

Éléments d'informatique – Cours 1. Éléments d'architecture des ordinateurs, mini-assembleur

Pierre Boudes

15 septembre 2010



This work is licensed under the *Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License*.

Survol du contenu du cours (ce semestre)

Architecture de von Neumann

Représentation des informations

Cycle d'exécution

Instructions

Trace d'exécution

Démos et fin

Survol du contenu du cours (ce semestre)

Survol du contenu du cours (ce semestre)

- Éléments d'architecture des ordinateurs (+mini-assembleur)

Survol du contenu du cours (ce semestre)

- Éléments d'architecture des ordinateurs (+mini-assembleur)
- Éléments de systèmes d'exploitation

Survol du contenu du cours (ce semestre)

- Éléments d'architecture des ordinateurs (+mini-assembleur)
- Éléments de systèmes d'exploitation
- Programmation structurée impérative (éléments de langage C)

Survol du contenu du cours (ce semestre)

- Éléments d'architecture des ordinateurs (+mini-assembleur)
- Éléments de systèmes d'exploitation
- Programmation structurée impérative (éléments de langage C)
 - Structure d'un programme C
 - Variables : déclaration (et initialisation), affectation
 - Évaluation d'expressions

Survol du contenu du cours (ce semestre)

- Éléments d'architecture des ordinateurs (+mini-assembleur)
- Éléments de systèmes d'exploitation
- Programmation structurée impérative (éléments de langage C)
 - Structure d'un programme C
 - Variables : déclaration (et initialisation), affectation
 - Évaluation d'expressions
 - Instructions de contrôle : if, for, while

Survol du contenu du cours (ce semestre)

- Éléments d'architecture des ordinateurs (+mini-assembleur)
- Éléments de systèmes d'exploitation
- Programmation structurée impérative (éléments de langage C)
 - Structure d'un programme C
 - Variables : déclaration (et initialisation), affectation
 - Évaluation d'expressions
 - Instructions de contrôle : if, for, while
 - Types de données : entiers, caractères, réels, tableaux, enregistrements

Survol du contenu du cours (ce semestre)

- Éléments d'architecture des ordinateurs (+mini-assembleur)
- Éléments de systèmes d'exploitation
- Programmation structurée impérative (éléments de langage C)
 - Structure d'un programme C
 - Variables : déclaration (et initialisation), affectation
 - Évaluation d'expressions
 - Instructions de contrôle : if, for, while
 - Types de données : entiers, caractères, réels, tableaux, enregistrements
 - Fonctions d'entrées/sorties (scanf/printf)

Survol du contenu du cours (ce semestre)

- Éléments d'architecture des ordinateurs (+mini-assembleur)
- Éléments de systèmes d'exploitation
- Programmation structurée impérative (éléments de langage C)
 - Structure d'un programme C
 - Variables : déclaration (et initialisation), affectation
 - Évaluation d'expressions
 - Instructions de contrôle : if, for, while
 - Types de données : entiers, caractères, réels, tableaux, enregistrements
 - Fonctions d'entrées/sorties (scanf/printf)
 - Écriture et appel de fonctions

Survol du contenu du cours (ce semestre)

- Éléments d'architecture des ordinateurs (+mini-assembleur)
- Éléments de systèmes d'exploitation
- Programmation structurée impérative (éléments de langage C)
 - Structure d'un programme C
 - Variables : déclaration (et initialisation), affectation
 - Évaluation d'expressions
 - Instructions de contrôle : if, for, while
 - Types de données : entiers, caractères, réels, tableaux, enregistrements
 - Fonctions d'entrées/sorties (scanf/printf)
 - Écriture et appel de fonctions
 - Débogage

Survol du contenu du cours (ce semestre)

- Éléments d'architecture des ordinateurs (+mini-assembleur)
- Éléments de systèmes d'exploitation
- Programmation structurée impérative (éléments de langage C)
 - Structure d'un programme C
 - Variables : déclaration (et initialisation), affectation
 - Évaluation d'expressions
 - Instructions de contrôle : if, for, while
 - Types de données : entiers, caractères, réels, tableaux, enregistrements
 - Fonctions d'entrées/sorties (scanf/printf)
 - Écriture et appel de fonctions
 - Débogage
- Notions de compilation
 - Analyse lexicale, analyse syntaxique, analyse sémantique
 - préprocesseur du compilateur C (include, define)
 - Édition de lien

Survol du contenu du cours (ce semestre)

- Éléments d'architecture des ordinateurs (+mini-assembleur)
- Éléments de systèmes d'exploitation
- Programmation structurée impérative (éléments de langage C)
 - Structure d'un programme C
 - Variables : déclaration (et initialisation), affectation
 - Évaluation d'expressions
 - Instructions de contrôle : if, for, while
 - Types de données : entiers, caractères, réels, tableaux, enregistrements
 - Fonctions d'entrées/sorties (scanf/printf)
 - Écriture et appel de fonctions
 - Débogage
- Notions de compilation
 - Analyse lexicale, analyse syntaxique, analyse sémantique
 - préprocesseur du compilateur C (include, define)
 - Édition de lien
- Algorithmes élémentaires

Survol du contenu du cours (ce semestre)

- Éléments d'architecture des ordinateurs (+mini-assembleur)
- Éléments de systèmes d'exploitation
- Programmation structurée impérative (éléments de langage C)
 - Structure d'un programme C
 - Variables : déclaration (et initialisation), affectation
 - Évaluation d'expressions
 - Instructions de contrôle : if, for, while
 - Types de données : entiers, caractères, réels, tableaux, enregistrements
 - Fonctions d'entrées/sorties (scanf/printf)
 - Écriture et appel de fonctions
 - Débogage
- Notions de compilation
 - Analyse lexicale, analyse syntaxique, analyse sémantique
 - préprocesseur du compilateur C (include, define)
 - Édition de lien
- Algorithmes élémentaires
- Méthodologie de résolution, manipulation sous linux

*« L'informatique n'est pas plus la science des ordinateurs
que l'astronomie n'est celle des télescopes. »*

E. W. Dijkstra

Architecture de von Neumann

- John William Mauchly et John Eckert
- Idée d'une machine à programme stocké.
- Qu'est-ce que c'est ?

Représentation en binaire des informations

Représentation en binaire des informations

Definition (bit)

- Le chiffre binaire, ou *bit*, est l'équivalent binaire de nos chiffres décimaux. Il peut valoir soit 0 soit 1. Un bit est une quantité élémentaire d'information (oui ou non, ouvert ou fermé, etc.).

Représentation en binaire des informations

Definition (bit)

- Le chiffre binaire, ou *bit*, est l'équivalent binaire de nos chiffres décimaux. Il peut valoir soit 0 soit 1. Un bit est une quantité élémentaire d'information (oui ou non, ouvert ou fermé, etc.).
- L'information manipulée par un ordinateur est constituée de bits.

Représentation en binaire des informations

Definition (bit)

- Le chiffre binaire, ou *bit*, est l'équivalent binaire de nos chiffres décimaux. Il peut valoir soit 0 soit 1. Un bit est une quantité élémentaire d'information (oui ou non, ouvert ou fermé, etc.).
- L'information manipulée par un ordinateur est constituée de bits.
- les cases mémoires et les registres contiennent des mots mémoire : des suite de n bits, où n est fixé une fois pour toute par l'architecture matérielle.

Représentation en binaire des informations

Definition (bit)

- Le chiffre binaire, ou *bit*, est l'équivalent binaire de nos chiffres décimaux. Il peut valoir soit 0 soit 1. Un bit est une quantité élémentaire d'information (oui ou non, ouvert ou fermé, etc.).
- L'information manipulée par un ordinateur est constituée de bits.
- les cases mémoires et les registres contiennent des mots mémoire : des suite de n bits, où n est fixé une fois pour toute par l'architecture matérielle.
- les instructions du langage machine sont exprimées en binaire.

Représentation en binaire des informations

Definition (bit)

- Le chiffre binaire, ou *bit*, est l'équivalent binaire de nos chiffres décimaux. Il peut valoir soit 0 soit 1. Un bit est une quantité élémentaire d'information (oui ou non, ouvert ou fermé, etc.).
- L'information manipulée par un ordinateur est constituée de bits.
- les cases mémoires et les registres contiennent des mots mémoire : des suite de n bits, où n est fixé une fois pour toute par l'architecture matérielle.
- les instructions du langage machine sont exprimées en binaire.
- le **langage assembleur** en est un notation pratique pour les humains.

Représentation en binaire des informations

Definition (bit)

- Le chiffre binaire, ou *bit*, est l'équivalent binaire de nos chiffres décimaux. Il peut valoir soit 0 soit 1. Un bit est une quantité élémentaire d'information (oui ou non, ouvert ou fermé, etc.).
- L'information manipulée par un ordinateur est constituée de bits.
- les cases mémoires et les registres contiennent des mots mémoire : des suite de n bits, où n est fixé une fois pour toute par l'architecture matérielle.
- les instructions du langage machine sont exprimées en binaire.
- le **langage assembleur** en est un notation pratique pour les humains.

Nous en verrons plus sur les codages en binaire des données dans quelques cours.

Cycle d'exécution

- Le registre **compteur de programme** (CP) contient l'adresse du mot mémoire représentant la prochaine instruction

Cycle d'exécution

- Le registre **compteur de programme** (CP) contient l'adresse du mot mémoire représentant la prochaine instruction
- le contenu de ce mot est transféré de la mémoire centrale dans le **registre d'instruction** (RI)

Cycle d'exécution

- Le registre **compteur de programme** (CP) contient l'adresse du mot mémoire représentant la prochaine instruction
- le contenu de ce mot est transféré de la mémoire centrale dans le **registre d'instruction** (RI)
- CP est incrémenté (c'est à dire que sa valeur augmente de 1)
- le contenu de RI est décodé afin de déterminer l'opération à exécuter

Cycle d'exécution

- Le registre **compteur de programme** (CP) contient l'adresse du mot mémoire représentant la prochaine instruction
- le contenu de ce mot est transféré de la mémoire centrale dans le **registre d'instruction** (RI)
- CP est incrémenté (c'est à dire que sa valeur augmente de 1)
- le contenu de RI est décodé afin de déterminer l'opération à exécuter
- l'opération est exécutée (le contenu d'un ou plusieurs registres est modifié)

Cycle d'exécution

- Le registre **compteur de programme** (CP) contient l'adresse du mot mémoire représentant la prochaine instruction
- le contenu de ce mot est transféré de la mémoire centrale dans le **registre d'instruction** (RI)
- CP est incrémenté (c'est à dire que sa valeur augmente de 1)
- le contenu de RI est décodé afin de déterminer l'opération à exécuter
- l'opération est exécutée (le contenu d'un ou plusieurs registres est modifié)
- Fin du cycle d'exécution et démarrage d'un nouveau cycle

Instructions

Une instruction type comporte un code d'opération et, si nécessaire, une ou deux opérandes.

Instructions

Une instruction type comporte un code d'opération et, si nécessaire, une ou deux opérandes.

Vocabulaire : dans l'expression arithmétique usuelle $3 + 5$, le signe $+$ est l'opérateur et les nombres 3 et 5 sont les opérandes.

stop	Arrête l'exécution du programme.
noop	N'effectue aucune opération.
saut i	Met le compteur de programme à la valeur i .
sautpos $ri\ j$	Si la valeur contenue dans le registre i est positive ou nulle, met le compteur de programme à la valeur j .
valeur $x\ ri$	Initialise le registre i avec la valeur x .
lecture $i\ rj$	Charge, dans le registre j , le contenu de la mémoire d'adresse i .
écriture $ri\ j$	Écrit le contenu du registre i dans la mémoire d'adresse j .
inverse ri	Inverse le signe du contenu du registre i .
add $ri\ rj$	Ajoute la valeur du registre i à celle du registre j (la somme obtenue est placée dans le registre j).
soustr $ri\ rj$	Soustrait la valeur du registre i à celle du registre j (la différence obtenue est placée dans le registre j).
mult $ri\ rj$	Multiplie par la valeur du registre i celle du registre j (le produit obtenu est placé dans le registre j).
div $ri\ rj$	Divise par la valeur du registre i celle du registre j (le quotient obtenu, arrondi à la valeur entière inférieure,

Trace d'exécution

<i>Instructions</i>	Cycles	CP	r0	r2	10	11
Initialisation	0	1	?	?	14	?
lecture 10 r0	1	2	14			
valeur 5 r2	2	3		5		
soustr r2 r0	3	4	9			
sautpos r0 8	4	8				
écriture r0 11	5	9				9
stop	6	10				

Démos et fin