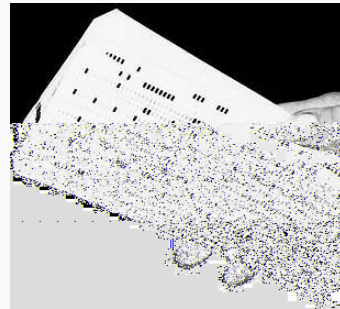


Les fonctions du langage

- ♦ l'outil d'écriture des programmes
 - ▶ par des hommes ou des programmes
 - ▶ pour la machine ou les autres hommes
 - ▶ à différents niveaux d'abstraction
- ♦ le support d'interaction avec la pensée
 - ▶ les langages induisent des styles distincts: impératif, fonctionnel, logique, temporel, etc...
- ♦ le support de guerres de religions
 - ▶ rares sont ceux qui aiment deux styles
 - ▶ et ceux qui font abstraction du NIH (Not Invented Here)

Le langage machine

- ♦ langage de plus bas niveau (initialement le seul)
- ♦ programmes directement exécutables par la machine
 - ▶ programmation fastidieuse
 - ▶ programmes non portables



Le langage machine

- ♦ langage de plus bas niveau (initialement le seul)
- ♦ programmes directement exécutables par la machine
 - ▶ programmation fastidieuse
 - ▶ programmes non portables

Le calcul de la factorielle, en code machine Intel

```

10111000 00000001 00000000 00000000 00000000
10111010 00000010 00000000 00000000 00000000
00111001 11011010
01111111 00000110
00001111 10101111 11000010
01000010
11101011 11110110
11000011

```

Le langage machine

- ♦ langage de plus bas niveau (initialement le seul)
- ♦ programmes directement exécutables par la machine
 - ▶ programmation fastidieuse
 - ▶ programmes non portables

Instruction Set Architecture



Synergistic Processor Unit

Multiply

Required v 1.0

mpy rt,ra,rb

0 1 1 1 0 0 0 1 0 0

RB

RA

RT



Le langage assembleur : représentation textuelle du langage machine

- ♦ utilisation de noms pour les instructions, les registres, etc
- ♦ utilisation de noms pour désigner les points dans le programme, les cases de la mémoire, etc
- ♦ possibilité de mettre des commentaires sur le code

Le calcul de la factorielle, en langage assembleur Intel

```
;; Calcul de la fonction factorielle
;; En entree: l'argument N est dans le registre EBX
;; En sortie: la factorielle de N est dans le registre EAX
```

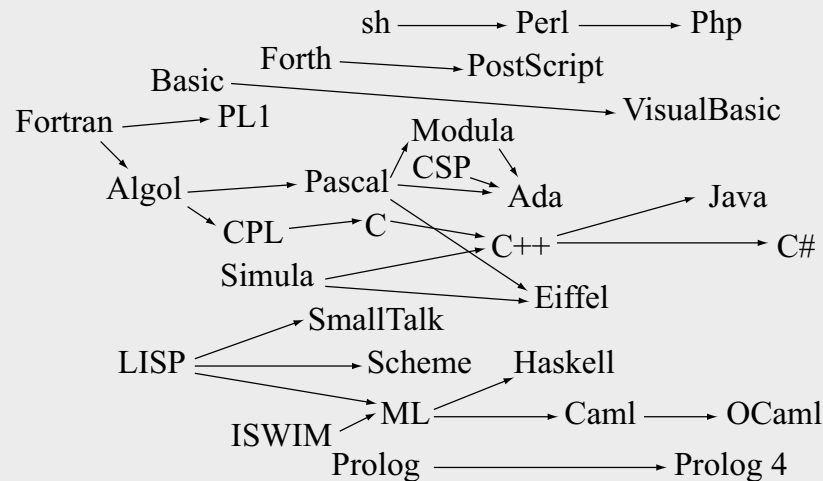
Factorielle:

```
    mov eax, 1      ;; resultat initial = 1
    mov edx, 2      ;; indice de boucle = 2
L1:  cmp edx, ebx    ;; tant que indice <= N ...
     jg L2
     imul eax, edx   ;; multiplier le resultat par l'indice
     inc edx        ;; incrementer l'indice
     jmp L1         ;; fin tant que
L2:  ret            ;; fin de la fonction Factorielle
```

Le langage assembleur : représentation textuelle du langage machine

- ♦ utilisation de noms pour les instructions, les registres, etc
- ♦ utilisation de noms

50 60 70 80 90 00 10



http://www.oreilly.com/news/graphics/prog_lang_poster.pdf

<http://www.digibarn.com/collections/posters/tongues/>

Les composants d'un langage

- ♦ des principes de calcul
 - ▶ impératif, fonctionnel, logique, temporel, etc
- ♦ des principes d'architecture
 - ▶ modularité, héritage, polymorphisme, etc
- ♦ une syntaxe et un style syntaxique
 - ▶ détermine les programmes bien construits
- ♦ un système de types
 - ▶ évite d'ajouter des choux et des carottes
- ♦ une sémantique plus ou moins formelle
 - ▶ définit le sens des programmes
- ♦ un outillage
 - ▶ compilateurs, débogueurs, manuels, exemples, bibliothèques, interfaces avec d'autres langages
- ♦ une communauté d'utilisateurs

Pourquoi tant de langages ?

- ♦ beaucoup d'aspects à traiter ensemble
 - ▶ données, actions = programming in the small
 - ▶ architecture = programming in the large
- ♦ beaucoup d'innovations successives
 - ▶ fonctionnel, polymorphisme, modules, objets, parallélisme, etc
- ♦ énormément de compromis possibles
 - ▶ plusieurs grandes classes et beaucoup de dialectes

Raisons du succès (dans le temps)

- | | |
|--|---|
| 1 S'adresser à un large public | C (associé à la popularité d'Unix) |
| 2 Faire le job | Cobol (conçu pour l'écriture de rapports) |
| 3 Offrir une nouvelle fonctionnalité | Java (écrit une fois, exécuté partout) |
| 4 Combler un créneau | Mathematica (approprié aux calculs complexes) |
| 5 Offrir un minimum d'élégance | Icon (doté d'une syntaxe familière) |
| 6 Fédérer autour d'un leader charismatique | Perl (élaboré par Larry Wall) |

La programmation impérative

- ♦ basé sur l'effet de bord permettant de remplacer en mémoire une valeur par une autre (opération appelée affectation)
- ♦ utilisation de noms symboliques pour les adresses
- ♦ un programme est une suite de traitements exécutés séquentiellement
- ♦ possibilité de
 - ▶ réaliser des tests
 - ▶ itérer des traitements
 - ▶ modulariser le traitement (sous-programmes, procédures, fonctions)

Définition itérative de factorielle (en C)

```

int fact (int n) {
    // factorielle de n
    int r = 1, i;
    for (i = 2; i <= n; i++)
        r = r * i;
    return r;
}
  
```

Longueur & hauteur dans le problème de Collatz (en Java) »
