

ALMO

Objectifs et Structure du module Présentation d'éléments d'architecture

Pourquoi ce cours zéro

- ALMO signifie Architecture Matérielle et Logicielle des Ordinateurs
C'est un sujet très vaste qui va de la programmation du processeur en assembleur (le langage de base) à la multi-programmation sur des ordinateurs multi-coeurs.
- Le module ALMO propose de répondre à des questions telles que :
 1. Que trouve-t-on dans un ordinateur ?
 2. Comment un programme fait pour lire des données depuis un clavier et envoyer ses résultats sur un écran ?
... et en même temps ... :-)
faire de la décompression d'image ou autre chose ?
 3. Comment exécuter rapidement un programme quand la mémoire est lente ?
- L'objectif de ce premier cours est de présenter les différents aspects abordés et de poser les questions auxquelles nous allons apporter des réponses au fil des semaines.

Plan du cours

1. Pourquoi ce cours zéro
2. Contenu du module
3. Organisation pratique
4. Présentation du processeur MIPS



crédit : <http://coursz.com/definition-d-un-ordinateur>

Sujets abordés dans le module ALMO

Dans l'ordre

- Architecture Matérielle et Logicielle d'un Ordinateur
- Programmation en assembleur du processeur MIPS
- Chaîne de compilation C
- Accès aux périphériques via les appels (au) système (d'exploitation)
- Organisation et gestion de la mémoire
- Hiérarchie des caches
- Structure d'un programme exécutable, chargement en mémoire
- Interaction du système d'exploitation avec les périphériques
- Multiprogrammation
- Virtualisation de la mémoire

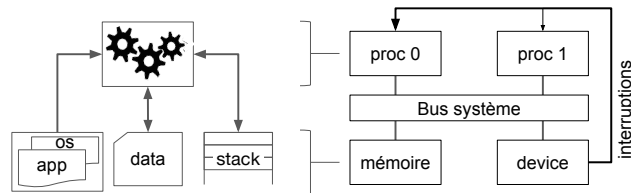
Architecture Matérielle et Logicielle d'un Ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs (unités de calcul) de la mémoire (et même une hiérarchie de mémoire)
- des périphériques (écran, clavier, disque, réseau, capteurs, etc.)
- une infrastructure de communication entre le(s) core(s), la mémoire et ses contrôleurs d'E/S

Un ordinateur exécute :

- une ou plusieurs *applications*
- sous le contrôle d'un système d'exploitation



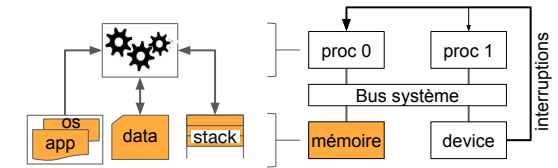
Architecture Matérielle et Logicielle d'un Ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs
- de la mémoire
- des périphériques (devices)
- une infrastructure de communication

Un ordinateur exécute :

- une ou plusieurs *applications*
- sous le contrôle d'un système d'exploitation



Mémoire

où sont enregistrés les instructions, les données et les contextes d'exécution

- Qu'est-ce qu'un bit, un octet, un mot, une ligne, une page ?
- Qu'est-ce qu'un espace d'adressage ?
- Qu'est-ce que la hiérarchie de la mémoire ?
- Que signifie cache de mémoire ?
- Qu'est-ce que la mémoire virtuelle ?

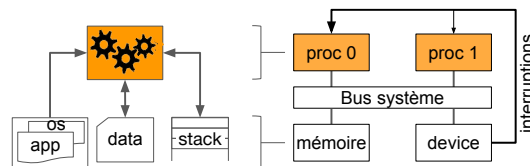
Architecture Matérielle et Logicielle d'un Ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs
- de la mémoire
- des périphériques (devices)
- une infrastructure de communication

Un ordinateur exécute :

- une ou plusieurs *applications*
- sous le contrôle d'un système d'exploitation



Coeur (core)

composant exécutant les instructions du programme (séquence d'instructions)

- Est-ce que processeur et coeur sont synonymes ?
- Quel est le schéma de principe d'un coeur de calcul ?
- Quelles sont les types d'instructions ?
- Que veut dire RISC ou CISC ?
- Les instructions travaillent sur des registres, qu'est-ce qu'un registre ?
- En combien de temps s'exécute une instruction ?
- Comment fonctionne un ordinateur multi-coeurs ?
- Qu'est-ce qui limite le nombre de coeurs dans un ordinateur ?

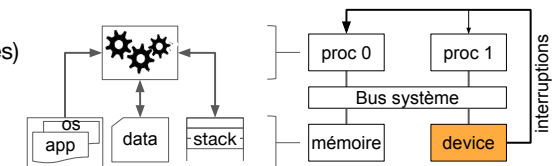
Architecture Matérielle et Logicielle d'un Ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs
- de la mémoire
- des périphériques (devices)
- une infrastructure de communication

Un ordinateur exécute :

- une ou plusieurs *applications*
- sous le contrôle d'un système d'exploitation



Périphériques

composants connectés au(x) coeur(s) réalisant des fonctions spécifiques

- Quels sont ces périphériques ?
- Comment sont-ils commandés ?
- Quelle différence entre un périphérique cible et un périphérique initiateur ?
- Comment un périphérique demande-t-il un service au système ?
- Que signifie ISR ?

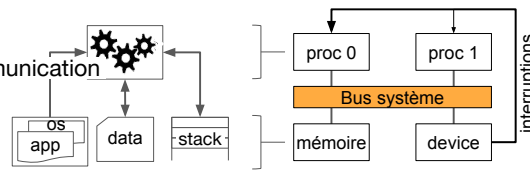
Architecture Matérielle et Logicielle d'un Ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs
- de la mémoire
- des périphériques (devices)
- une infrastructure de communication

Un ordinateur exécute :

- une ou plusieurs applications
- sous le contrôle d'un système d'exploitation



Infrastructure de communication

utilisé par un coeur pour accéder à la mémoire et aux périphériques.

- Quels sont les types d'infrastructure de communication ?
- Qu'est-ce qu'un bus de communication ?
- Combien de coeurs peuvent utiliser la mémoire sans se gêner ?
- Quelle est la durée d'attente avant d'accéder à la mémoire ?

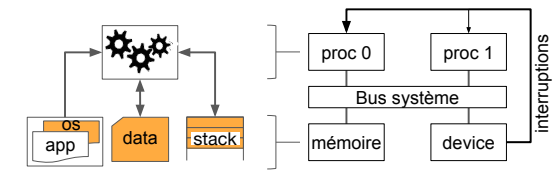
Architecture Matérielle et Logicielle d'un Ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs (unité de calcul)
- de la mémoire
- des périphériques (devices)
- une infrastructure de communication

Un ordinateur exécute :

- une ou plusieurs applications
- sous le contrôle d'un système d'exploitation



Système d'exploitation

programme permettant l'exécution des applications et gérant l'allocation du matériel (coeur de calcul et mémoire) et l'accès aux entrées-sorties.

- Quels sont les services proposés par un système d'exploitation ?
- Comment garantir les privilèges du système d'exploitation ?
- Comment gérer l'exécution de plusieurs threads ?
- Qu'est-ce qu'un pilote de périphériques ?
- Comment une application peut-elle attendre un événement externe ?

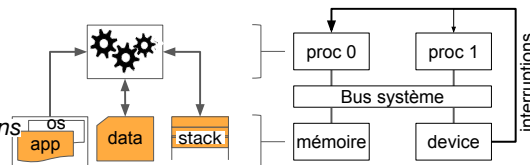
Architecture Matérielle et Logicielle d'un Ordinateur

Un ordinateur contient :

- un ou plusieurs coeurs (unité de calcul)
- de la mémoire
- des périphériques (devices)
- une infrastructure de communication

Un ordinateur exécute :

- une ou plusieurs applications
- sous le contrôle d'un système d'exploitation



Applications

par exemple : un lecteur multimédia, un jeu, un traitement de texte.

- Quelle différence entre application, tâche, processus, thread ?
- De quoi se compose un processus ?
- Qu'est-ce qu'un thread (fil d'exécution) ?
- Qu'est-ce qu'un contexte d'exécution ?

Organisation et évaluation du module

Equipe pédagogique

- Franck Wajsburt (cours et groupe 1 lundi)
- Alain Greiner (groupe 2 lundi)
- Quentin Meunier (groupe 3 mercredi)
- Emmanuel Encrenaz et Pirouz Bazargan (groupe 4 vendredi)

Conseils

- Ne pas venir en cours réduit fortement vos chances de succès.
- Les slides sont sur le site, lisez-les avant de venir en cours.
- Les TD doivent être préparés (lus au minimum) chez-vous.
- Des séances de soutien et de révision vous seront proposées en cours d'année et avant les examens

Evaluation

- 1 TP noté par semaine (10 en tout) ⇒ 20% de la note
- 4 interrogations écrites pendant les TD ⇒ 20% de la note
- 1 examen composé de 4 parties ⇒ 60% de la note

Calendrier 2019

sem	Cours	TD/TP	Interro
37	C0 Vue d'ensemble du module		
38	C1 Architecture Externe et Assembleur MIPS32	TD1 Assembleur MIPS32 / Instructions de base TP1 Simulateur MARS	
39	C2 Utilisation de la pile & appels de fonctions	TD2 Utilisation de la pile / appels de fonctions TP2 Fonctions imbriquées et récursives	
40	C3 Chaîne de compilation	TD3 Assembleur : programme de tri TP3 Génération de code avec GCC : exécution avec MARS	
41	C4 Bus système et périphériques / GIET 1	TD4 Bus système et périphériques TP4 Exécution de code sur architecture mono-processeur	Assemb
42	C5 Hiérarchie mémoire et techniques de cache	TD5 Principe des mémoires caches TP5 Effets de cache	
43	C6 Caches (suite)	TD6 Influence des caches sur les performances TP6 Mémoires cache : mesure de performance	
44	Vacances de la toussaint		
45	C7 Accès aux périphériques : ICU, TTY, TIMER / GIET 2	TD7 Analyse GIET TP7 Communications par interruptions	Cache
46	Partiel corrigé		
47	C8 Périphériques DMA : contrôleurs disque et réseau / GIET 3	TD8 Périphériques TP8 contrôleurs IOC, DMA et FB	
48	C9 Multiplexage temporel & Commutation de tâches / GIET 4	TD9 Commutation de tâches TP9 Commutation de tâches	Giet
49	C10 Programmation parallèle multi-tâches	TD10 Partage du bus systématiquement TP10 Architecture multi-processeur	
50	C11 Principe Mémoire Virtuelle et MMU/TLB	TD11 mémoire virtuelle paginée TP11 Rattrapage	Switch
51	Révisions guidées		
52	Vacances de fin d'année		
1	Vacances de fin d'année		
2	1ère Session d'examen		

Outils utilisés en TP

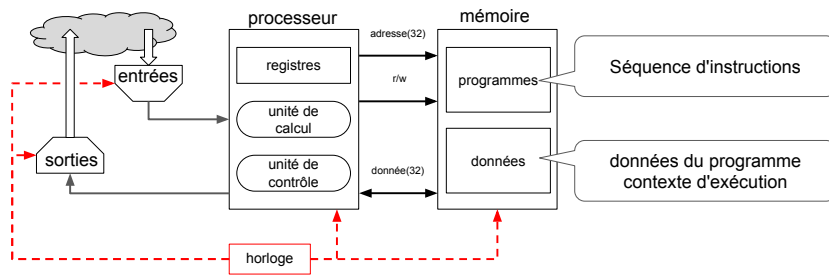
- Simulateur du processeur MIPS32
Permet l'exécution en pas-à-pas des programmes assembleur
 - mars
- Cross-compileur MIPS32 pour MARS
Produit le code assembleur à partir des programmes C
 - mipsel-mars-elf-gcc
- Simulateur de la machine ALMO
Simule un ordinateur complet processeur—mémoire—périphériques
 - simul_almo_generic
- Chaîne de compilation pour la machine ALMO
Produit le code binaire exécutable pour le simulateur
 - mipsel-unknown-elf-as : assembleur
 - mipsel-unknown-elf-gcc : compilateur
 - mipsel-unknown-elf-ld : editeur de lien
 - mipsel-unknown-elf-objdump : desassembleur

Ressources et documents

- web : <https://www-soc.lip6.fr/trac/sesi-almo>
 - Programme du module et Enoncés de TD et TP
 - Configuration de l'environnement des TP
 - Documents
 - Architecture externe et Langage d'assemblage du MIPS
 - Code du système d'exploitation GIET
 - Enoncés des TD et TP (non autorisé à l'examen)
 - annales corrigées d'examens (non autorisé à l'examen)
- fichiers : /Infos/lmd/2019/licence/ue/LU3IN004-2019oct/
 - code source des TP
 - code source du "système d'exploitation" GIET
 - binaires des outils
 - script d'environnement

Processeur et mémoire principes

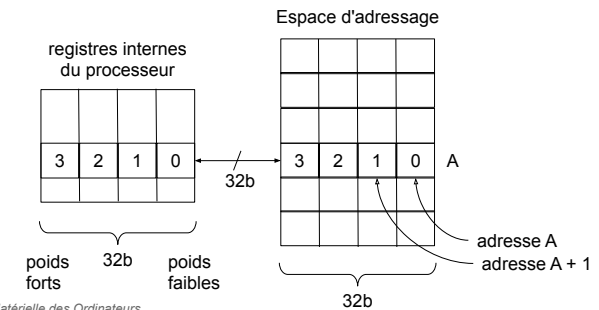
Architecture de Von Neumann



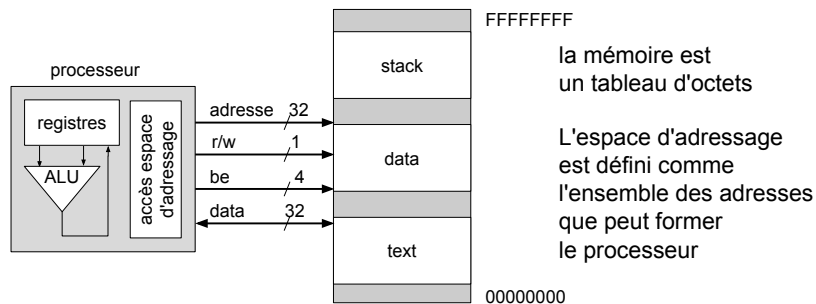
- La mémoire stocke le programme, ses données et le contexte d'exécution.
- Des contrôleurs d'entrée-sortie permettent d'accéder à l'extérieur.
- L'unité de contrôle séquence l'exécution des instructions du programme.
- L'unité de calcul exécute des opérations arithmétiques et logiques.
- Les registres sont de la mémoire utilisés par l'unité de calcul.
- Une horloge cadence le fonctionnement, p. ex. une instruction par cycle.

Endianness

- Les registres font 32 bits donc un mot de 4 octets (3 [fort] - 2 - 1 - 0 [faible])
- Quand on veut désigner un mot en mémoire, on donne toujours l'adresse du premier octet (celui dont l'adresse est la plus petite).
- Il existe deux conventions : **little endian** et **big endian**
- Convention **little endian**
 - L'octet de poids faible d'un mot est rangé à l'adresse la plus petite en mémoire
- Convention big endian : l'octet de poids fort est à l'adresse la plus petite



Espace d'adressage du processeur



- Les adresses font 32 bits ce qui signifie 4 GiB
 - 1kiB = 2^{10} 1MiB = 2^{20} 1GiB = 2^{30}
 - 4 GiB = $2^{32} = 2^{(2+10+10+10)} = 4 * 2^{10} * 2^{10} * 2^{10} = 4$ Gibi Bytes
- Chaque octet a une adresse mais les échanges sont par mots de 4 octets.
- L'ALU (boîte à opérations) travaillent sur des nombres de 32 bits
⇒ le MIPS est une architecture 32 bits
- Les registres sont donc aussi sur 32 bits.

La mémoire n'est pas l'espace d'adressage

- L'espace d'adressage, c'est l'ensemble des adresses que peut former le processeur;
- La mémoire, c'est le composant dans lequel sont stockés :
 - Le code du programme (les instructions)
 - Les données du programme
 - La pile d'exécution du programme
- Certaines adresses de l'espace d'adressage permettent d'adresser les registres de commandes des périphériques.
- Ces registres sont des cases de mémoires ou des "commandes" (lire ou écrire à ces adresses permet de commander le périphérique).

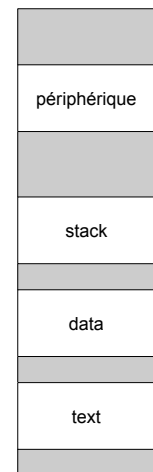
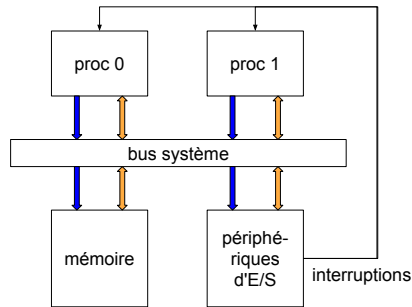


Schéma de principe d'un ordinateur

- Le but de l'UE est de comprendre le fonctionnement d'un ordinateur.
- Un ou plusieurs processeurs se partagent le même espace d'adressage et les mêmes périphériques d'entrées-sortie.
- Nous aborderons la programmation du processeur, l'accès à la mémoire, l'accès au périphérique sous l'angle matériel et logiciel.



En résumé, nous avons vu :

- L'objectif du module : vous faire comprendre comment fonctionne un ordinateur et comment les programmes s'exécutent.
- Les questions auxquelles vous aurez des réponses au cours des semaines : sur l'architecture et sur la programmation.
- L'organisation et le planning du module
- Les outils utilisés pendant les travaux pratiques
- Quelques concepts matériels
 - architecture de Von Neumann
 - espace d'adressage
 - endianness ou comment sont rangés les octets d'un mot en mémoire
 - schéma de principe d'un ordinateur
 - quelques caractéristiques du processeur MIPS32

MIPS32

- Le MIPS32 est de type RISC
 - RISC : Reduced Instruction Set Computer : 57 instructions.
 - Toutes les instructions sont codées sur 1 mot (4 octets).
 - Le MIPS démarre une instruction par cycle d'horloge.
-
- Le MIPS32 dispose de 32 registres de travail (GPR : General Purpose Register)
 - L'accès à l'espace d'adressage se fait uniquement par des lectures et des écritures de registres GPR (pas de calcul)
 - Le MIPS32 contient une unité de calcul fonctionnant avec les GPR et plusieurs coprocesseurs spécialisés
 - c0 : coprocesseur du système d'exploitation
 - c1 : coprocesseur de calcul flottant
 - c2 : coprocesseur de gestion de la mémoire
 - Le MIPS32 dispose de deux modes de fonctionnement
 - mode utilisateur avec des droits limités (instructions, registres, espace)
 - mode système (ou kernel) avec tous les droits

Prochain cours

Architecture externe et programmation assembleur du MIPS32

- Types d'instruction
- Principe d'exécution d'une instruction
- Programmation assembleur
 - Concepts : symbole, directives, macro-instructions, ...
 - Structure d'un programme
 - Allocation des données globales
 - Programmation des boucles
- Présentation de l'application simulateur MARS