

# Connaissances de base de l'informatique

David Janiszek

[David.Janiszek@parisdescartes.fr](mailto:David.Janiszek@parisdescartes.fr)

<http://www.mi.parisdescartes.fr/~janiszek/>

Université Paris Descartes  
U.F.R. de Mathématique et d'Informatique



31 Aout 2016

# Les principaux objectifs du module

- Présenter l'environnement de travail mis à votre disposition
  - Être autonome le plus rapidement possible
- Présenter les outils essentiels d'UNIX
- Introduire quelques concepts importants de l'informatique
- Ouverture sur l'informatique professionnelle

# L'équipe pédagogique

- David Janiszek (responsable)
- Nicolas Lomenie
- Azedine Mani
- Thierry Raedersdorff

# Nous contacter

- N'hésitez à nous poser des questions mais  
vérifiez d'abord que votre question n'est pas dans les supports
- En dehors des séances, contactez nous par courriel  
Nomenclature : prenom.nom@parisdescartes.fr  
⇒ Ajoutez le tag [CBI] au sujet sinon pas de réponse !
- Beaucoup d'informations sont données à l'oral en présentiel  
⇒ soyez présents aux séances de TP

# L'organisation du module

- Le module compte pour 3 ECTS
  - il faut 30 ECTS pour valider un semestre
  - 1 ECTS = 25 heures de travail (travail encadré et personnel inclus)
- Le module est organisé en 4 TP de 3H30
  - c'est peu ! être absent vous pénalise beaucoup
- Les TP sont répartis sur les deux premières semaines du S1
  - c'est rapide ! il faut travailler dès maintenant
- Les TP sont à réaliser en binôme
  - mais l'évaluation finale est individuelle
- L'évaluation du module prendra la forme d'un examen

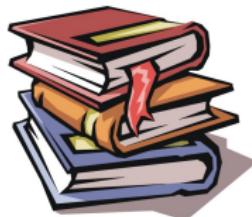
# L'évaluation

- L'évaluation est réalisée par un examen d'une durée de 45 minutes
- La date et le lieu de l'examen sont disponibles sur la page du cours
- L'examen prend la forme d'un QCM comprenant 40 questions portant sur les 4 TP réalisés
  - le temps par question est court
  - aucun document n'est autorisé
  - aucune sortie pendant l'épreuve n'est autorisé
  - une deuxième session aura lieu au mois de janvier (max)

# La notation

- Barême du QCM de l'examen
  - proposition juste cochée = 1 ;
  - proposition non cochée = 0 ;
  - proposition erreur cochée = -1 ;
- ⇒ Ne cocher que si vous êtes sûr sinon vous perdrez des points
- ⇒ 50 % de bonnes réponses + 50% de mauvaises réponses → 0/20
- Taux de réussite : <30 % en première session !

# Les ressources

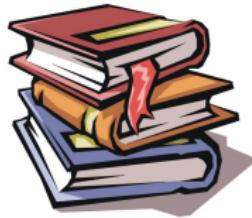


- Le site Web du module

- information sur l'organisation, l'emploi du temps, etc.
- accès aux cours
- accès à tous les sujets de TP
- une FAQ Unix (Foire Aux Questions)
- une liste de liens utiles pour aller plus loin
- <http://www.ens.math-info.univ-paris5.fr/cbi/>

- Twitter : @updcbi

# Les ressources (suite)



- Des références bibliographiques

- Trouvez le livre qui vous convient !
- Exemples de livres de cours
  - *Unix : Guide de l'étudiant.* Harley Hahn. Dunod 2nde édition, 1998.
  - *Unix et Linux – Utilisation et administration.* Lean-Michel Lévy. Pearson Education, 2007.

- Exemples de livres d'exercices

- *Linux – Initiation et utilisation.* Jean-Paul Armspach, Pierre Colin, et Frédérique Ostré-Waerzeggers. Dunod, 2nde édition, 2004.

# Comment réussir ce module ?

## Les règles du succès

### Régle n°1 : Assistez à tous les TPs

- En cas d'absence rattrapez le TP rapidement
- Une absence non rattrapée  $\implies$  un décrochage assuré
- Difficulté croissante

### Régle n°2 : Lisez le sujet de TP avant la séance encadrée

- Vous gagnerez beaucoup de temps

### Régle n°3 : Posez des questions !

- Les enseignants sont là pour y répondre

### Régle n°4 : Ne laisser pas passer un mot ou un concept sans le comprendre

- Demandez systématiquement à un enseignant de TP

# Comment réussir ce module ? (suite)

## Les règles du succès

### Régle n°5 : Relisez votre TP le soir après l'avoir fait

- Profitez de la séance suivante pour poser vos questions

### Régle n°6 : Pratiquez un maximum

- Pour assimiler toutes les notions abordées dans les TP ; provoquez des situations de mise en pratique.

### Régle n°7 : Soyez curieux

- Pour aller au delà du cours et consolider ce que vous apprenez.

### Régle n°8 : Formez un petit groupe de travail

- Pour éviter le découragement et l'isolement (On est plus fort à plusieurs)

# Pendant ce temps là ...

de l'autre côté de la planète ...



Beaucoup d'exercices



Tous les étudiants

C'est aussi le cas dans les classes préparatoires !  
⇒ un seul marché du travail !

# Pourquoi UNIX ?

# Pourquoi UNIX ?

- Système d'exploitation créé en 1969
- UNIX et ses descendants :
  - Linux (Open-source) → donc gratuit !
    - Android
  - BSD
    - NetBSD (Open-source)
    - FreeBSD (Open-source)
    - OpenBSD (Open-source)
    - Mac OS X
    - iOS
  - Solaris, AIX, HP\_UX, Unixware ... (Payant)

# Pourquoi UNIX ? (suite)

- La famille de système d'exploitation la plus utilisée au monde (statistiques de 2014)
  - Supercalculateurs : 97 % de Linux + 2,8 % d'UNIX
  - Serveurs web : 97 % de Linux + 1,6 % de BSD
  - Tablettes : 62 % d'Android, 36% d'iOS
  - Smartphones :49 % d'Android, 11% d'iOS
  - Console de jeu : 29,6 % de BSD (PS3,PS4)
  - Mais aussi : SmartTV, Box internet, matériel réseau (routeur), voitures, ...
- Pour information :
  - Ordinateurs perso. : 1,3 % de Linux + 7,2 % de BSD (Mac OS X)

⇒ Largement utilisé par les professionnels

# Un peu d'histoire...

# Les débuts de l'informatique moderne

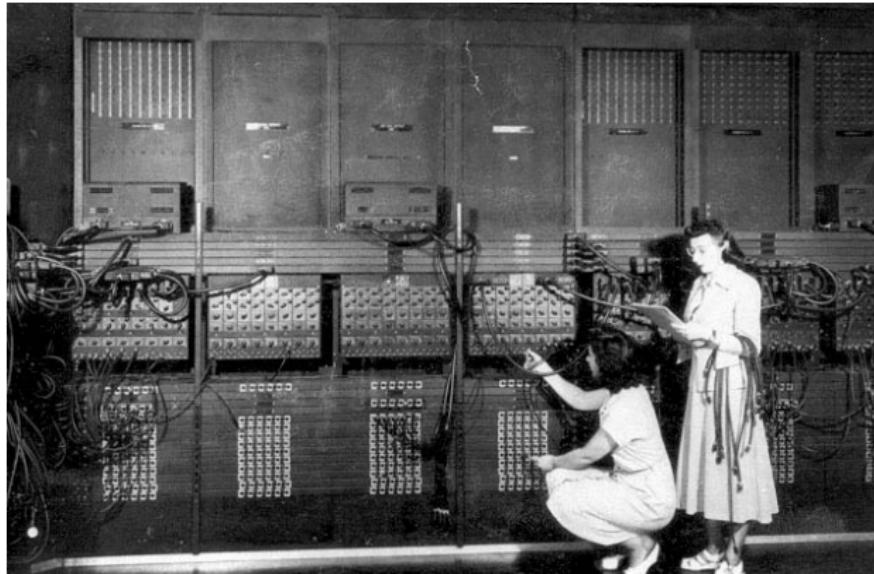
Ordinateur	Date	Pays	Numération	Électronique	Programmable
Zuse Z3	1941	Allemagne	binnaire	non	par carte
Atanasoff-Berry Computer	1941	Etats-Unis	binnaire	oui	non
Colossus	1943	Royaume-Uni	binnaire	oui	par recablage
IBM ASCC	1944	Etats-Unis	décimal	non	par carte
ENIAC	1944	Etats-Unis	décimal	oui	par recablage

TABLE: Les premiers ordinateurs numériques

# Les débuts de la programmation ...

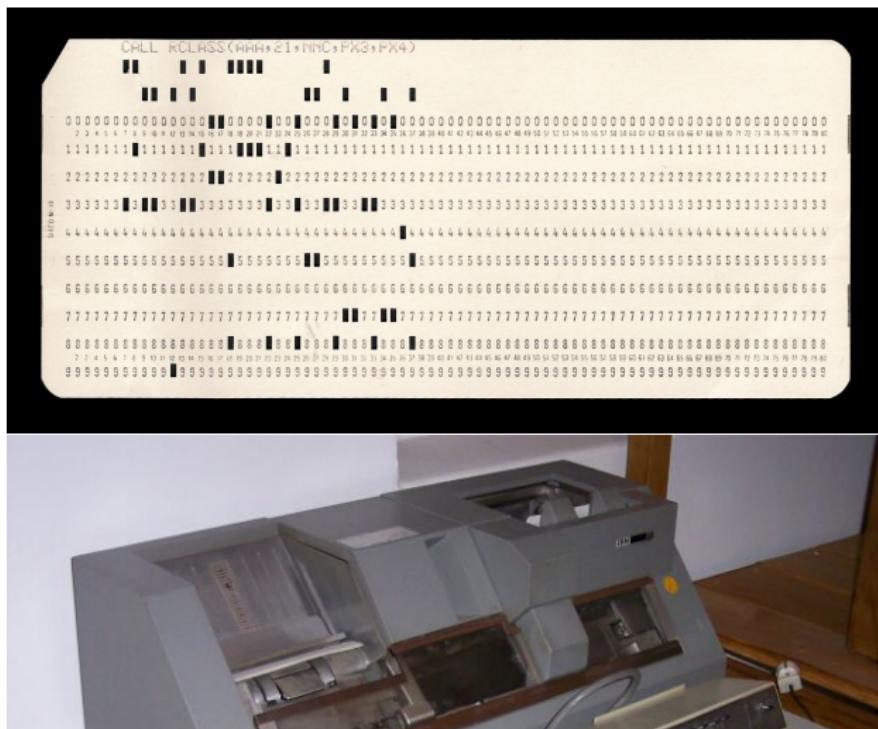
Programmation d'un des premiers ordinateurs électroniques :

Electronic Numerical Integrator Analyser and Computer (ENIAC)



# Les débuts de la programmation ... (suite)

Programmation à l'aide de carte perforée :



# Aux débuts de l'informatique ...

- Un ordinateur pour effectuer une tâche
- L'architecture de l'ordinateur est le programme
- Problème : il faut modifier ou construire un nouvel ordinateur lorsque la tâche évolue !

# Les premiers programmes

Tâche	Nom	Année
Compilateur (langage A-0)	×	1951
Prévision météorologique	×	1955
Composition musicale	×	1956
Editeur de texte (plein écran)	O-26	1967
Traitemennt de texte	MT/ST	1969
Tableur	LANPAR	1969

TABLE: Quelques-uns des premiers logiciels

# Les débuts de l'informatique personnelle...

Le premier micro-ordinateur basé sur un microprocesseur :  
Micral (1972) de R2E (réalisé par des français)



# Les débuts de l'informatique personnelle...(suite)

Le premier micro-ordinateur complet :  
Apple I (1976->1977)



# Les débuts de l'informatique personnelle...(suite)

Son successeur :

Apple II (1977->1988)



# Les débuts de l'informatique personnelle...(suite)

Le premier Personal Computer (PC) :  
IBM PC (1981)



# Les débuts de l'informatique personnelle...(suite)

Le premier ordinateur grand public avec interface graphique et souris :

Lisa (1983) de Apple



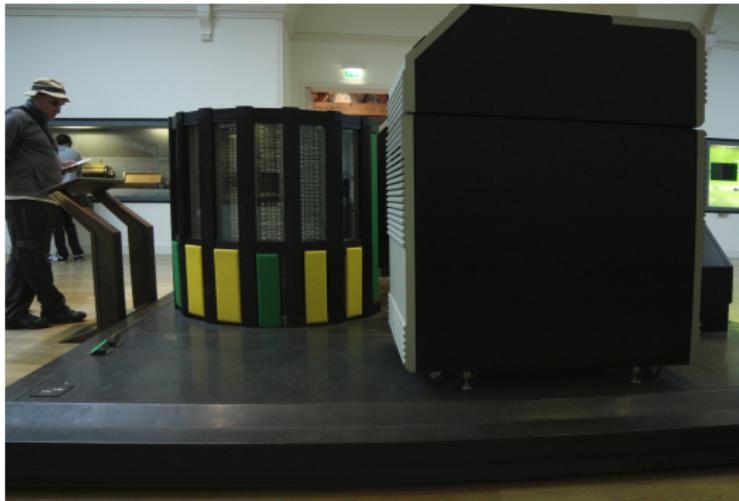
# Les débuts de l'informatique personnelle...(suite)

Le premier ordinateur familial :  
ZX80 (1980) de Sinclair



# Super-calculateur

Le CRAY-2 (1985) de Cray Inc. est le 1er ordinateur à dépasser 1Gflops



# Super-calculateur (suite)

Processeur : 283Mhz - Mémoire : 4 Go - Poids : 2,5 tonnes



# 1Gflops c'est ...

Apple Power Mac G4 (1999)



# 1Gflops c'est ... (suite)

Processeur : 400Mhz - Mémoire : 128 Mo - Poids : 13 kilos



# 1Gflops c'est ... (suite)

Apple iPhone 3GS (2009)

Processeur : 600 Mhz - Poids : 135g



# Aujourd'hui

Supercalculateur Curie au CEA (Bull)

26ème rang mondial (top500.org)

77184 processeurs Xeon E5

2098 Tflops



# Remerciements

J'adresse mes remerciements à :

- Damien Pellier
- Julie Mauclair