INF720 - Langage C et Systèmes d'Exploitation

Introduction SE

Responsable : Thomas Robert (C234-4)
Auteurs : Thomas Robert & Etienne Borde

Informations pratiques

- **Note inf720 : 2 parties… consignes valides pour la partie SE…**
- **10 pts : min (10, 0.7 exam +0.6 TPs)**
- **Un examen final**
- **Un contrôle continu**
  - **3 niveaux de validation par TP 0 1 ou 2 pts (hors tp d'introduction)**
  - **Niveaux clairement identifiés dans le TP normalement ou sur le site de rendu**

Rappel : un programme C jusqu'à présent

- Du texte décrivant
  - Une séquence d'instructions
  - Le programme est structuré en fonctions
  - Avec un point de démarrage standard appelé point d'entrée
  - Comment s'appelle le point d'entrée du programme en C sous linux ?
  - Des fonctions fournies par des bibliothèques pour des opérations d'entrée sorties
    - Sur quoi se font les entrées sorties par votre expérience au sein de l'UE ?
- A coté, il y a :
  - Des bibliothèques et une chaine de compilation pour créer des exécutables
  - Un mécanisme de chargement et de lancement du programme (la ligne de commande et les mécanismses sous jacents)
    - Quel composant fournit ces services ?
- Ce que nous allons voir : les services offerts par le système d'exploitation

Savez vous répondre aux questions :

- Comment un calculateur stocke/accède aux données (matériel)?
- Quelles opérations ***élémentaires*** sait faire un calculateur ?
- Pourquoi passe-t-on par un langage de programmation ?
  *(pourquoi ne décrit on pas un programme en opérations **élémentaires**)*
- Quelles sont les étapes menant de la définition d'un programme à son exécution ?
- A quoi sert un système d'exploitation

Objectif : connaître les réponses et savoir les expliquer
(principe – limites/contraintes d'usage)

Le lien avec la sécurité

- Constat : énorme écart entre
  - Le comportement effectif du calculateur
  - Le comportement supposé par l'utilisateur d'un programme
- « On vous cache plein de choses très compliquées »
  - Sur ce qui se passe dans le processeur
  - Sur comment certaines services de protection sont implémentés
- Quel peut être le problème si on prend ceci pour acquis ?
  - Qui a entendu parler de meltdown et spectre ?
  - Attaque Hardware qui contredit une garantie de confidentialité au niveau hardware.

Quel est l'intérêt de connaître les réponses

- **Savoir ce qui faisable :** comprendre comment l'ordinateur fonctionne et va « exécuter vos programmes » permet de réaliser des applications plus complexes mais aussi de savoir où il peut y avoir des problèmes ….

- **Comprendre les programmes existants et voir leur défauts:** comprendre l'usage des ressources de calcul et stockage mais aussi les motifs de programmation usuels permet d'avoir plus d'assurance pour analyser ces programmes

- **Se donner les moyens pour comprendre et réaliser les Tps/cours autres que INF720 ….**

Suite de la séance

- **Introduction au concept de plateforme d'exécution**
  - Fonctionnalités et composants clés
  - Calculateur : principes de fonctionnement
  - Petite paranthèse sur les mécanismes de protection « élémentaires » du processeur de type contrôle d'accès et de flux.

- **Du programme à son exécution**
  - Rappel des caractéristiques d'un langage de programmation
  - Stratégies d'exécution :
    compilation en instruction machine ou interprétation

- **Objectif: introduction et motivation des notions utilisées en apprentissage du C**
  - Représentation de l'information binaire
  - Adresse
  - Compilation

Support d’exécution une introduction

Fonctionnalités et exemples

- **Fonctions :**
  - Fournir les moyens de stocker de l'information
  - Fournir les moyens d'exécuter le traitement correspondant à un programme : notion de processus
  - Fournir des moyens d'interagir avec d'autres supports d'exécution ou l'environnement physique.
  - Réguler les intéractions entre les processus et le ressources (réguler = protégéer + partager)
- **Exemples de plateformes différentes :**

Portables **Ordinateur de bord** **Poste de travail**

Fonctionnalités dérivées

- **Partager = répartir entre différents « acteurs » / « utilisateurs »**
- **=> composant phare des OS : le concept d'utilisateur**
- **Protéger => un modèle de protection**
  - **Ce que nous verrons = contrôle d'accès essentiellement**
  - **Attention : notre but n'est pas de faire SR2I204 en avance … ce sera un avant goût avec un petite plongée dans les détails….**

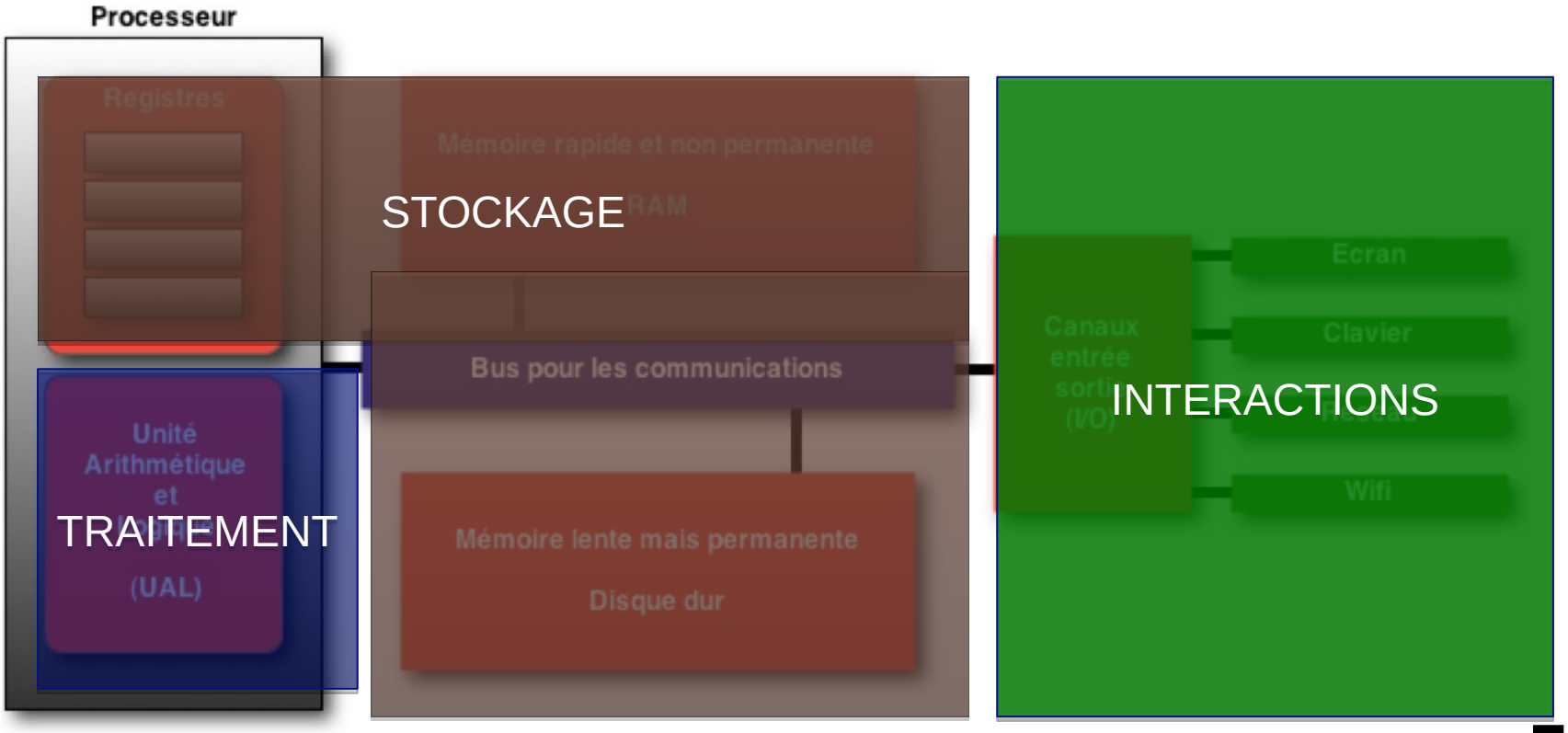
Un support d'exécution : les éléments clés

- **Un calculateur = regroupement de composants électroniques fournissant des capacités/ressources**
  - de stockage des données
  - de capture de stimuli extérieurs (physiques)
  - de communication de données avec d'autres calculateurs
  - de calcul et transformation de données, contrôle d'exécution

- **Un système d'exploitation = logiciel pour organiser, simplifier et optimiser l'accès aux ressources**
  - Mise en place du partage et protection des ressources *(traitement / stockage)*
  - Simplification de l'usage et de l'accès aux ressources

Vision architecturale du calculateur

- **Intérêt : présenter une cartographie des ressources + de leurs interactions**



Stockage d'information

- **Un bit : unité de stockage d'information**
  peut prendre 2 valeurs : 0 et 1 (2 états possibles)
- **Les données manipulées par un processeur == séquences finies de bits**
  - 4 bits : 0110; 8 bits : 11010011…
- **Une séquence de N bits peut représenter n'importe quel élément d'un ensemble de taille inférieure à 2^N**
- **Ces séquences de bits peuvent être interprétées comme:**
  - Des nombres : 1001 = représentation base 2 de 9
  - Des mois : 000010000000 pour Mai

Une séquence de bits =
CE QUE VOUS DECIDEZ D'EN FAIRE (INTERPRETATION)

Un petit mot sur les systèmes numériques et le principe de représentation….

- **Séquence N bits interprétable comme un nombre en base 2**
- **ATTENTION : cela ne veut pas dire que l'on stocke toujours un nombre en mémoire**

  000010000000 voulait dire « Mai » et pas 2^7
- **Le calculateur est mécanique !**
- **Il a des opérations typées qui interprète le champ binaire sans se poser de questions ...**
- **Représentations classiques :**
  - Système binaire : $b_7b_6….b_0$

    $$\sum_{i=0}^{7} b_i \cdot 2^i$$
  - Complément à 2 : $b_7b_6….b_0$

    $$-(b_7 \cdot 2^7) + \left(\sum_{i=0}^{6} b_i \cdot 2^i \right)$$
  - Symbole ASCII : des symboles « % » « A » « 7 »
    http://en.wikipedia.org/wiki/ASCII

Mauvaises interprétation ….

- **Un programmeur utilise le type de variable (unsigned int) et lit un champ binaire depuis un fichier contenant des nombres en compléments à deux …**

- **Les nombres sont de petits nombres négatifs normalement, qu'en est il si l'on affiche via printf la valeur de la variable unsigned int ?**

Organisation du stockage

- **Organisation des bits en octet = 8 bits (granularité usuelle), plus pratique à écrire en base 16 (symboles: 0123456789ABCDEF)**

- **Problème du volume :**
  - Mozilla : 178 Mo de données utilisées dans certains cas
  - Base de données client : ~quelques Téra octets

Comment retrouver les données intéressantes dans un tel volume ?

- **Index et valeur référence :**
  - Index : une fonction associant un identifiant à une donnée (mieux quand il est unique)
  - Valeur de Référence : une valeur d'identifiant pouvant être utilisée pour récupérer la donnée qui lui est associée via l'index.
  - « résoudre la référence Vref » = Index(Vref) = obtenir la valeur

En pratique (1) mémoire

- **Mémoire (RAM) : composant permettant de stocker un ensemble d'octets de grande taille multiple d'une puissance de 2 ^{K}**

- **Caractéristiques :**
  - Index **:** 1 relation associant 1 **adresse** (i.e. un numéro) à chaque case mémoire de taille 1 octet
  - Référence à **un octet : adresse – contenu binaire de la case mémoire**

  - Remarques:
    - Granularité d'accès : groupe de 1 à 8 octets d'index consécutifs (ce groupe est appelé **mot mémoire**)
    - Référence à **plusieurs octets contigus :**
      1 adresse @ + nombre d'octets (1 à 8) à lire à partir de @

En pratique (2) Les autres stockages

- **Justification technologique de leur intérêt (et/ou):**

  Plus rapide, moins couteux utilisable directement par le processeur …
  beaucoup de raisons très variées
- **Registres : stockage mémorisant quelques octets**
  - taille = mot mémoire (8 octets sur une machine 64 bits)
  - Directement nommés => pas d'index à base de numéros
  - là où le processeur réalise ses calculs
- **Stockage de masse (disque dur) :**
  - Accès = bloc pour transferts vers ou depuis la mémoire (ensemble d'octet ~ 1000 octets)
  - Index structuré : (cas disque dur non SSD)
    - Plateau, cylindre … ~ adresse postale: rue, ville, pays

Bilan sur l’organisation du stockage

- **3 types de stockages essentiellement :**

  Registres, Mémoire, Stockage de masse (disque dur)

- **Notion d’index et de référence**
  - Index : le système d’association (id, valeur), l’id peut être défini en plusieurs morceaux
  - Référence : un id (permettant de retrouver une valeur si il est complet et à jour…)

- **Références aux données en mémoire par le processeur**
  - Index: adresse 1 octet
  - Référence: adresse (i.e. un numéro)

Capacités de traitement

- **Composant clé : le coeur d'exécution**
- **Traitements = instructions (opérations *élémentaires*)**
- **Jeu d'instructions = {des opérations élémentaires supportées par le processeur}**
  - Stockées en mémoire, donc
    - représentation binaire
    - Chaque instruction élémentaire à une adresse
  - Jeu d'instruction limitées
    - Calculs numériques simple (+ ; - ; * ; / …)
    - Transferts de données entre supports de stockage (mémoire registre ou registre registre ou registre mémoire)
    - Evaluations de conditions booléennes
    - Utilisent principalement les registres (pour des questions de performance)

      ….

- **Etat de l'exécution = 1 registre pour la prochaine exécution + 1 registre pour l'adresse de la pile + registre d'état (i.e. conséquence de l'exécution d'une instruction précédante**

Le cas particulier du Déroutement avec restauration

On veut calculer

$$\sum_{j=121}^{242} j$$

Rappel Somme 1…N = N*(N-1)/2
=> calcul possible en O(1)

idée N dans Ra et résultat dans Rb
Rb doit contenir ((Ra-1) * Ra)/2

Calcul de N*(N-1)/2 :

@	Instruction
911	Rb ← Ra-1
912	Rb ←Ra*Rb
913	Rb ←Rb/2
914	….
915	….

Le cas particulier du Déroutement avec restauration

On veut calculer

$$\sum_{j=121}^{242} j$$

Rappel Somme 1…N = N*(N-1)/2
=> calcul possible en O(1)

idée N dans Ra et résultat dans Rb
Rb doit contenir ((Ra-1) * Ra)/2

Calcul de N*(N-1)/2 :

@	Instruction
1	Ra ← 120
2	Call 911
3	Rc ← Rb
4	Ra ← 242
5	Call 911
6	Rb ← Rb -Rc

@	Instruction
911	Rb ← Ra-1
912	Rb ← Ra*Rb
913	Rb ← Rb/2
914	Ret
915	….

Bilan sur les capacités d'exécution

- **Représentation en mémoire de séquence d'opérations dont l'exécution est automatisée**
- **Capacité à définir la prochaine instruction à exécuter (e.g. déroutement) par la valeur d'un registre ou par une instruction (i.e. calcul ou saut constant)**
- **Capacité à réutiliser une même séquence d'instructions dans différents « contextes » (i.e. appel de fonctions)**

Le besoin derrière le concept de processus ?

- **Sur un processeur moderne : possibilité de mener en parallèle plusieurs exécutions (1 par « coeur » d'exécution**
- **1 coeur d'exécution = une forme d'automate avec un registre pour la prochaine instruction (beaucoup plus complexe en vrai mais on schématise…)**
- **Pb historique : il n'y avait q'un seul coeur avant ….**
- **Comment faisait on pour avoir son client mail préféré + son traitement de texte + un terminal actif….**
- **=> on crée l'illusion d'avoir autant de « contexte d'exécution" que nécessaire => le concept de processus**

Le besoin derrière le concept de système de fichier

- Le contenu d'un disque n'est pas directement accessible en mémoire
  - 2 niveaux de structuration le bloc et l'octet
  - 1 octet sur le disque est donc identifié par un numéro de bloc et un numéro d'octet dans le bloc
- Il faut simplifier et cela ne réponds pas aux usages que l'on voudrait faire : organiser des unités d'information de taille variables indexées de manière lisible.
- Système de fichier =
  - Concept de fichier = unité de stockage de taille variable indépendante de l'organisation physique!
  - Concept de dossier = méthode d'indexation des fichiers