

Les booléens, langage machine et système d'exploitation

Les booléens (chap 6)

Les tables de vérité:

P → bit entrée
Q → bit sortie

→ ou "-" ou "non" → négation

Porte Not → $Q = \neg P$ c'est l'inverse de bit entrée

$P \rightarrow 1 \Rightarrow Q \rightarrow 0$	} inversement de la valeur
$\rightarrow 0 \Rightarrow \rightarrow 1$	

→ ou "et" ou "&" → conjonction

P et Q → bits entrée
R → résultat

Porte AND → $R = P \& Q$ c'est une addition mais si deux 1 présent

$P \& Q \Rightarrow R$

0 0 0

0 1 0

1 0 0

1 1 → 1 Vrai seulement avec deux 1 en entrée

→ ou "ou" ou "+" → disjonction

Porte OR → $R = P + Q$ c'est vrai dès qu'un 1 est présent

$P + Q \Rightarrow R$

0 0 0

0 1 → 1

1 0 → 1

1 1 → 1

} Vrai car un 1 est présent

Les lois de Morgan:

$$\neg(x \mid y) = \neg x \& \neg y$$

$$\neg(x \& y) = \neg x \mid \neg y$$

Porte NAND $\rightarrow R = P \uparrow Q$ c'est vrai dès qu'un 0 est présent

$$P \uparrow Q \Rightarrow R$$

$$0 \quad 0 \rightarrow 1$$

$$0 \quad 1 \rightarrow 1$$

$$1 \quad 0 \rightarrow 1$$

$$1 \quad 1 \rightarrow 0$$

Vrai car un 0 est présent

opposé de AND $R \rightarrow 1 \rightarrow 0$
inverse de OR $1+0 \rightarrow 1$

Porte NOR $\rightarrow R = P \downarrow Q$ c'est une addition mais si deux 0

$$P \downarrow Q \Rightarrow R$$

$$0 \quad 0 \rightarrow 1$$

$$0 \quad 1 \rightarrow 0$$

$$1 \quad 0 \rightarrow 0$$

$$1 \quad 1 \rightarrow 0$$

Vrai seulement avec deux 0

opposé de OR $R \rightarrow 1 \rightarrow 0$
inverse de AND $1+1 \rightarrow 0$

beaucoup

Porte XOR $\rightarrow R = P \oplus Q$ c'est vrai quand les bits sont différents

$$P \oplus Q \Rightarrow R$$

$$0 \quad 0 \rightarrow 0$$

$$0 \quad 1 \rightarrow 1$$

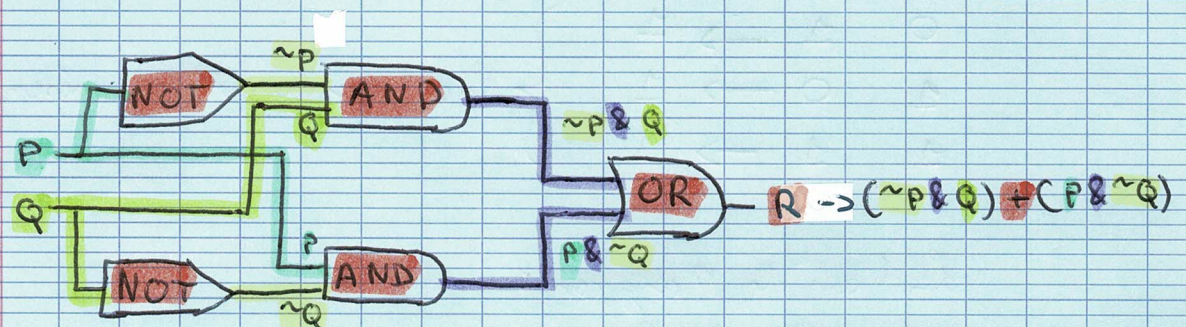
$$1 \quad 0 \rightarrow 1$$

$$1 \quad 1 \rightarrow 0$$

Vrai car bits différents

le OR rem le double 1 $1+1=0$

On peut combiner les portes



Les booléens, langage machine et système d'exploitation

Langage machine (chap 7)

Histoire :

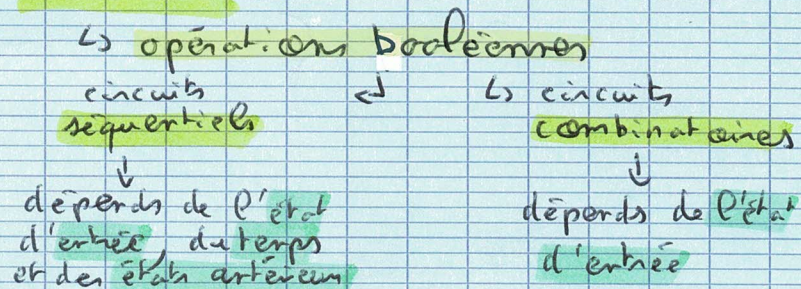
Transistor → 1947 par John Bardeen, William Shockley et Walter Brattain.

Le Colossus → 1943 - "ordinateur" avec des tubes à vide

Transistors sont regroupés dans des circuits intégrés
gravés sur le silicium

Les circuits logiques :

L'élément de base → Transistors



Les composants :

La mémoire vive (RAM - Random Access Memory)

Elle peut lire et écrire des données dans un octet

↳ forme la mémoire séquentiel

La mémoire morte (ROM - Read-Only Memory)

On ne peut pas modifier l'utilisation car elle ne peut pas écrire, seulement lire.

Elle stocke les firmwares ou le BIOS.
des données permanentes

Le microprocesseur (CPU - Central Processing Unit)

- 3 parties {
- L'unité arithmétique (ALU ou VAL)
 - ↳ chargée de l'exécution des calculs
 - L'unité de commande
 - ↳ exécute les instructions (programmes)
 - Les registres
 - ↳ mémorisent l'information transitoire

Les bus → pour relier les parties d'un ordinateur

↳ bus d'adresse → fait circuler des adresses

↳ bus de données → fait circuler des données

↳ bus de contrôle → spécifie le type d'action (écriture, lecture)

Le langage machine :

- 2 parties {
- "code opérande" → indique la nature des données
 - ↳ valeur immédiate
 - ↳ un registre du CPU
 - ↳ donnée située dans la RAM
 - "code opération" → donne au processeur le type de traitement à réaliser.

Le programme exécutable se situe dans la RAM

Registre particulière → IP (Instruction Pointer) → adresse de la prochaine instruction à exécuter

Les booléens, langage machine et système d'exploitation

IR (Instruction Register) → conversion physique des bits du processeur
→ activation du circuit de réalisation de l'opération

Fonctionnement cyclique du CPU:

- Copie le registre IR dans la RAM à l'adresse de l'IP
- Décode l'instruction IR → activation du circuit de l'opération
- Exécute l'instruction → met à jour l'IP pour continuer dans le programme

Modèle de Von Neuman:

- Une seule mémoire vive dans le programme et les données
 - Copie de la mémoire RAM dans IR pour une instruction
- Composé de →
- l'unité logique ou arithmétique (UAL) ①
 - l'unité de contrôle ②
 - la mémoire ③
 - dispositifs d'entrée/sortie ④

- ① Coordonne l'exécution des instructions
- ② Effectue les opérations
- ③ Stocke les instructions du programme et les données à traiter
- ④ Permet la communication

Les bockéens, langage machine et système d'exploitation

/home/username → pour l'utilisateur username
/root/ → pour l'administrateur du système (le root)

B - Naviguer :

ls → liste les fichiers et répertoires du dossier courant

ls /chemin/vers/dossier → liste les fichiers et répertoires du dossier donné en argument

cd /root → se déplacer pour changer de dossier

chemin absolu → commence par un /

↳ chemin complet à partir de la racine

chemin relatif → par de slash (ou . /), on part du répertoire courant :

→ cd /Munic ou ./Munic

→ cd .. → remonte d'un dossier

→ cd ../Documents → remonte dans le dossier Documents

C - Les droits :

l'option -l pour long précis dans sa sortie :

ls -l Documents/git/flask-csv/ → ligne de commande pour voir flask-csv

d a r x r w x 1 ben docker 2690 4 juil. 10:39 brouillon..py nb

↓
les droits :
d → si c'est un dossier
r → droit de lecture
w → droit d'écriture
x → droit d'exécution

le propriétaire
↓
taille

date et heure

non
↑
suffisant dossier

3 types d'octe pour les droits:

→ 1. l'utilisateur

→ 2. le groupe

→ 3. les autres

-rw-r--r-- => - c'est un fichier (pas de d)

- l'utilisateur peut lire et écrire (rw-)

- le groupe peut lire (r--)

- les autres peuvent lire (r--)

D - Gérer les droits:

Pour un fichier: `chmod [ugoa][+-=][rwx]* nom-du-fichier`

↓ ↓ ↓
commande types d'octe options les droits

Pour un répertoire: `chmod -R` (pareil pour la suite)

directement depuis la valeur octale:

`chmod 755 mon_dossier` → pour donner aux 3 octets les droits directement

voir tableau
octale fichier

E - Gérer le propriétaire:

`chown` → pour changer le propriétaire d'un fichier ou
↳ `chown [-hLPR] [utilisateur] [:groupe] cible [cible2]` dossier.

Autres commandes:

`mkdir` → créer un répertoire dans le dossier courant

`*` → permet de manipuler plusieurs fichiers en même temps

↳ `*.py` → tous les fichiers .py du répertoire courant

↳ `*/*.txt` → tous les fichiers *.txt dans le sous-répertoire