L'écho du A

Jean-Jacques.Levy@inria.fr

http://www.jeanjacques-levy.com/

tel: 01 39 63 56 89

Catherine Bensoussan

cb@lix.polytechnique.fr

Laboratoire d'Informatique de l'X

Aile 00, LIX

tel: 34 67

http://w3.edu.polytechnique.fr/informatique

Objectif de la semaine

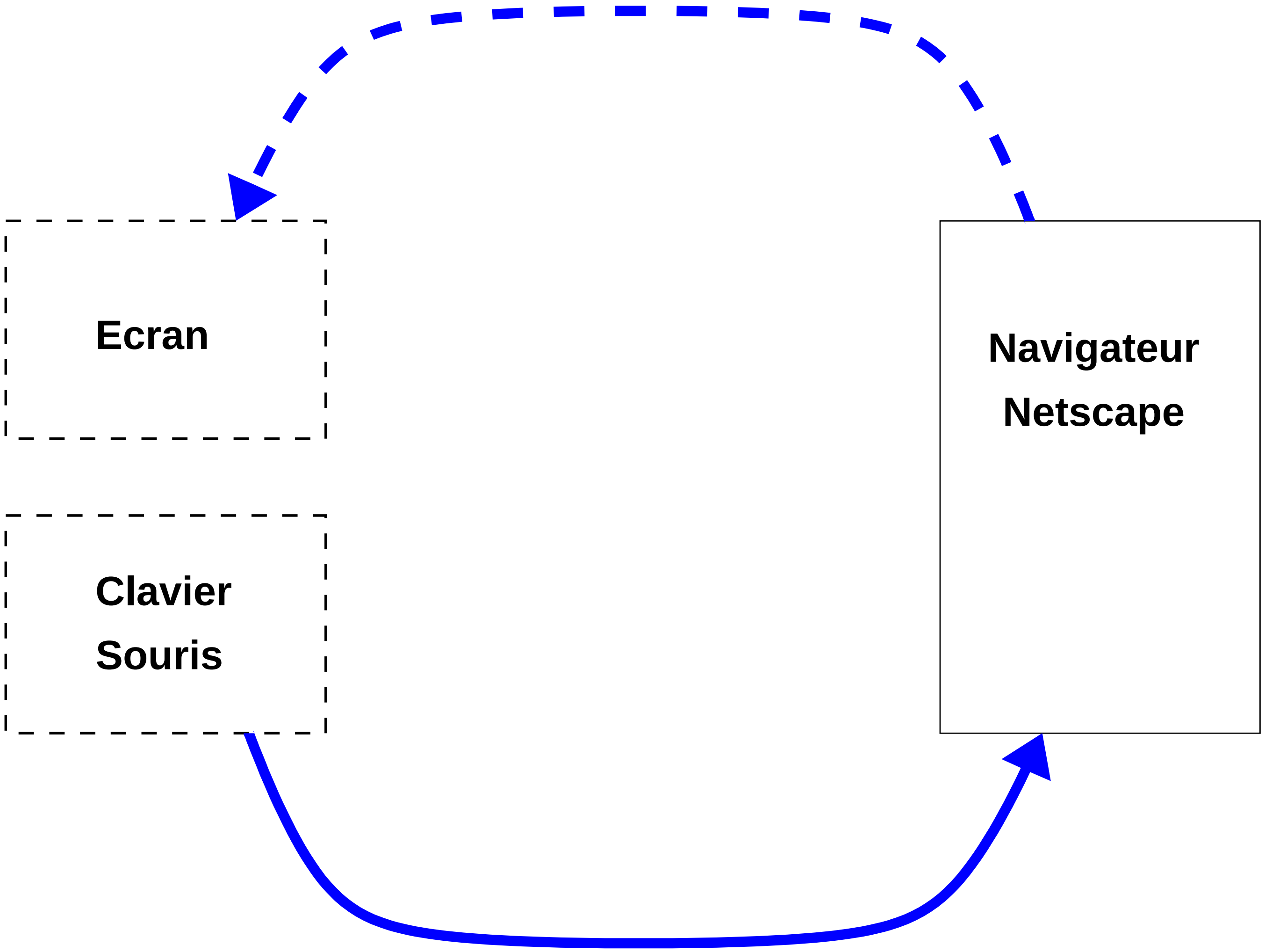
- Rattraper les différences de connaissances avec les élèves ayant pris l'option informatique en MP.
- Apprendre à se servir de l'environnement informatique de l'X
- Ecrire de petits programmes
- 5 cours + 5 TDs + 5 tutorats
- Finir les programmes et les faire marcher.
- Plus tard, dans la scolarité à l'X ou après, de plus gros programmes.

  L'informatique est une des activités les plus complexes conçues par l'homme.

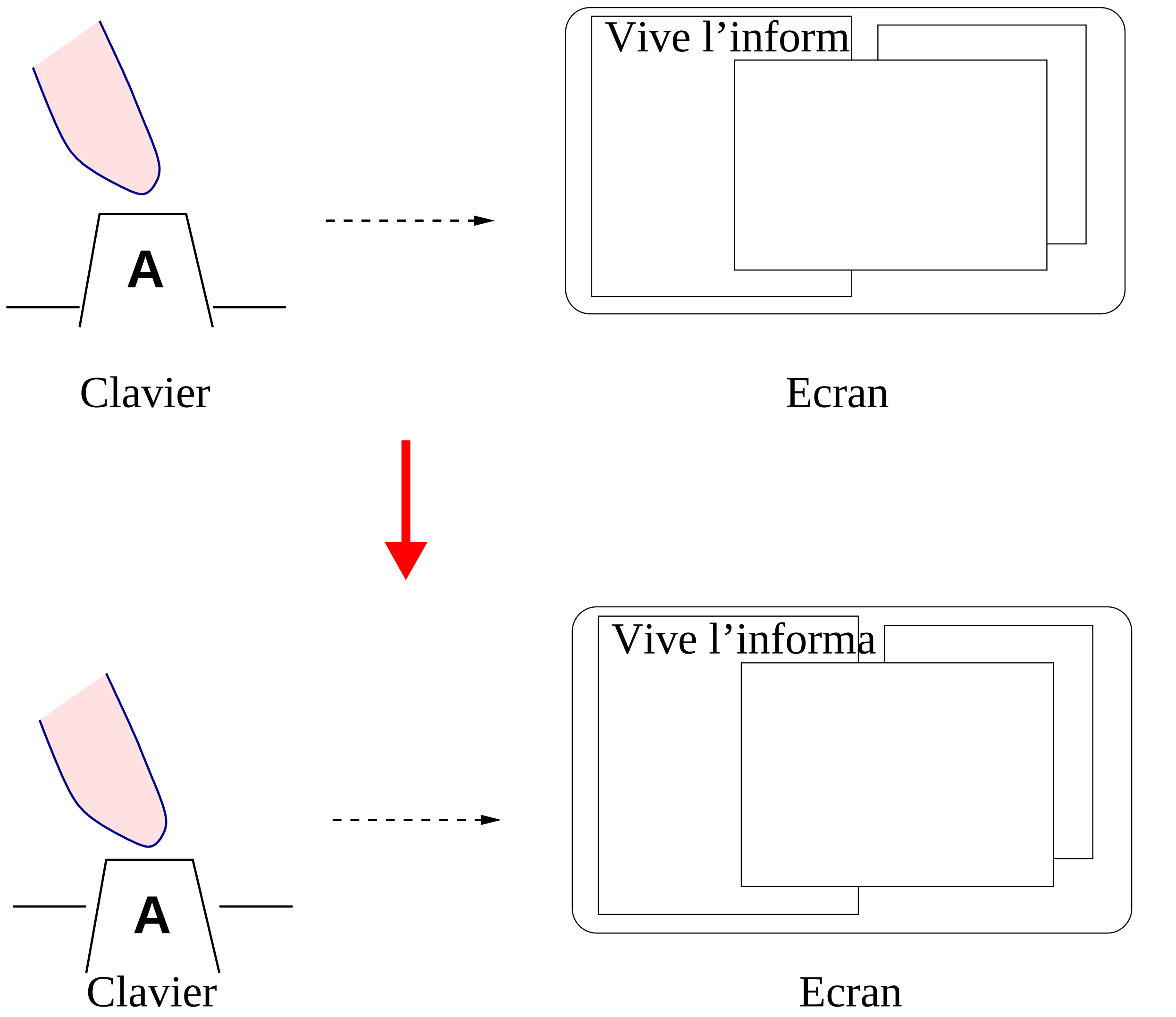
Plan

1. Fenêtres multiples
2. Ecrans digitaux
3. Applications multiples
4. Indépendance du système d'exploitation
5. Systèmes d'exploitation
6. Matériel
7. Quelques commandes utiles
8. La programmation mère de tous les problèmes
9. Langages de programmation

Clavier – Application



Du clavier à l'écran



Le matériel et le traitement bas niveau

1. Contact sur le front descendant de la touche A

   Signal émis clavier $\rightarrow$ ordinateur

2. Interruption du processeur (on sauve l'état courant)

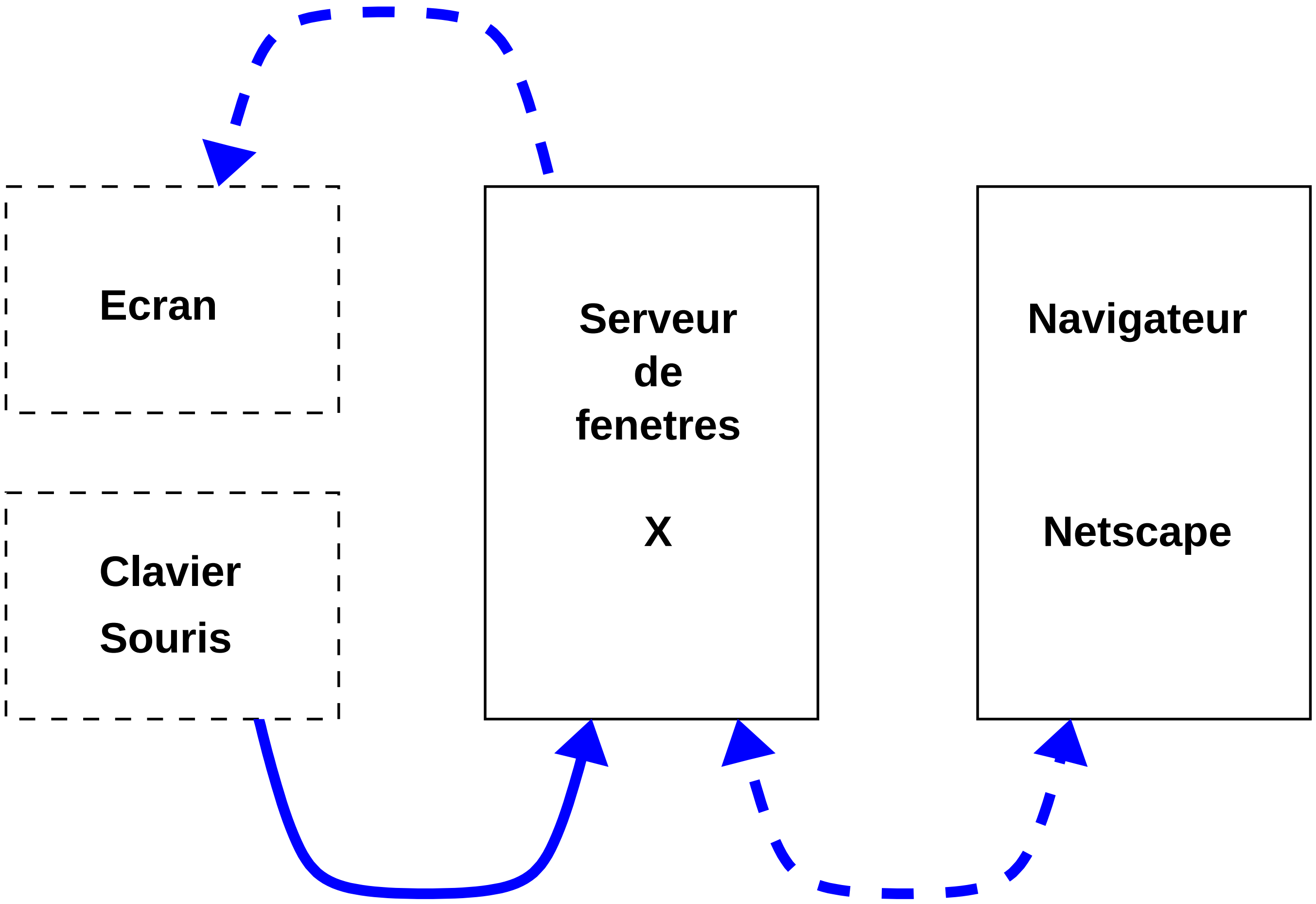
   Traitement de l'interruption: un événement nouveau rangé dans la queue des événements d'entrée du système d'exploitation

3. Le processeur repart dans l'état sauvé. Il reprend le calcul courant.

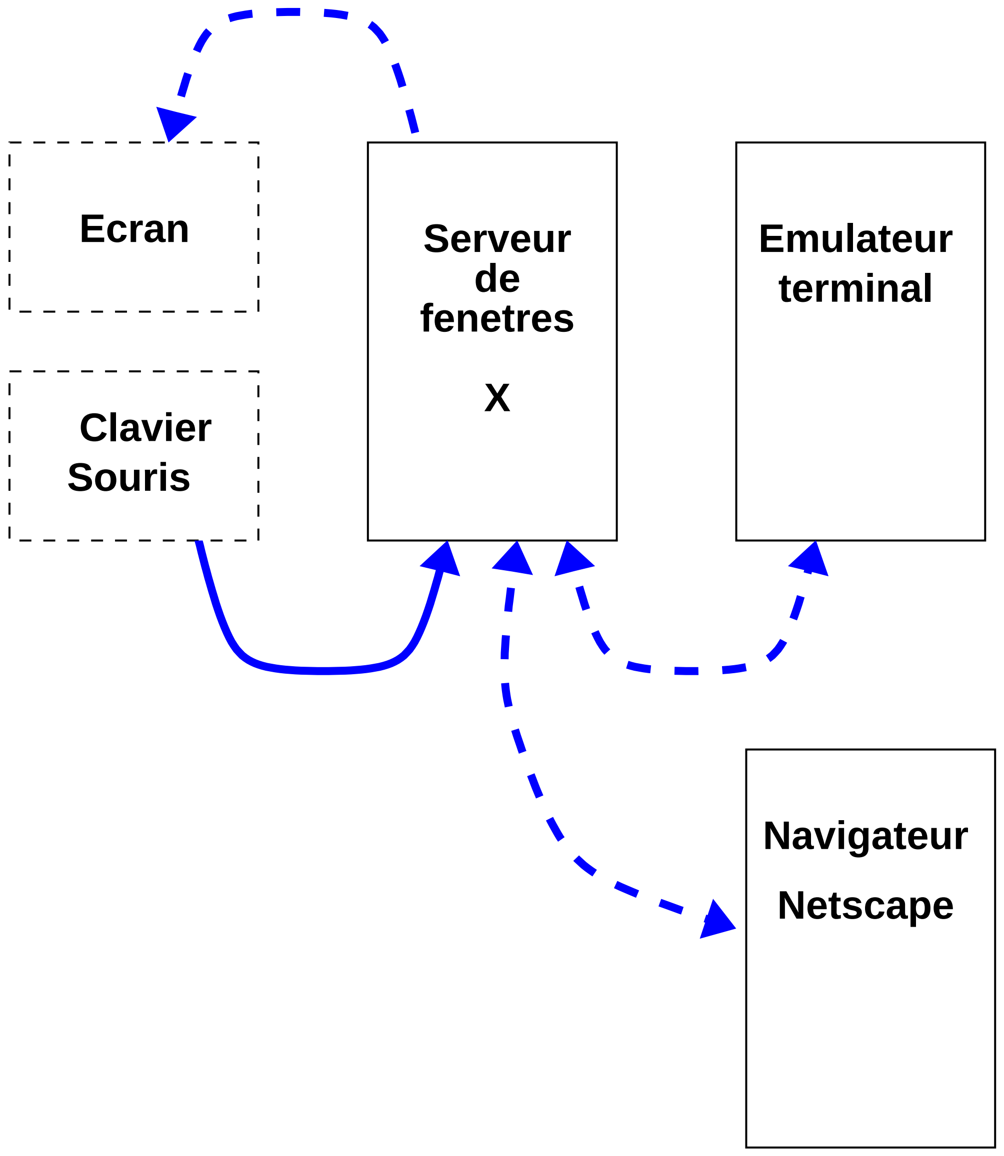
Remarque: événements asynchrones

Matériel: ordinateur, processeur
Logiciel: système

Clavier – Serveur de fenêtres – Application



Clavier – Serveur fenêtres – Deux applications



Communication serveur de fenêtres → application

1. Le serveur de fenêtres (X-window) prend la main et range les événements dans la queue des événements d'entrée du processus propriétaire de la fenêtre où ces événements se sont produits.

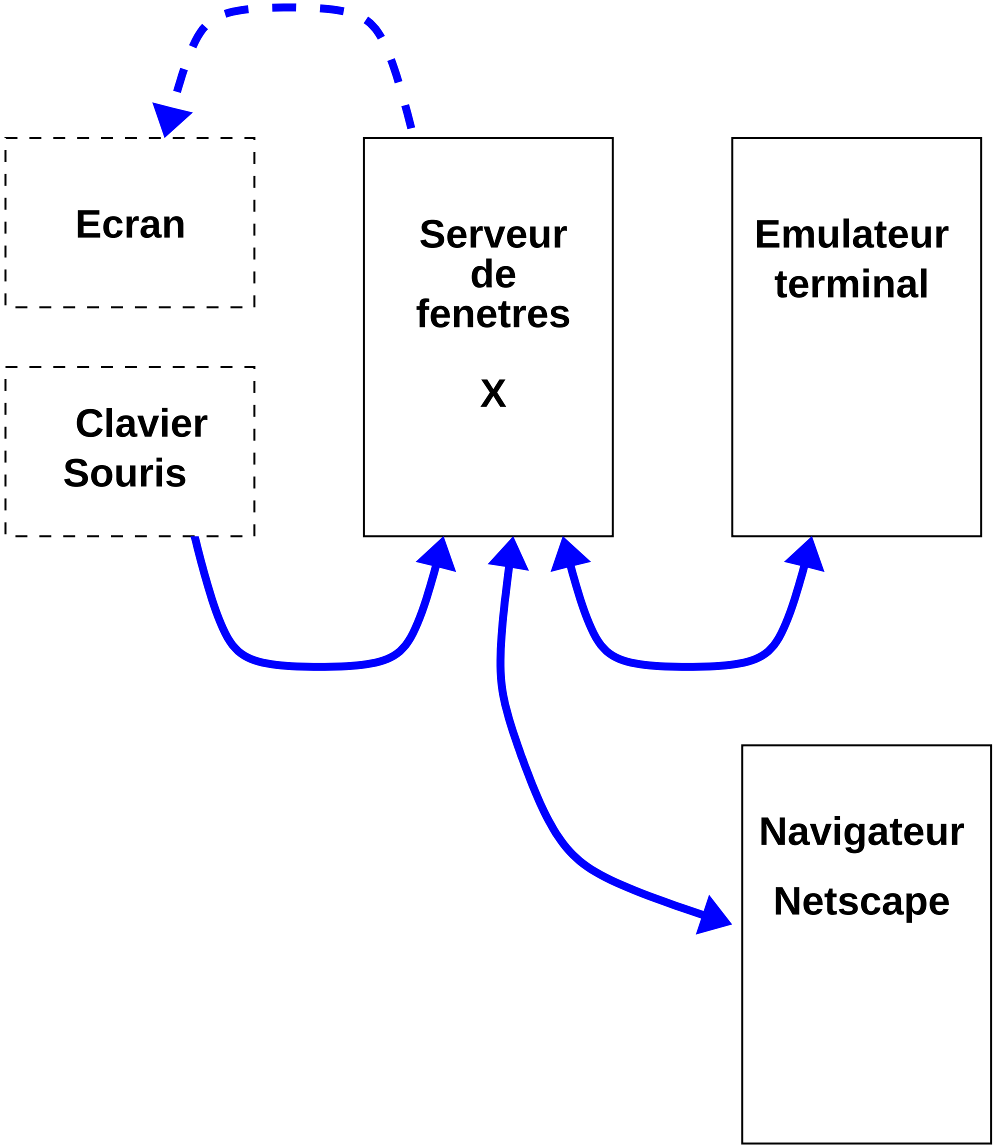
2. Le processus propriétaire de la fenêtre (`xterm`, `emacs`, `netscape`) prend la main et traite ses événements, par exemple en affichant sur l'écran le caractère contenu dans l'événement.

Remarques:

- 2 processus actifs: le serveur de fenêtres, l'application
- Les événements contiennent les coordonnées du curseur pour retrouver la fenêtre propriétaire de ces événements.

Matériel: écran
Logiciel: serveur X-window, processus

Clavier – Serveur fenêtres – Deux applications



Le système de fenêtres

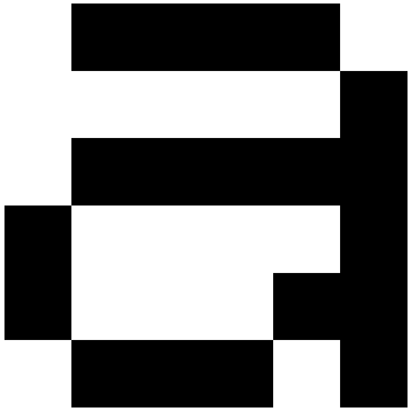
1. L'application envoie l'ordre au serveur de fenêtres d'afficher le caractère A aux coordonnées (x, y) de la fenêtre n.
2. Le serveur prend la main et affiche le caractère, en copiant la matrice de points correspondant à la lettre A dans la mémoire vidéo de l'écran bitmap

Remarque:

- il peut y avoir plusieurs ordres d'affichage à traiter en parallèle. Le serveur de fenêtres est responsable de la synchronisation des opérations sur l'écran.

Matériel: écran bitmap, mémoire vidéo
Logiciel: synchronisation

Opérations *bitmap* – graphique interactif



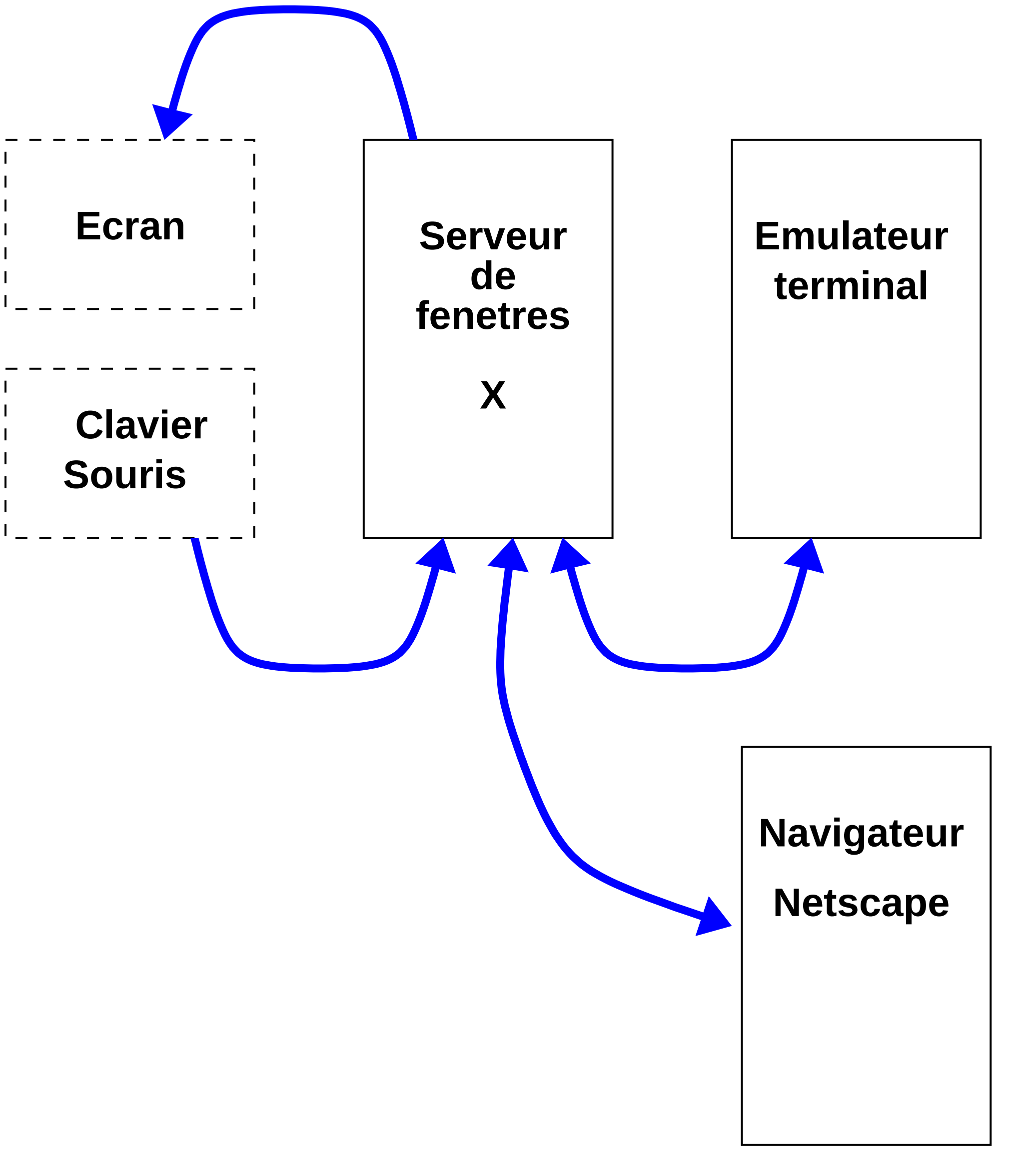
0	1	1	1	1	0	30
0	0	0	0	0	1	1
0	1	1	1	1	1	31
1	0	0	0	0	1	33
1	0	0	0	1	1	35
0	1	1	1	0	1	29

1. **L'écran est un grand tableau de points (bits, octets, mots de 24 bits). Comment copier efficacement un tableau de points dans un sous-tableau de l'écran? (bit-blt)**

2. **Comment tenir compte des parties cachées et des bords des fenêtres? (clipping)**

Matériel: bit, octet, mot
Logiciel: *clipping*, *bit-blt*, *raster-op*, **bresenham**, *splines*, **etc.**

Clavier – Serveur fenêtres – Deux applications (fin)



Applications indépendantes du système de fenêtres

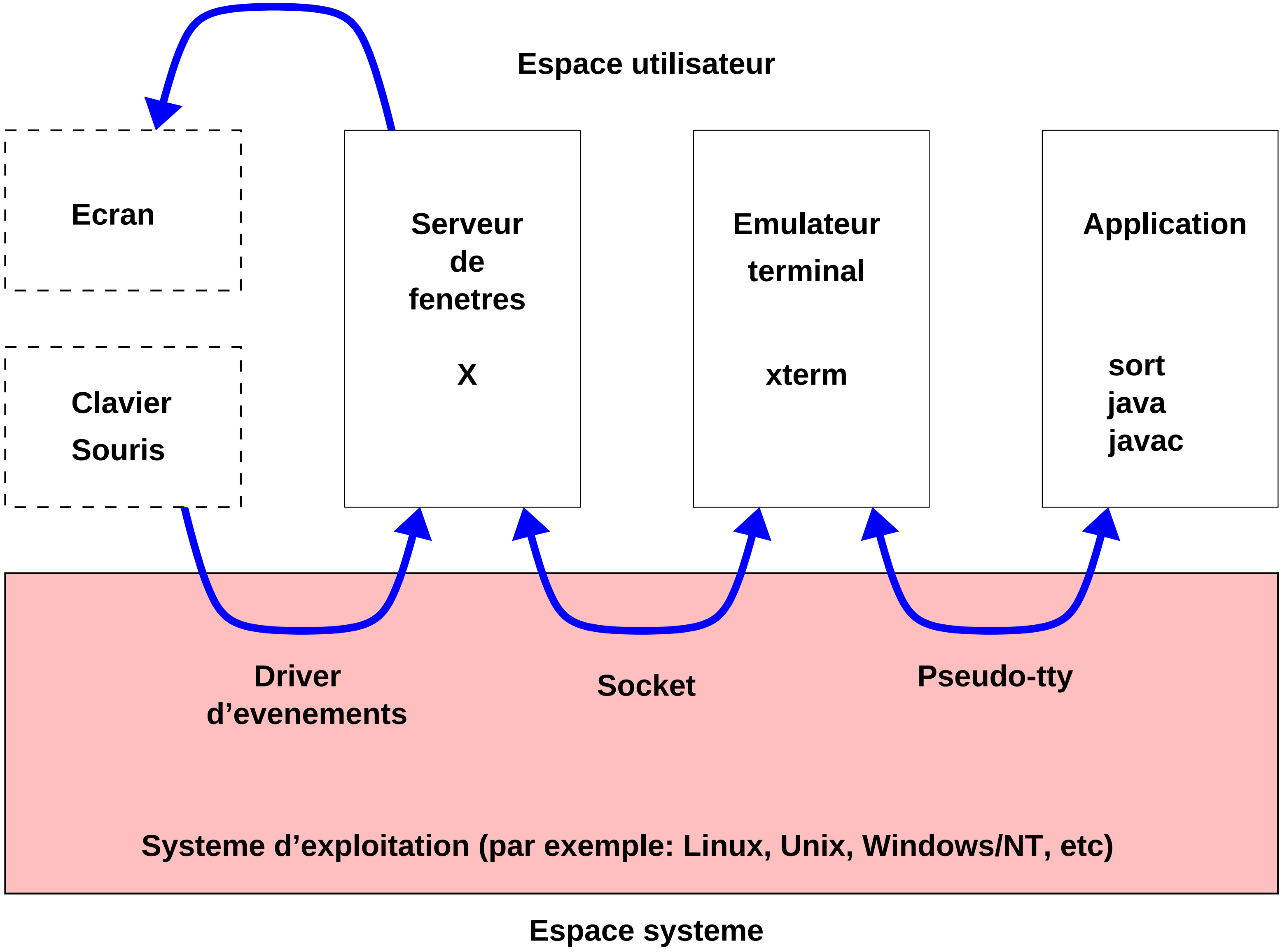
1. Beaucoup d'applications veulent lire ou écrire des caractères sans se soucier du système de fenêtres qui affiche ces caractères.

2. L'émulateur terminal (`xterm`) sous X-window émule un terminal VT100 en fournissant un pseudo *tty* (*terminal typewriter*) à l'application.

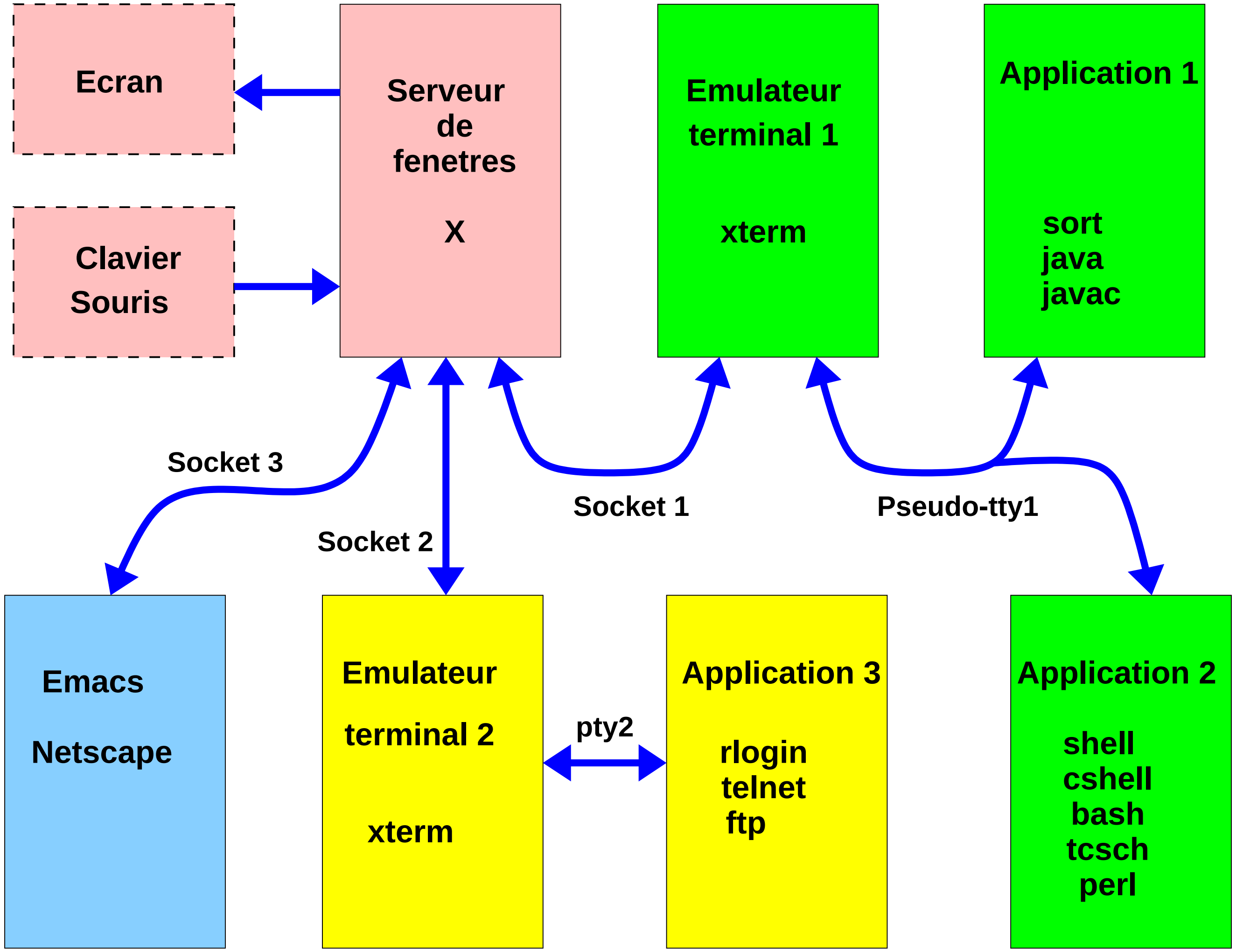
3. Les applications lisent et écrivent des caractères sur ce pseudo tty. Les applications deviennent portables; elles peuvent tourner sous Unix, Linux, MacOS, Windows/NT, VMS, Windows, etc.

Matériel:
Logiciel: pseudo-tty, portabilité

Espace utilisateur – Espace système



Applications multiples – Multi utilisateurs



Applications indépendantes de leur localisation sur le réseau

1. les connexions (*sockets*) d'Unix peuvent relier 2 applications distantes sur le réseau.

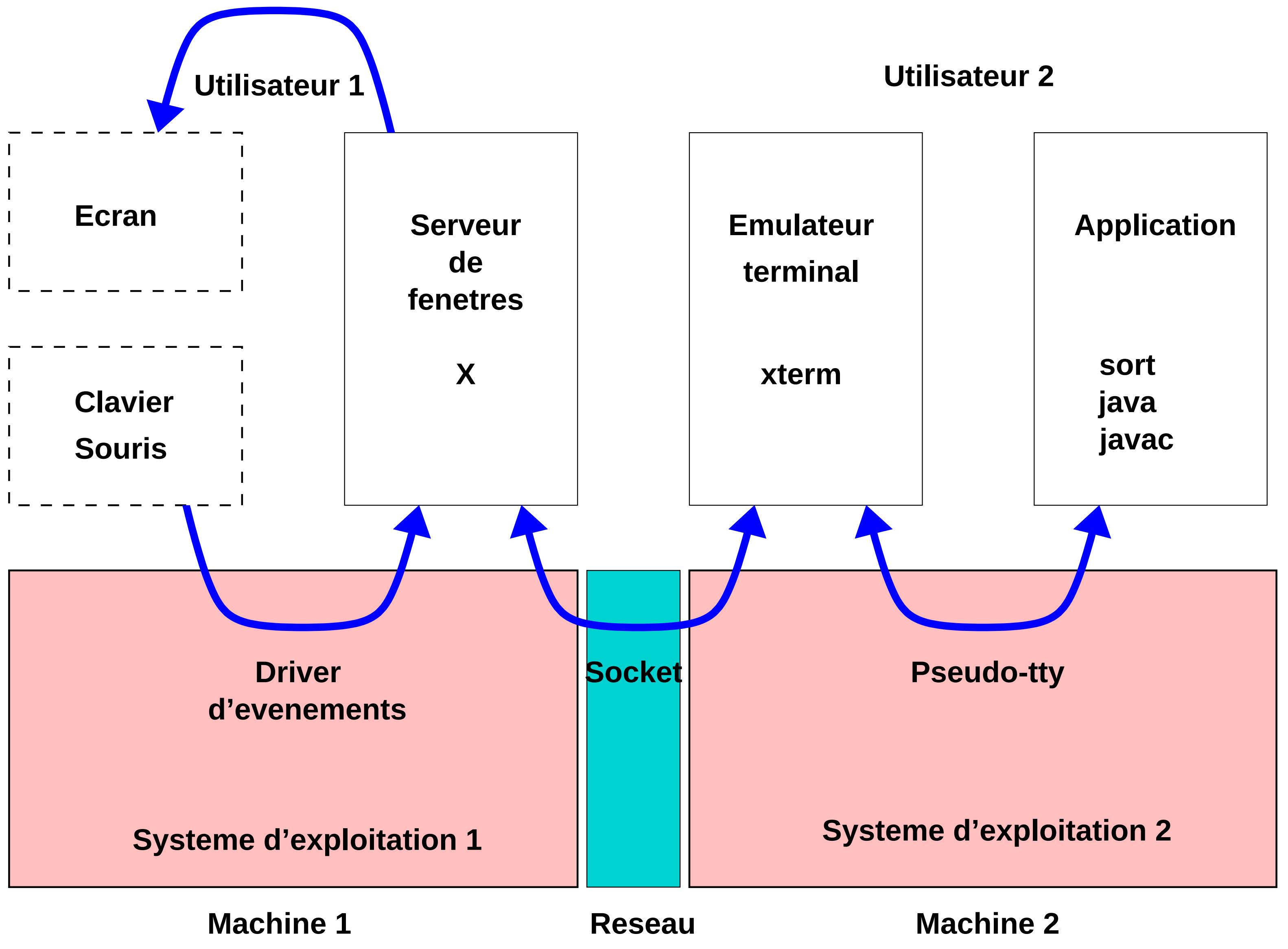
2. serveur de fenêtres et applications peuvent donc se trouver sur des machines différentes.

3. sur la figure précédente, chaque couleur correspond à une machine potentiellement différente.

4. pour désigner un serveur de fenêtres on utilise la variable d'environnement `DISPLAY` (cf. plus tard)

Matériel: réseau
Logiciel: portabilité

Espace utilisateur – Espace système



Quelques leçons

- Résoudre les problèmes par petites briques (modularité)
- Faire du logiciel indépendant du système (portabilité)
- Plusieurs utilisateurs sur une machine (multi-utilisateurs)
- Isoler les petites parties critiques (serveur de fenêtres, système d'exploitation)

Système d'exploitation (*Operating System*, OS)

Noms: Unix, Linux, MacOS, Windows/NT, Windows, DOS, etc.

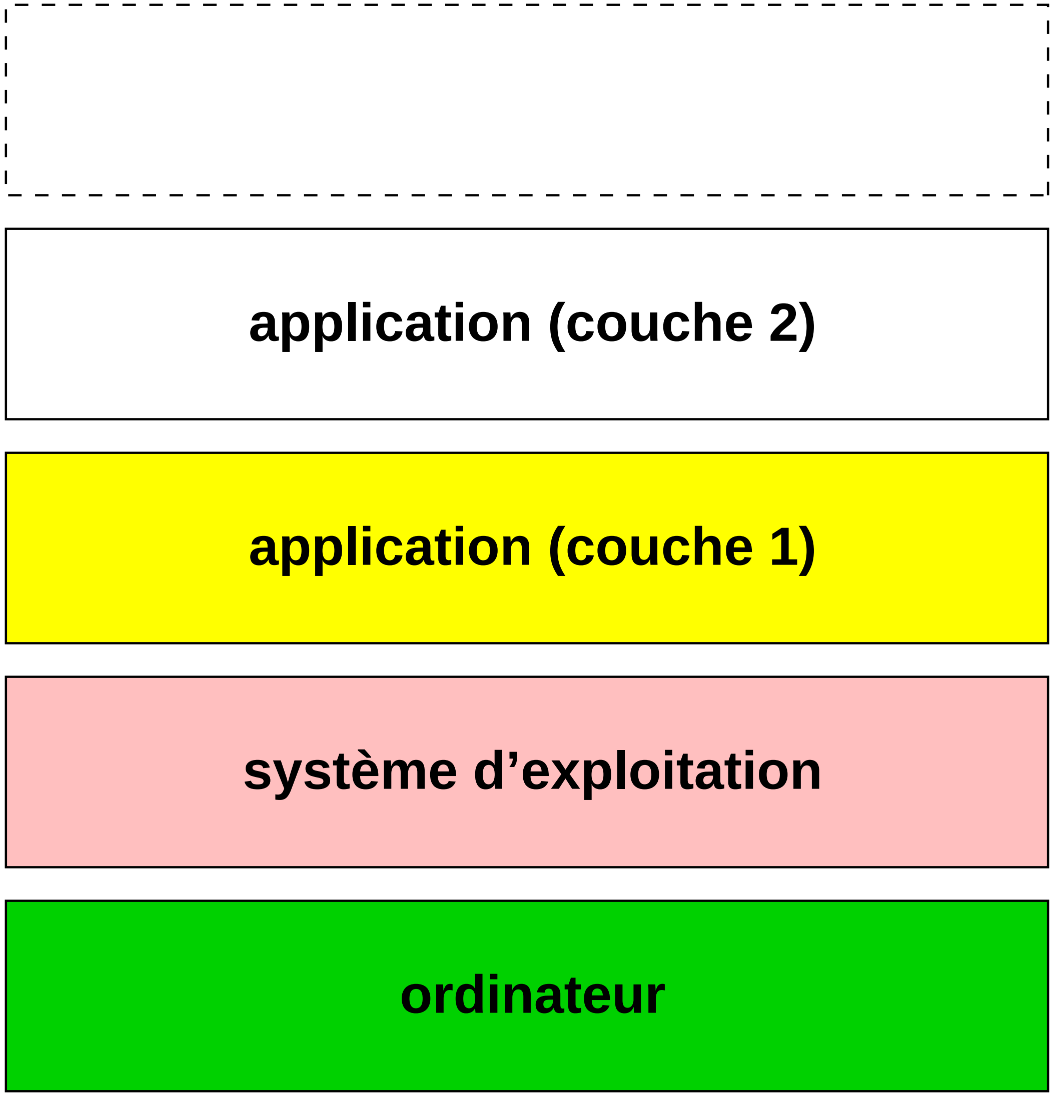
Fonctionalités:

- gérer l'exécution des processus (les programmes qui s'exécutent)
- gérer les données: fichiers, répertoires (sur disques, disquettes, cd-rom, bandes, réseau, etc)
- gérer plusieurs utilisateurs sur une même machine (éventuellement logés à distance)

Mais aussi:

- assurer quelques services (la poste, les forums, la navigation sur le Web)

Les différentes couches d'un système informatique



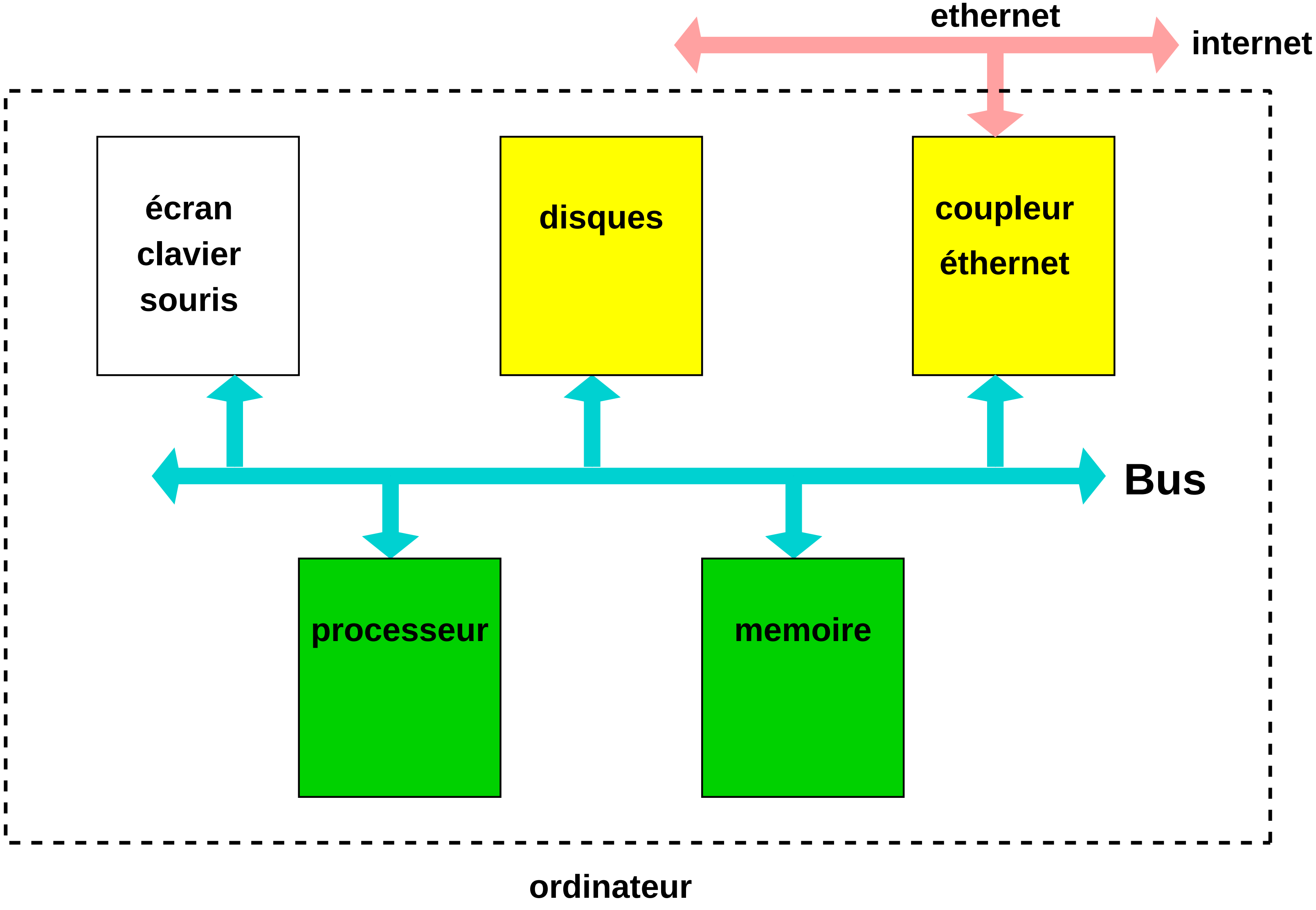
Matériel moderne

- processeurs (500 Mips)
- mémoire (30 ns accès, 500 mégaoctets)
- disques (20 gigaoctets)
- ethernet (100 mégabits/sec)
- clavier (5 caractères / sec)

Remarques:

- ordinateurs sont très rapides ($>$ Kasparov)
- ordinateurs sont programmés (si on change la règle du cavalier aux échecs, Deep Blue ne sait plus jouer!)

Architecture d’un ordinateur (von Neumann 1941)



Représentation des scalaires

- **entiers: complément à 2 sur n bits ($n = 8, 16, 32, 64$)**
  $-2^{n-1} \leq x < 2^{n-1}$ **(arithmétique modulo 2^n)**
  $s = 0 \Rightarrow$ **nombre positif v**
  $s = 1 \Rightarrow$ **nombre négatif $v - 2^{n-1}$**
  $2^{31} = 2\ 147\ 483\ 648$
  $2^{63} = 9\ 223\ 372\ 036\ 854\ 775\ 808$

s	v

- **réels: flottant sur 32 ou 64 bits (norme IEEE 754)**
  mantisse m sur 24 ou 53 bits, exposant e sur 8 ou 11 bits
  $x = m\ 10^e$

m	e

- **caractères: 16 bits (en Java)**
  codage Unicode

- **booléens: 1 bit (souvent plus)**
  vrai, faux

Le beau matériel de l'X

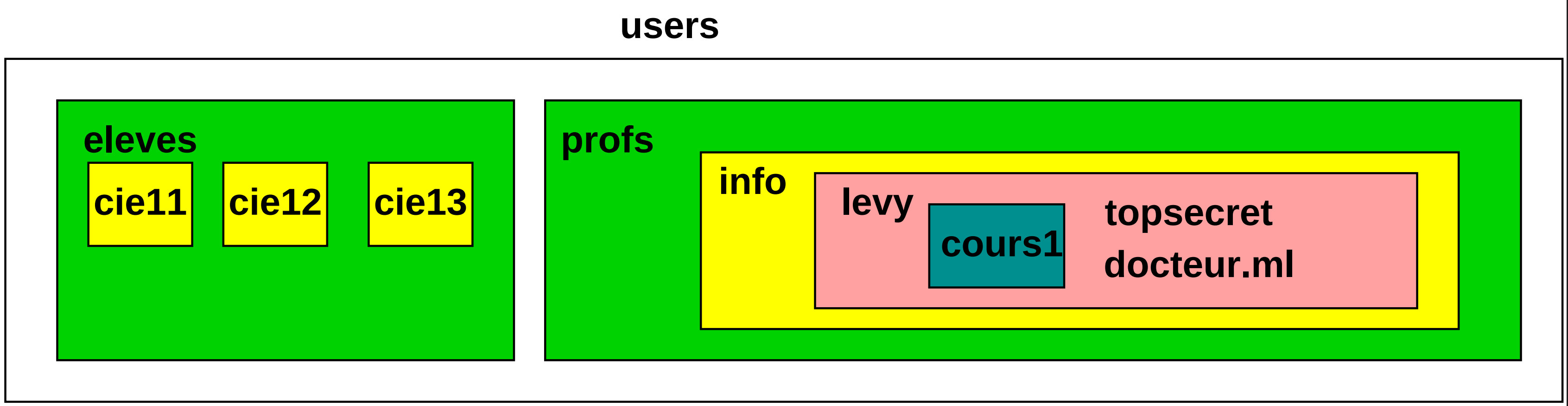
- **100 postes de travail, dont:**
  - **salle 32: 24 PC Linux (noms d'orchidées:** `isotria.polytechnique.fr`**)**
  - **salle 33: 24 PC Linux (noms de départements:** `moselle.polytechnique.fr`**)**
  - **salle 35: 24 PC Linux (noms d'os:** `cubitus.polytechnique.fr`**)**
- **2 serveurs fichiers:** `dalton`, `poly`
- **2 serveurs communication (Web, internet):** `sil`, `kelen`
- **1 serveur mémoire:** `spirou`
- **réseau Ethernet (10 Mega bits/sec) dans les chambres**

Commandes dans une fenêtre terminal

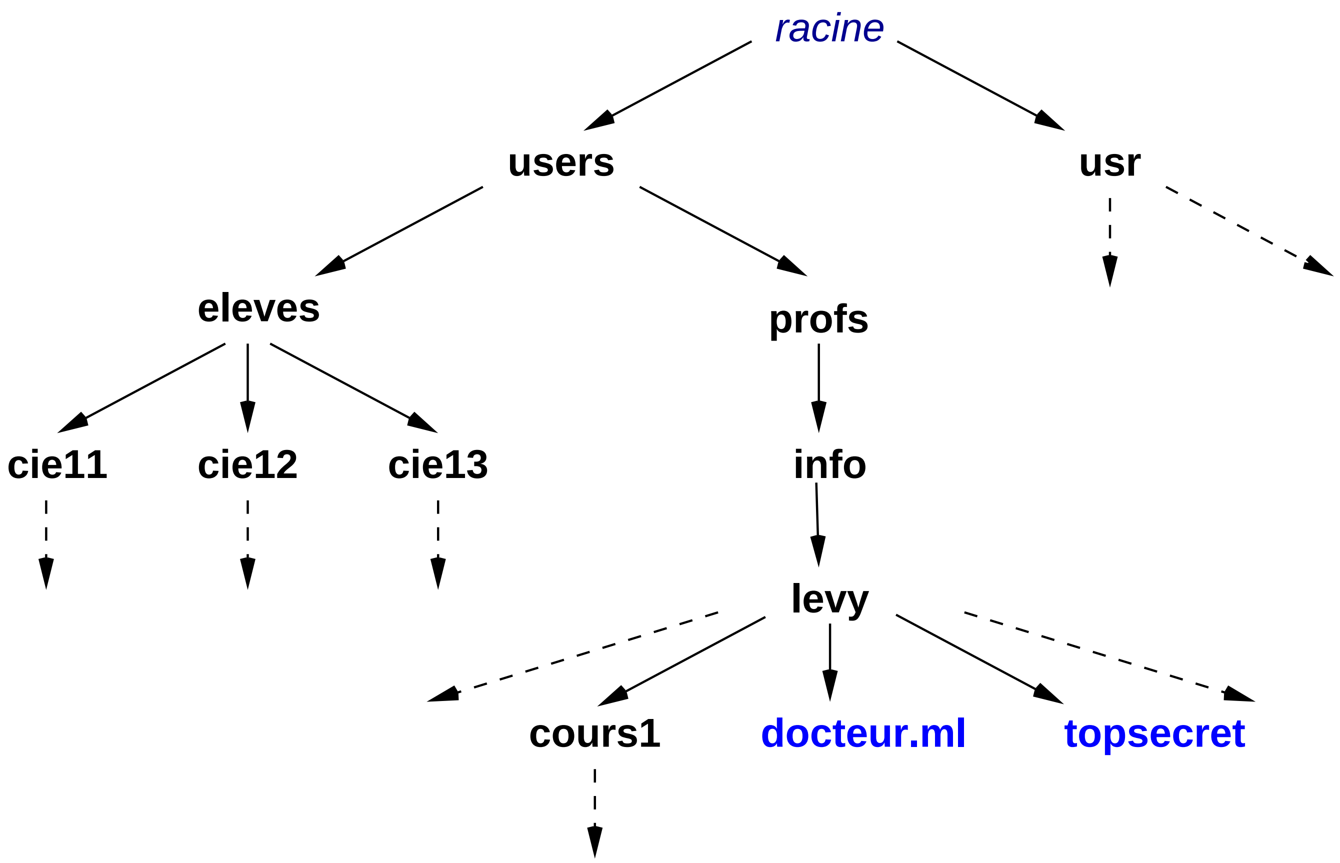
- `passwd` **pour changer de mot de passe**
- `tty` **donne le nom du** *pseudo tty* **utilisé**
- `mail Henri.Poincare@polytechnique.fr` **pour envoyer du courrier à Henri Poincaré à Polytechnique.**
- `mail` **pour lire son courrier**
- `slogin cubitus.polytechnique.fr` **pour se loger à distance sur la machine cubitus.polytechnique.fr**
- `hostname` **pour connaitre le nom de sa machine**
- `emacs bonjour.java` **pour éditer le fichier bonjour.java**
- `netscape&` **pour naviguer sur le** *World Wide Web*
- `ps aux` **pour lister tous les processus actifs**
- `kill -9 3228` **tue le processus 3228**

Beaucoup de ces commandes ont des raccourcis graphiques.

Système de fichiers



Système de fichiers



Notation absolue: `/users/profs/info/levy/topsecret`

Notation par nom d'utilisateur: `~levy/topsecret`

Commandes sur les fichiers

- `ls` **liste les fichiers et répertoires (*list*)**
- `cd cours1` **va dans le répertoire cours1 (*change directory*)**
- `cd ..` **va dans le répertoire englobant**
- `pwd` **imprime le répertoire courant (*print working directory*)**
- `more bonjour.java` **affiche le contenu du fichier bonjour.java**
- `chmod +r bonjour.java` **donne à tous le droit d'accès en lecture du ficher bonjour.java (*change mode*)**

Les fichiers sont ce qu'il reste lorsque l'électricité est coupée (Données persistantes).

La programmation, mère de l'informatique

Dorénavant, tout le monde écrit des programmes ($\geq$ calculette)

La programmation est:

- une activité précise, rigoureuse, comme les math, avec une sanction immédiate: le bon fonctionnement de l'exécution.
- consommatrice de temps. On y arrive si on utilise les bonnes méthodes qui s'apprennent.
- esthétique, si on fait les bonnes abstractions, comme les math!

Langages de programmation

- instructions machines (binaire)
- assembleur (isomorphe au langage machine)
- langage C (typage faible) $\rightarrow$ programmation système
- Pascal (typé)
- ML (Caml, Ocaml) (typé, fonctionnel, esthétique, efficace)
- Java (typé, simplification de C++)

Java très populaire, car de facto le langage du Web. Il existe dans tous les navigateurs du monde (Netscape, Internet Explorer).

Java est orienté-objet, mais nous n'utiliserons que peu cette caractéristique.

Java très bien vendu par Sun microsystems, missile contre Microsoft.