages du numérique actuel mériteraient d'avoir disparu, selon vous ?
- Dans un monde frugal, quels usages du numérique actuel mériteraient d'être préservés, selon vous ?
- Dans un monde frugal, quels nouveaux usages du numérique souhaiteriez-vous voir advenir ?
- (autres idées de questions bienvenues)

Objectif : alimenter/orienter notre vision d'un **avenir frugal désirable**

Vers une grille d'évaluation des langages machine nue

En quoi un langage est-il frugal, ou ne l'est pas ?

Aspects techniques :

- Auto-suffisance logicielle
 - dépendance à un OS
- Portabilité
- Poids de l'exécutif
- Modèle de concurrence
- Entrées-sorties
- Typage

Aspects humains :

- Acceptabilité
- Utilisabilité
- Maintenabilité
- Explicabilité
- Justice sociale
- Usages