

<http://licence.ifsic.univ-rennes1.fr>  
<http://maitrise.ifsic.univ-rennes1.fr>

# L I C E N C E M A Î T R I S E

L I C E N C E  
E T M A Î T R I S E D ' I N F O R M A T I Q U E

## P R O G R A M M E D E S É T U D E S



# SOMMAIRE

|  |          |
|--|----------|
| <b>PRÉSENTATION GÉNÉRALE</b>   | <b>1</b> |
| Objectif.....  | 1        |
| Enseignements.....   | 1        |
| Architectures des ordinateurs et des systèmes  | 1        |
| Langages, algorithmique et programmation   | 2        |
| Mathématiques pour l'informatique  | 2        |
| Responsables de la formation.....  | 2        |
| <b>PROGRAMME DES ÉTUDES</b>  | <b>2</b> |
| Licence d'Informatique.....  | 2        |
| Maîtrise d'Informatique.....   | 3        |
| <b>ADMISSION</b>   | <b>4</b> |
| Candidature .....  | 4        |
| Diplômes requis  | 4        |
| Dossiers   | 5        |
| Statistiques .....   | 5        |
| <b>MATIÈRES ENSEIGNÉES</b>   | <b>6</b> |
| LICEINFO / ALG1 : Algorithmique des graphes  | 7        |
| LICEINFO / ALG2 : Méthodes algorithmiques  | 8        |
| LICEINFO / ARC1 : Architecture et systèmes : architecture 1                                    | 9        |
| LICEINFO / SYS1 : Architecture et systèmes : système 1   | 10       |
| LICEINFO / ARC2 : Architecture et systèmes : architecture 2                                    | 11       |
| LICEINFO / SYS2 : Architecture et systèmes : système 2   | 12       |
| LICEINFO / ANG : Anglais   | 13       |
| LICEINFO / BD : Bases de données   | 14       |
| LICEINFO / COMP : Introduction à la compilation  | 15       |
| LICEINFO / NET : Initiation aux services Internet  | 16       |
| LICEINFO / PROJ : Projet   | 17       |
| LICEINFO / LF : Langages formels   | 18       |
| LICEINFO / LOG : Logique   | 19       |
| LICEINFO / PRG1 : Programmation 1  | 20       |
| LICEINFO / PRG2 : Programmation 2  | 21       |
| MAITINFO / ADT : Analyse, développement et test  | 22       |
| MAITINFO / OBJ : Programmation objet et parallèle  | 23       |
| MAITINFO / SYS1 : Systèmes d'exploitation 1 : Gestion du parallélisme dans les systèmes d'ex-  |          |
| ploitation   | 24       |
| MAITINFO / COMP : Compilation  | 25       |
| MAITINFO / AR : Option 1 : Automates et réseaux  | 26       |
| MAITINFO / MFA : Option 1 : Modélisation par files d'attente                                   | 27       |
| MAITINFO / PLC : Option 1 : Programmation logique et par contrainte                            | 28       |
| MAITINFO / MFGL : Méthodes formelles pour le génie logiciel                                    | 29       |
| MAITINFO / SYS2 : Systèmes d'exploitation 2 : Gestion des informations dans les systèmes d'ex- |          |
| ploitation   | 30       |
| MAITINFO / REPR : Réseaux et protocoles  | 31       |
| MAITINFO / ALGO : Option 2 : Algorithmique distribuée  | 32       |
| MAITINFO / IMA : Option 2 : Images numériques  | 33       |
| MAITINFO / IA : Option 2 : Intelligence artificielle   | 34       |
| MAITINFO / PROJ : Méthodes de génie logiciel   | 35       |
| MAITINFO / ANG : Anglais   | 36       |



# **PRÉSENTATION GÉNÉRALE**

## **Objectif**

La filière Licence-Maîtrise d'informatique de l'Université de Rennes 1 constitue une formation professionnelle en conception, réalisation, maintenance de logiciels d'une part, administration de systèmes et de réseaux informatiques d'autre part.

Un prolongement de cette formation à travers un approfondissement des connaissances informatiques et un stage facilitant l'insertion dans le monde du travail peut être obtenu au sein des DESS Isa et Mitic (ouverture prévue à la rentrée 2001-2002) de l'Université de Rennes 1. La formation à et par la recherche (DEA et doctorat d'informatique) constitue également une poursuite de formation possible et naturelle après la maîtrise.

Face à une technologie, des métiers et un marché du travail en évolution permanente, nous voulons privilégier les connaissances fondamentales, qui ont une durée de vie supérieure et fournissent une base solide permettant à l'étudiant de compléter sa formation et de s'adapter.

S'il y a deux diplômes distincts dans la filière, son contenu est conçu pour privilégier deux sorties cohérentes : en fin de maîtrise (Bac+4) et en fin de DESS ou DEA (Bac+5). L'évolution des métiers de l'informatique nous incite à encourager la sortie à Bac+5.

## **Enseignements**

L'enseignement est dispensé sous forme de cours qui donnent le cadre théorique et définissent les idées directrices, de travaux dirigés qui facilitent par des exercices l'assimilation des concepts vus en cours et de travaux pratiques qui permettent d'acquérir le savoir-faire en face des machines et le sens de la mise au point des programmes.

L'enseignement de la filière peut se décomposer selon trois grands axes.

### **Architectures des ordinateurs et des systèmes**

Cette partie traite des différents composants matériels d'un ordinateur, puis de la structure et de la conception des systèmes d'exploitation permettant d'offrir à l'utilisateur un environnement de travail. On

aborde successivement les systèmes mono-utilisateur, les systèmes de gestion de bases de données, les systèmes multiprogrammés et les réseaux d'ordinateurs.

## **Langages, algorithmique et programmation**

On étudie d'une part l'algorithmique et les méthodes de programmation et d'autre part différents paradigmes de langages. On aborde la programmation impérative, fonctionnelle, logique, orientée objet et enfin parallèle. Le but est de donner aux étudiants la possibilité d'apprendre et de comprendre rapidement n'importe quel langage, plutôt que de faire des spécialistes d'un nombre forcément restreint de langages. On approfondit ces enseignements par l'étude de grands domaines d'utilisation de méthodes algorithmiques comme la compilation, l'intelligence artificielle, la synthèse d'images numériques, la compression de données etc.

## **Mathématiques pour l'informatique**

On étudie différents modèles permettant de décrire des problèmes rencontrés en informatique et d'en exprimer les solutions (langages formels, graphes, logique, files d'attente...).

## **Responsables de la formation**

Licence : André COUVERT

Maîtrise : Olivier RIDOUX

# **PROGRAMME DES ÉTUDES**

## **Licence d'Informatique**

La licence d'informatique comporte huit unités d'enseignement semestrielles :

| <b>Unités d'enseignement</b>            | <b>Volume<br/>horaire</b> |
|---|---------------------------|
| <b>U.E. ALGORITHMIQUE DES GRAPHS</b>    | 44                        |
| <b>U.E. ARCHITECTURE - SYSTÈME 1</b>    |                           |
| Architecture 1                          | 48                        |
| Système 1                               | 40                        |
| <b>U.E. LANGAGES FORMELS - LOGIQUE</b>  |                           |
| Langages formels                        | 44                        |
| Logique                                 | 26                        |
| <b>U.E. PROGRAMMATION 1</b>             | 56                        |
| <b>U.E. MÉTHODES ALGORITHMIQUES</b>     | 44                        |
| <b>U.E. ARCHITECTURE - SYSTÈME 2</b>    |                           |
| Architecture 2                          | 38                        |
| Système 2                               | 46                        |
| <b>U.E. PROGRAMMATION 2</b>             | 56                        |
| <b>U.E. ENVIRONNEMENT PROFESSIONNEL</b> |                           |
| Anglais                                 | 34                        |
| Bases de données                        | 30                        |
| Compilation                             | 22                        |
| Initiation aux services Internet        | 2                         |
| Projet Compilation                      | 24                        |
| <b>VOLUME HORAIRE TOTAL</b>             | 554                       |

## Maîtrise d'Informatique

La maîtrise comporte sept unités d'enseignement semestrielles et une unité d'enseignement annuelle :

| <b>Unités d'enseignement</b>                        | <b>Volume<br/>horaire</b> |
|---|---------------------------|
| <b>U.E. 1 : GÉNIE LOGICIEL ET PROGRAMMATION</b>     |                           |
| Génie logiciel (1) : analyse, développement et test | 48                        |
| Programmation objet et parallèle                    | 60                        |
| <b>U.E. 2 : SYSTÈME D'EXPLOITATION</b>              |                           |
| Système d'exploitation (2) : gestion de processus   | 52                        |
| <b>U.E. 3 : COMPILATION</b>                         |                           |
| Compilation   | 48                        |
| <b>U.E. 4 : MATHS POUR L'INFORMATIQUE</b>           |                           |

## ADMISSION

| Unités d'enseignement                                       | Volume horaire |
|---|----------------|
| Automates et réseaux  | 48             |
| Programmation logique et par contraintes                    | 48             |
| Modélisation par files d'attente                            | 48             |
| <b>U.E. 5 : MÉTHODES FORMELLES POUR LE GÉNIE LOGICIEL</b>   |                |
| Génie logiciel (2) : méthodes formelles                     | 60             |
| <b>U.E. 6 : SYSTÈME D'EXPLOITATION RÉSEAUX</b>              |                |
| Système d'exploitation (1) : gestion de mémoire             | 52             |
| Réseaux et protocoles                                       | 60             |
| <b>U.E. 7 : DOMAINES DE RECHERCHE</b>                       |                |
| Algorithmique avancée                                       | 48             |
| Images numériques   | 48             |
| Intelligence artificielle                                   | 48             |
| <b>U.E. 8 : ENVIRONNEMENT PROFESSIONNEL (U.E. annuelle)</b> |                |
| Projet (Travail d'études et de recherche)                   | 48             |
| Anglais   | 36             |
| <b>VOLUME HORAIRE TOTAL</b>                                 | <b>560</b>     |

## ADMISSION

### Candidature

### Diplômes requis

#### Licence

Admission de droit pour les étudiants titulaires d'un Deug Sciences et Technologie, mentions : Mass, Mias ou SM/STPI (avec option informatique). L'admission est également de droit pour les étudiants titulaires d'un Deug Technologie industrielle mention Génie des systèmes si leur formation comporte un enseignement d'adaptation portant sur l'informatique. Pour tous les autres étudiants (titulaires d'un DUT Informatique, d'un BTS Informatique, autre mention de Deug ...) l'admission est soumise à dérogation et nécessite le dépôt d'un dossier de demande de validation d'études.



## Maîtrise

Admission de droit pour les étudiants titulaires d'une Licence Informatique. Pour tous les autres étudiants, l'admission est soumise à dérogation et nécessite le dépôt d'un dossier de demande de validation d'études.

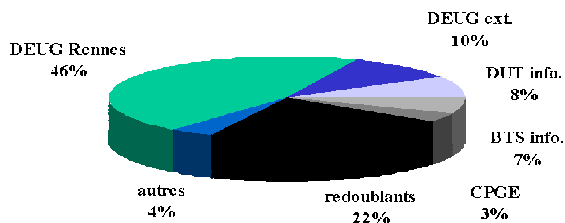
## Dossiers

Les dossiers de demande de validation d'études en vue d'une inscription en licence ou en maîtrise sont à retirer à la scolarité du 2ème cycle au cours du mois de mai et à retourner fin juin. Lors de la demande de dossier à la scolarité, il convient de préciser le diplôme d'origine. La commission de validation se réunit vers la mi-juillet.

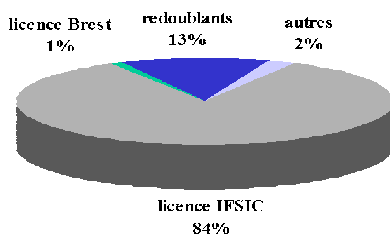
Scolarité 2ème cycle  
Campus de Beaulieu  
35042 Rennes Cedex

## Statistiques

*Répartition des titres d'accès (licence)*



*Répartition des titres d'accès (maîtrise)*



## **MATIÈRES ENSEIGNÉES**

Les différentes matières sont décrites dans les pages suivantes : ne page par matière qui comprend une présentation, un plan, le sigle de la matière et le nombre d'heures d'enseignement.

# Licence informatique

## Algorithmique des graphes

L'objectif de ce cours, dans le cadre de la licence informatique, est double. Il permet, d'une part, l'apprentissage d'un modèle servant de base à de nombreux problèmes (graphes d'états, modèles en intelligence artificielle, réseaux, etc.). Ce modèle utilise des notions fondamentales de mathématiques discrètes (ensembles, relations, inductions) qui sont rappelées. Il contribue, d'autre part, à enrichir le corpus algorithmique : en effet, les graphes sont des objets complexes permettant d'aborder d'intéressantes notions algorithmiques à travers la résolution de problèmes classiques (techniques d'exploration, algorithmes gloutons, complexité, types abstraits, ...).

### PLAN

1. Un peu de mathématiques discrètes : ensembles, relations binaires, inductions
2. Le type abstrait graphe et ses représentations
3. Fermeture transitive
4. Exploration de la descendance d'un sommet
5. Circuits et composantes fortement connexes
6. Chemins de valeur optimale
7. Problèmes d'ordonnement
8. Flots sur les réseaux de transport

# **Licence informatique**

## **Méthodes algorithmiques**

Le but de ce cours est d'introduire différentes méthodes de résolution de problèmes : diviser pour résoudre, essais successifs, recherche d'heuristiques, séparation et évaluation progressive, algorithmes gloutons, programmation dynamique, tris, ... De plus, les complexités des algorithmes, en temps et en place mémoire, sont abordées afin de donner les limitations des diverses méthodes étudiées.

### **PLAN**

1. Introduction à l'algorithmique et à la complexité des algorithmes
2. La méthode diviser pour résoudre et la récursivité
3. La programmation dynamique
4. La méthode des essais successifs ou retour arrière
5. La notion de non déterminisme
6. Une alternative à l'exploration combinatoire : les algorithmes gloutons

# Licence informatique

## Architecture et systèmes : architecture 1

L'objectif du cours d'architecture est d'enseigner aux futurs informaticiens les bases nécessaires en architecture matérielle des ordinateurs, principalement pour favoriser une meilleure compréhension des couches basses de système. Cette première partie est consacrée à la conception des circuits digitaux.

### PLAN

1. Algèbre de Boole et représentation des informations
2. Conception de circuits digitaux
  - circuits logiques ;
  - circuits combinatoires ;
  - circuits séquentiels ;
  - décomposition en unités de traitement et de contrôle.

## **Licence informatique**

### **Architecture et systèmes : système 1**

Cet enseignement a deux objectifs. D'une part, donner une connaissance précise du fonctionnement d'un ordinateur, du point de vue de son utilisateur. D'autre part, traiter des problèmes systèmes liés à la désignation : comment représenter en machine les objets du programme, comment accéder à ces représentations. Ces notions sont illustrées par des travaux pratiques en langage d'assemblage.

#### **PLAN**

1. Structure et fonctionnement de base d'un ordinateur
2. Mécanismes d'adressage
3. Accès aux objets en mémoire (gestion statique et dynamique)

# Licence informatique

## Architecture et systèmes : architecture 2

L'objectif de ce cours est d'enseigner aux futurs informaticiens les bases nécessaires en architecture matérielle des ordinateurs, principalement pour favoriser une meilleure compréhension des couches basses de système. Cette seconde partie de l'enseignement d'architecture est orientée vers l'organisation des systèmes : processeurs, mémoires centrales, bus, entrées-sorties, interruptions, hiérarchies de mémoire.

### PLAN

1. Structure d'un ordinateur
  - structure interne des processeurs ;
  - technologie et utilisation des mémoires ;
  - signaux de bus.
2. Entrées/Sorties
  - synchronisation par attente active ;
  - synchronisation par interruptions ;
  - transmissions séries.
3. Hiérarchies de mémoire
  - principes généraux ;
  - caches ;
  - mémoires virtuelles.

## **Licence informatique**

### **Architecture et systèmes : système 2**

On présente d'abord les différentes phases conduisant à l'exécution d'un programme (comment élaborer les informations permettant de réaliser les accès aux objets du programme, quels sont les logiciels de base effectuant cette élaboration), puis on aborde les systèmes d'entrée/sortie et de gestion des fichiers. Une introduction au langage de commandes termine cette partie.

#### **PLAN**

1. Logiciels de base : assembleur, éditeur de liens, chargeur
2. Système d'E/S physiques
3. Structure et gestion des fichiers
4. Langage de commandes (exemple d'Unix)



# **Licence informatique**

## **Anglais**

Objectif : consolidation par la pratique des quatre compétences de base que sont la compréhension et l'expression, écrites et orales dans le cadre d'une langue généraliste permettant d'intégrer une équipe de travail. L'accent est mis sur la pratique orale, la prise de parole, individuelle ou en groupe, par le biais d'exposés, de travail par paires ou de jeux de rôles, et l'étude de documents audio-visuels. Les supports des cours sont des extraits de la presse anglophone ainsi que des documents audiovisuels divers pour favoriser le repérage de formats standards, l'étude de documents audio bruts, et l'écoute sélective. Une remédiation personnalisée est proposée ponctuellement en grammaire par le biais de l'EAO en salle multimédia, en fonction des besoins exprimés ou identifiés au cours des exposés. Le cours d'anglais de licence (34 heures) est divisé en créneaux de une heure et demi répartis en cinq unités tout le long de l'année. L'assiduité est obligatoire.

## **PLAN**

1. "Socializing"
2. "Telephoning 1"
3. "Making Presentations"
4. "The Internet"
5. "Computer History" Chaque unité comprend environ quatre créneaux dont un est consacré à la remise à niveau grammaticale grâce à l'EAO

# **Licence informatique**

## **Bases de données**

Ce cours n'a pas de prérequis. Son objectif est de faire rapidement le point sur l'évolution des bases de données. Nous étudions plus particulièrement les bases de données relationnelles. Ces notions sont illustrées par des travaux pratiques en langage SQL.

### **PLAN**

1. Notions de bases : modèles hiérarchique, réseau, relationnel
2. Algèbre relationnelle
3. Conception de schémas relationnels
  - dépendances fonctionnelles ;
  - formes normales ;
  - algorithmes de décomposition.
4. le langage SQL
  - création de tables ;
  - interrogation.

# Licence informatique

## Introduction à la compilation

Ce cours introduit les deux niveaux d'analyse présents dans un compilateur : l'analyse lexicale (automate d'états fini) et l'analyse syntaxique (automate à pile). On étudie ensuite comment sont compilées les constructions de base d'un langage impératif (déclarations, expressions, instructions, procédures). Enfin, on présente les notions nécessaires à la compilation d'unités écrites séparément et à la constitution d'un programme exécutable par un éditeur de liens.

### PLAN

1. Programmation par automate d'états fini
2. Analyse syntaxique descendante gauche droite
3. Éléments de compilation
  - table des identificateurs ;
  - calcul de type et compilation des expressions ;
  - compilation des instructions et des procédures ;
  - compilation séparée et édition de liens.
4. Automates à pile, grammaires LL(1)

# **Licence informatique**

## **Initiation aux services Internet**

Cette conférence de deux heures, facultative, est dispensée en début d'année. Elle comporte une partie technique et une partie consacrée aux règles de comportement sur l'Internet. Au plan technique, il s'agit de donner à chaque étudiant des bases minimales pour manipuler les services de l'Internet qui lui sont accessibles à partir du réseau de l'Ifsic. Pour ce qui concerne les règles de comportement sur l'Internet, on insiste sur les aspects légaux et déontologiques et on précise la réglementation propre à l'Institut.

### **PLAN**

1. L'Internet et ses principaux services
  - quelques éléments techniques : réseaux informatiques, interconnexion de réseaux hétérogènes et protocole TCP/IP, état actuel de l'Internet, financement ;
  - description sommaire des principaux services : courrier électronique, forums, transferts de fichiers, consultation de documents indexés, navigation et accès à l'information.
2. Le World Wide Web
  - utilisation de Netscape ;
  - outils de recherche ;
  - langage Html et outils de construction de serveurs.
3. Netiquette
  - les lois et la déontologie ;
  - règles propres à l'Ifsic.
4. Conclusions
  - aspects économiques et sociaux ;
  - perspectives d'évolution.

# **Licence informatique**

## **Projet**

Le but visé est de familiariser les étudiants avec des outils favorisant la production de logiciels. En s'appuyant sur l'enseignement de la compilation, il s'agit d'abord de réaliser un programme dont le squelette est un automate d'états fini puis d'écrire un compilateur et un éditeur de liens pour un langage sous-ensemble de Pascal. Ce projet présente, de plus, l'avantage de faire travailler les étudiants par groupes de quatre.

# **Licence informatique**

## **Langages formels**

Dans ce cours sont définis les mots, les langages et les opérations sur ces objets. Les langages étudiés sont essentiellement les langages algébriques et rationnels (types 2 et 3). Différentes approches coexistent pour décrire un langage : il peut être reconnu par un automate, engendré par une grammaire, dénoté par une expression rationnelle, solution d'un système d'équations ; on s'attache à montrer l'équivalence de ces modes de description, par des manipulations sur ces outils eux-mêmes.

### **PLAN**

1. Introduction : la théorie des langages et ses applications en informatique
2. Le monoïde libre
3. Engendrer des mots, des langages : les grammaires algébriques
4. Reconnaître des mots, des langages : les automates finis
5. Langages reconnaissables
6. Langages rationnels et théorème de Kleene
7. Formes de grammaires
8. Automates à pile
9. La hiérarchie de Chomsky
10. Décidabilité

# Licence informatique

## Logique

Ce cours de logique vise deux buts : d'une part, donner une base solide en raisonnement logique, ceci est l'aspect « calcul » ; d'autre part, introduire la notion de système formel, essentielle en informatique. Enfin, ce cours donne l'occasion de présenter, de façon informelle, différentes notions approfondies plus tard comme les langages, la sémantique, la calculabilité, la décidabilité.

### PLAN

1. Introduction
2. Calcul des propositions
  - langage et sémantique du calcul des propositions ;
  - systèmes formels.
3. Calcul des prédicats
  - langage et sémantique du calcul des prédicats ;
  - systèmes formels.

# **Licence informatique**

## **Programmation 1**

Cet enseignement porte sur les structures de données classiques (listes, arbres, ensembles, tables) par une approche type abstrait/ programmation objet : on définit d'abord la structure du type étudié ainsi que les primitives de manipulation (méthodes) puis, dans un second temps, on examine les représentations possibles en mémoire (mises en oeuvre). Le langage utilisé pour les travaux pratiques est Java.

### **PLAN**

1. Eléments de programmation objet : classe, objet, héritage, généricité
2. Listes
3. Récursivité
4. Arbres binaires
5. Ensembles / Tables, étude du polymorphisme
6. Partage de tables



# Licence informatique

## Programmation 2

Ce cours présente les bases de la programmation séquentielle et de la programmation fonctionnelle. Il insiste tout particulièrement sur les propriétés logiques des structures de programmes et sur leur utilisation dans le processus de développement raisonné de programmes.

### PLAN

1. Introduction à la programmation
2. La programmation fonctionnelle
  - Principes généraux ;
  - l'exemple de CAML.
3. Les types abstraits de données
4. La programmation séquentielle
  - Structures de programmes ;
  - Calculs de programmes ;
  - Construction de programmes ;
  - Analyse et preuve de programmes.

## **Maîtrise informatique**

### **Analyse, développement et test**

Ce cours rassemble des informations sur les techniques et les outils de développement utilisés actuellement par les informaticiens. Il est composé de deux parties : les techniques de conception et d'analyse orientées objets, et le test. La première partie met l'accent sur la structuration de logiciels au moyen de patrons de conception. Les problèmes du test et de la gestion de la qualité sont abordés dans la seconde partie. Les travaux pratiques utilisent la notation UML.

#### **PLAN**

1. Analyse et conception par objets, but et moyens de l'analyse par objets, conception par "design patterns"
2. Gestion de la qualité, test

# Maîtrise informatique

## Programmation objet et parallèle

Après des compléments sur la notion de type et les types fonctionnels, on étudie les types structurés pour terminer par le typage des classes et des objets. On étudie ensuite la relation existant entre le concept d'objet et celui de processus, puis on examine divers outils permettant d'exprimer la communication entre processus et le contrôle de systèmes de processus communicants. Il est alors possible d'étudier quelques aspects des langages orientés objets parallèles permettant de construire des systèmes d'objets communicants.

### PLAN

1. Eléments de programmation objet
  - Principes généraux ;
  - Principaux composants de Java.
2. De l'objet au processus
3. Communications et contrôle
4. Appel de procédures à distance
5. Rendez-vous
6. Partage
7. L'approche Gamma
8. Programmation orientée objet et parallélisme
9. Sémantique du parallélisme
10. Quelques éléments de mise en oeuvre

# **Maîtrise informatique**

## **Systèmes d'exploitation 1 : Gestion du parallélisme dans les systèmes d'exploitation**

Les deux cours de système de maîtrise ont pour objectif de présenter les concepts fondamentaux sur lesquels s'appuient les systèmes d'exploitation multiprogrammés, ainsi que les éléments principaux de leurs mises en œuvre.

Ce premier cours s'intéresse au contrôle de l'exécution dans un contexte de parallélisme : on y voit d'une part un modèle pour l'expression du contrôle d'activités concurrentes et d'autre part son utilisation et sa mise en œuvre dans les systèmes d'exploitation.

### **PLAN**

1. Introduction
  - rôles et structure du système d'exploitation, historique.
2. Processus séquentiel
  - processus : définition, propriétés ;
  - section critique, mécanisme de synchronisation.
3. Mise en œuvre d'un noyau de synchronisation
  - mécanismes matériels de contrôle de l'exécution ;
  - mise en œuvre du noyau ;
  - allocation de l'unité centrale.
4. Synchronisations entre processus
  - schémas classiques de synchronisation ;
  - autres outils de synchronisation.
5. Gestion des entrées/sorties
  - support matériel ;
  - gestion des synchronisations entre processeur et interface d'entrées/sorties ;
  - structure du système d'entrées/sorties ;
  - gestion des disques.
6. Allocation de ressources
  - interblocage ; prévention, détection/guérison.
7. Structure du système d'exploitation - exemples
  - modèle d'exécution ;
  - architecture du système.

MAITINFO / SYS1 : Cours : 20 heures ; TD : 20 heures ; TP : 12 heu-

# Maîtrise informatique

## Compilation

Ce cours présente les composants fondamentaux d'un compilateur et les principales techniques de compilation. Les langages impératifs ainsi que les fondements de l'architecture des ordinateurs sont supposés connus. Les travaux dirigés et travaux pratiques ont pour but la construction d'un compilateur pour un langage impératif.

### PLAN

1. Introduction à la compilation
2. Analyse lexicale et analyse syntaxique
3. Vérification de types
4. Environnement d'exécution
5. Génération de code intermédiaire
6. Introduction à la compilation optimisante
7. Analyse de flots de données

# **Maîtrise informatique**

## **Option 1 : Automates et réseaux**

Ce cours se divise en deux parties assez indépendantes. La première fait suite au cours de langages formels de licence et étudie les notions de calculabilité et de décidabilité à travers l'étude d'une hiérarchie de modèles de calculs basés sur des automates finis. La deuxième partie présente le modèle des réseaux de Petri et son utilisation dans la modélisation de systèmes séquentiels ou répartis. La fin du cours illustre par le problème de la synthèse de réseau la relation entre automates finis et réseaux de Petri.

### **PLAN**

#### **1. Modèles de calcul basés sur les automates**

- modèles de calcul ;
- automates finis ;
- automates à pile ;
- machines de Turing ;
- fonctions récursives ;
- non-calculabilité.

#### **2. Réseaux de Petri**

- introduction aux réseaux de Petri ;
- condition/event nets, place/transition nets ;
- réseau borné et graphe de couverture ;
- le problème de la synthèse des réseaux de Petri.

# Maîtrise informatique

## Option 1 : Modélisation par files d'attente

Pour évaluer quantitativement un système informatique, les files d'attente constituent un outil de prédilection. Au niveau des modèles analytiques, les files markoviennes permettent d'obtenir facilement des valeurs telles que les temps de réponse moyens ou les débits. Toutefois, comme beaucoup d'études de systèmes réels ne trouvent pas de réponse entièrement satisfaisante via l'étude de modèles suffisamment simples pour admettre une solution analytique, ce cours contient aussi un chapitre consacré à la simulation des systèmes à événements discrets.

### PLAN

1. Chaînes de Markov à temps continu (CMTC)
  - cas général ;
  - processus de naissance et de mort (PNM) ;
  - processus de Poisson.
2. Simulation
  - principes ;
  - étude du logiciel QNAP2 ;
  - réalisation de programmes de simulation.
3. Files d'attente markoviennes
4. Réseaux de files d'attente

# **Maîtrise informatique**

## **Option 1 : Programmation logique et par contrainte**

Introduction à la programmation logique et ses aspects novateurs. Vision logique, vision impérative. On met l'accent sur la programmation d'applications. Les objets de base sont introduits progressivement. Des applications privilégiées telles que l'analyse syntaxique sont particulièrement étudiées.

### **PLAN**

1. Introduction
  - langages impératifs, langages déclaratifs ;
  - langages fonctionnels, langages relationnels ;
  - faits, clauses, variables existentielles, variables universelles ;
  - non-déterminisme.
2. Présentation générale, un premier survol
  - arbre de recherche ;
  - arithmétique, évaluer en Prolog ;
  - unifier n'est pas évaluer ;
  - récursivité, les listes ;
  - prédicat dif, contrainte dif.
3. Les termes, les formules, interpréter les formules
  - syntaxe des termes, terme clos, instance de terme, unification ;
  - sémantique déclarative, déduction naturelle ;
  - sémantique impérative, effacement de but.
4. Maîtriser la complexité, accumulateurs
5. Analyse et constructions de termes
6. La négation
  - normaliser des formules logiques, forme clausale, résolution ;
  - clauses de Horn et résolution slr ;
  - hypothèse du monde clos, programme complété, négation par l'échec.
7. Structures de données incomplètes
8. Analyse syntaxique
9. Prédicats prédéfinis extra-logiques, list-of

MAITINFO / PLC : Cours : 18 heures ; TD : 18 heures ; TP : 12 heu-



# Maîtrise informatique

## Méthodes formelles pour le génie logiciel

Pour résoudre les problèmes posés par la production industrielle de logiciels de qualité, diverses méthodes et techniques ont été développées. Nous nous intéressons dans ce module aux méthodes dites formelles dont la caractéristique principale est d'utiliser, pour décrire un logiciel, un langage suffisamment formalisé pour que l'on puisse faire des preuves, au moins partiellement automatisées, sur la description. Le module est construit autour d'une méthode particulière, la méthode B, développée par J.R. Abrial, qui offre d'une part un langage permettant à la fois de formaliser une spécification et de décrire les étapes de la conception, et d'autre part une méthodologie pour s'assurer de la validité de ce qui est produit. Au delà de l'apprentissage d'un langage de spécification particulier l'objectif de ce module est de familiariser les étudiants avec une méthode de travail. On insistera plus sur l'utilisation du langage pour décrire une spécification que sur la réalisation des preuves qui suppose l'utilisation d'outils adéquats.

### PLAN

1. Introduction
  - qualité du logiciel, génie logiciel, méthodes formelles.
2. Méthode B : un exemple introductif
3. Outils de base
  - outils de preuve : logique ;
  - outils élémentaires de description : ensemble, relations, fonctions ;
  - applications.
4. Machine abstraite
  - définition et composition.
5. Programmation
  - séquençement et boucle.
6. Spécification par étapes
  - raffinement, implémentation.

## **Maîtrise informatique**

### **Systèmes d'exploitation 2 : Gestion des informations dans les systèmes d'exploitation**

Les deux cours de systèmes de maîtrise ont pour objectif de présenter les concepts fondamentaux sur lesquels s'appuient les systèmes d'exploitation multiprogrammés, ainsi que les éléments principaux de leurs mises en oeuvre. Le contenu du cours "Systèmes d'exploitation 1 : Gestion du parallélisme dans les systèmes d'exploitation" est un pré-requis pour cette seconde partie.

Ce second cours s'intéresse à la gestion et à l'accès aux informations dans les systèmes d'exploitation, on y voit en particulier les mécanismes permettant de protéger, de partager et d'accéder rapidement aux informations, quel qu'en soit le support.

#### **PLAN**

1. Mécanismes généraux de gestion de l'information
  - désignation et liaison ;
  - protection ;
  - hiérarchie de mémoire et cache.
2. Désignation et liaison des objets dans un programme
  - types d'objets manipulés ;
  - partage d'objets ;
  - édition de liens dynamique ;
  - espace virtuel segmenté.
3. Gestion de la mémoire
  - comportement des programmes ;
  - réimplantation dynamique ;
  - gestion de mémoire par zones ;
  - mécanismes de pagination, gestion d'une mémoire paginée.
4. Système de gestion de fichiers
  - partage de fichiers ;
  - exemple : Unix ;
  - systèmes distribués de gestion de fichiers.
5. Conclusion

# **Maîtrise informatique**

## **Réseaux et protocoles**

L'analyse comparée des différents mécanismes et techniques employés pour la réalisation des réseaux de télécommunication nous permet de dégager leurs rôles, leurs mérites et leurs évolutions. Beaucoup plus qu'une simple description de format ou de comportement, les exemples expliqueront la raison des choix effectués lors de la conception de tels réseaux et protocoles.

Les étudiants vont acquérir une large connaissance du monde téléinformatique, ses concepts et le vocabulaire associé, et obtenir une parfaite compréhension des mécanismes de base. Ils seront capables de développer des applications informatiques utilisant au mieux les caractéristiques des réseaux et protocoles employés, d'adapter ces réseaux et protocoles à de nouveaux besoins, et finalement de suivre leurs évolutions futures.

### **PLAN**

1. La normalisation, l'architecture de référence
2. Les supports de communication, la modulation du signal
3. Le codage pour la transmission
4. La protection contre les erreurs : retransmission et codes
5. Le protocole HDLC et le contrôle de flux
6. Les réseaux locaux : Ethernet
7. Interconnexion de réseaux : équipements et protocoles
8. Plans d'adressage et techniques de routage
9. Le protocole X25 : Transpac
10. Le niveau Transport : la qualité de service
11. Le niveau Session : activités et synchronisations
12. Le niveau Présentation : ASN-1 et BER
13. Le niveau Application

MAITINFO / REPR : Cours : 24 heures ; TD : 24 heures ; TP : 12 heures

# **Maîtrise informatique**

## **Option 2 : Algorithmique distribuée**

Le but de ce cours est d'étudier des techniques algorithmiques avancées. Il est composé de deux parties : une introduction à l'algorithmique répartie et des algorithmes de compression de données.

### **PLAN**

#### **1. Introduction à l'algorithmique répartie**

- étude d'un mécanisme de communication par messages ;
- description du comportement des processus par automates communicants ;
- étude d'applications concrètes distribuées utilisant les parcours répartis, les concepts de vagues ou de phases ;
- réalisation d'applications sur un réseau de SUN.

#### **2. Compression de données**

- codage des répétitions : Run Length Encoding ;
- codage préfixe des caractères : Huffman statique et dynamique ;
- dictionnaire par glissement de fenêtre (LZ77, ZIP) ;
- dictionnaire par prolongement de chaînes (LZW).

# Maîtrise informatique

## Option 2 : Images numériques

Ce cours est avant tout destiné à mettre en évidence l'utilisation, dans le domaine de l'image numérique, des techniques informatiques et des structures de données présentées aux étudiants au cours de leurs études de licence et maîtrise. Ce cours comporte une initiation au vocabulaire et à quelques concepts de base pour les images, et met en évidence les relations entre les techniques développées pour la synthèse et celles développées pour l'analyse.

### PLAN

1. Représentations
  - la matrice de pixels ; représentation des couleurs.
2. Matériel
3. Traitement d'images binaires
  - suivi de contour ;
  - squelettisation.
4. Filtrage
  - exemples et définitions dans le cas du filtrage monodimensionnel ;
  - calcul des coefficients du filtre à partir de la fonction de transfert ;
  - filtres passe-bas, dérivateur, polynomial, bidimensionnel, médian.
5. Traitement d'images en niveaux de gris
  - histogramme ;
  - modification de l'échelle des gris ;
  - segmentation dans l'espace des luminosités ;
  - introduction à la détection de contours.
6. Représentation d'objets tridimensionnels
  - représentation polyédriques ;
  - transformations.
7. Structuration de scènes
8. Visualisation d'objets
  - projection, perspective ; transformation 3D en 2D ;
  - élimination des faces cachées, rendu des objets (aspect).

# **Maîtrise informatique**

## **Option 2 : Intelligence artificielle**

Le but de ce cours est d'introduire les méthodes informatiques de résolution de problèmes propres à l'intelligence artificielle. Ces méthodes s'appliquent à des problèmes complexes qui peuvent être résolus par une manipulation symbolique et déductive de connaissances et pour lesquels une approche algorithmique n'est pas envisageable pratiquement. Ces méthodes reposent sur deux principes : la représentation de connaissances au sein de formalismes adéquats et l'utilisation de techniques de résolution de problèmes.

### **PLAN**

1. Qu'est-ce que l'intelligence artificielle ?
2. Représentation des connaissances
3. Résolution de problèmes
  - problèmes de satisfaction de contraintes ;
  - problèmes avec changement d'état ;
  - problèmes décomposables ;
  - jeux de stratégie.
4. Planification et génération de plans d'actions
5. Systèmes-experts

# **Maîtrise informatique**

## **Méthodes de génie logiciel**

Le projet de Maîtrise offre une approche expérimentale des problèmes et des méthodes de génie logiciel par la réalisation de travaux informatiques relativement importants, dans la durée (8 mois), et par équipes (12-15 étudiants).

Au cours du premier semestre, les étudiants réalisent un Travail d'Étude et de Recherche (T.E.R.) qui prépare la réalisation demandée. Cela comprend une étude bibliographique du domaine, une analyse du problème et la rédaction d'un rapport de spécification ainsi qu'une soutenance orale. Pendant cette période, l'enseignant présente la problématique du domaine et les méthodes informatiques qu'il souhaite voir employées.

Le second semestre contient la réalisation proprement dite pendant laquelle les étudiants mettent en oeuvre les spécifications qu'ils ont proposées, dans le cadre des méthodes souhaitées par l'encadreur. Le projet est complété par un rapport d'implémentation, une seconde soutenance orale, et si possible une démonstration du système réalisé.

Les étudiants préparent cet enseignement au sein de groupes d'une quinzaine d'étudiants. Ces groupes sont eux-mêmes découpés en sous-groupes concurrents ou coopérants selon les sujets. Les sujets proposés en 1999 sont les suivants :

### **PLAN**

1. Analyse et transformation de programmes C
2. Animation 3D pour Internet
3. Calcul d'éclairage
4. Système d'information génomique
5. Serveur NFS pour un système de fichiers relationnel
6. Conception et réalisation de didacticiel

MAITINFO / PROJ : TD : 24 heures ; TP : 300 heures.

## **Maîtrise informatique**

### **Anglais**

Le cours de maitrise se place dans la continuité du cours de licence dont il renforce les acquis, avec comme objectif principal la prise de parole en public dans des situations pratiques (exposés improvisés, demande de renseignements, négociations, résolution de conflits, en plus particulièrement travail au sein d'un groupe). Les trois autres compétences sont aussi renforcées : - la compréhension orale avec l'étude des titres du journal de Radio 4 et de documentaires video de la BBC - l'expression écrite avec la redaction de documents de standards differents et des conventions qu'ils imposent (niveaux de langue, formats...) - la lecture et le décodage de textes longs ou techniques. Au cours de l'année de maitrise, les étudiants doivent elaborer et mener a bien un projet personnel de leur choix (recherche de stage, travail linguistique, lecture, rencontres avec les lecteurs, projet de voyage...) et dont ils doivent rendre compte a intervalles reguliers. Une remediation personnalisee en grammaire et prononciation est proposee en EAO.

### **PLAN**

1. "Speaking about oneself : British education system, careers, job interviews"
2. "Artificial Intelligence : Life and Death of Alan Turing"
3. "Telephoning 2"
4. "Negotiating and mediating crises"
5. "Writing documents"
6. "The Internet Story (Birth of the Internet)"