Travaux pratiques d'informatique

Présentation de l'épreuve

L'épreuve de travaux pratiques d'informatique en MPI est une épreuve d'algorithmique et de programmation, avec utilisation d'un ordinateur, d'une durée de 3 heures. À partir d'un sujet imposé, elle demande de traiter sur machine des questions de programmation, d'effectuer des choix de modélisation et d'aborder certains aspects théoriques de l'informatique, tout en communiquant très fréquemment avec le jury.

L'objectif de l'épreuve est d'évaluer les capacités de programmation, la maitrise des méthodes classiques du programme, les capacités de modélisation, d'abstraction et d'inventivité ainsi que l'application de bonnes pratiques que l'on est en droit d'attendre de futurs ingénieurs.

Organisation de l'oral

Les candidats disposent d'un ordinateur fourni par le concours, configuré avec un environnement de travail adapté aux sujets demandés. Pour la session 2024, l'environnement choisi était à nouveau, comme en 2023, la distribution GNU/Linux Ubuntu 22.04, construite de façon analogue à l'environnement Pronaos (https ://gitlab.com/agreg-info/clef-agreg/).

Avant le début de l'épreuve, le jury a laissé plusieurs minutes aux candidats afin de prendre en main librement cet environnement de travail. Pendant ce temps, qui n'entre pas en compte dans l'évaluation, il a été possible de poser toutes les questions nécessaires aux membres du jury. Le jury compte reconduire ce temps de 10 minutes de prise en main de l'environnement lors des sessions à venir.

Les sujets ont été fournis durant l'épreuve en version papier ainsi qu'en version numérique au format PDF. Les sujets étaient tous accompagnés de fichiers auxiliaires, comprenant notamment :

— des fichiers sources, complets ou à compléter selon les cas ;

— des fichiers de données à exploiter ;

— des scripts d'aide à la compilation.

Pour cette session, ces fichiers ont été mis à disposition dans un dossier de travail spécifique à chaque candidat, dont l'accès est déverrouillé grâce à un code donné en début d'épreuve par le jury. Les candidats pouvaient travailler directement dans ce dossier, il ne leur était demandé aucun transfert de fichier.

Les candidats sont responsables des sauvegardes de leur travail, qu'ils doivent effectuer à intervalles réguliers.

Les sujets sont organisés en une suite de questions, à traiter généralement de façon linéaire. Ces questions sont réparties en trois catégories :

— des questions de programmation ;

— des questions à préparer pour une présentation orale ;

— des questions de rédaction à réaliser sur un compte-rendu.

Le concours fournit toujours les feuilles et les brouillons utiles à l'épreuve. Les candidats doivent se munir du matériel d'écriture usuel (stylos, crayons, gomme, règle). Aucun autre matériel n'est autorisé. Pour certaines applications numériques demandées par les sujets, les candidats ont convenablement réussi à utiliser les fonctions de calcul des ordinateurs.

Les sujets proposés suivent tous le format des épreuves en vigueur depuis la session 2023.

Analyse globale des résultats

Les prestations des candidats sont dans leur très large majorité d'excellente qualité. Le jury constate une hausse sensible du niveau par rapport à la session 2023, que ce soit sur les compétences en programmation ou sur les compétences théoriques.

Le niveau observé en programmation est très satisfaisant. Les programmes produits respectent un grand nombre de critères de qualité (organisation, présentation, conventions de nommage) et la syntaxe des langages est bien maitrisée.

Les connaissances de cours sont également globalement complètes, mais le jury a constaté çà et là plusieurs faiblesses récurrentes. Le jury sur le fait que le contenu du cours de MP2I et de MPI doit être connu, au point d'être restitué efficacement. Les sujets continueront de proposer des questions de cours, sans lequel il est vraiment difficile d'aborder correctement les questions de programmation qui suivent. Ces questions de cours ont été la source de différences marquées dans l'évaluation des candidats.

Commentaires sur les réponses apportées et conseils aux futurs candidats

Programme des épreuves orales

Les sujets posés permettent d'évaluer l'ensemble des notions présentes dans les programmes d'informatique des classes de MP2I et de MPI. Chaque sujet porte sur une ou plusieurs parties spécifiques de ces programmes. L'ensemble de tous les sujets couvrait globalement l'intégralité du programme des deux années.

Les langages évalués sont `C`, `OCaml` et `SQL`. Les sujets peuvent utiliser, selon la pertinence par rapport au thème abordé, un seul, deux ou trois langages (mais jamais `SQL` seul). La très grande majorité des sujets évalue au minimum deux langages. Le langage à utiliser pour chaque question est quasiment toujours imposé, mais certaines parties ont parfois été laissées librement au choix des candidats.

Évaluation des épreuves orales

Les sujets proposés restent volontairement longs. Les candidats les plus efficaces ont pu tout de même aborder la quasi totalité du contenu des sujets proposés. Les sujets comportent tous des parties d'application immédiate du cours, assez guidées, destinées à évaluer les compétences de base en programmation et les connaissances de cours, ainsi que des parties d'ouverture favorisant la prise d'initiative et permettant d'évaluer l'autonomie et la capacité de prise de recul des candidats.

L'évaluation de l'épreuve ne consiste cependant pas exclusivement, et loin de là, en une mesure du plus grand nombre de questions traitées. Le jury a valorisé non seulement l'efficacité en terme de programmation mais également l'intérêt porté à la discipline de programmation (compiler fréquemment, s'assurer que le code fonctionne, proposer spontanément des tests). L'épreuve de travaux pratiques d'informatique doit mener à un programme qui fonctionne effectivement. Le jury préfère des prestations avec un peu moins de questions traitées mais dans lesquelles les programmes écrits sont correctement compilés et soigneusement testés.

Les candidats étaient, pour cette session encore, bien préparés à l'épreuve. Ceci a permis lors des épreuves d'avoir des échanges entre le jury et les candidats de très haute qualité. Le jury a constaté que dans la très grande majorité des cas, les compétences visées par les programmes de MP2I et MPI étaient acquises, que les aspects théoriques étaient maitrisés et que la technique était manipulée avec efficacité. Les candidats

ont parfaitement démontré leurs aptitudes de futurs ingénieurs en informatique. Le jury adresse à nouveau ses félicitations à l'ensemble des candidats.

Évolution du format de l'épreuve

Le format de l'épreuve a permis d'évaluer correctement le niveau des candidats, et il n'est pas prévu d'évolution majeure pour la session 2025.

Le jury envisage de limiter la rédaction de questions sur le compte-rendu, qui servira uniquement de support pendant l'interaction orale. Les questions à noter sur le compte-rendu seront relues directement pendant la séance, et ne seront pas évaluées après l'épreuve. Le jury ne demande pas de rédaction très poussée comme on pourrait en trouver sur une copie à l'écrit, d'éventuelles imprécisions pouvant être levées par la discussion orale. Les candidats disposeront toujours de feuilles de brouillon ainsi que d'un support papier afin de préparer leurs réponses et d'indiquer certains résultats demandés.

Publication des sujets

Afin de renforcer la préparation à cette épreuve, le jury publie un nouvel échantillon de sujets qui ont été donnés lors de cette session.

Maitrise du cours

Le jury considère que la maitrise du cours demeure un élément essentiel : il encourage les candidats à bien apprendre le cours afin de traiter rapidement et efficacement les questions qui s'y rapportent. Les questions de cours ont souvent été discriminantes, d'autant plus que certains manques de connaissances ont été bloquants pour traiter l'approche proposée ensuite par le sujet. Dans de tels cas, le jury a été contraint de faire des rappels de cours très importants, ce qui se ressent nécessairement sur la note obtenue. À titre d'exemple, une très grande majorité de candidats n'est pas capable d'expliquer correctement en quoi consiste une table de hachage, ne parvenant pas à reconnaitre ni à compléter les implémentations proposées.

Le jury a constaté plusieurs confusions, où des candidats confondent des algorithmes de recherche de motifs avec des algorithmes de compression, ou bien confondent les méthodes algorithmiques usuelles (diviser pour régner, programmation dynamique, etc.) sans en connaitre vraiment le principe.

Le jury invite les candidats à faire preuve de précision dans l'utilisation des noms d'algorithmes au programme, dans une optique d'efficacité de la communication. La simple confusion de noms n'a cependant pas été sanctionnée quand il était clair que les candidats maitrisaient l'algorithme demandé et savaient le décrire précisément. À contrario, redonner simplement en tant que mot-clé le nom d'un algorithme sans en connaitre le principe n'a pas été valorisé.

Préparation technique

L'environnement de travail du concours permet d'évaluer l'intégralité des compétences prévues par les programmes de MP2I et MPI. Le jury constate que tous les candidats semblaient, pour cette session, mieux préparés à cet environnement. Ils ont su en maitriser tous les aspects sans aucune difficulté.

Le jury a fourni, avant l'épreuve et pendant l'épreuve, les indications nécessaires aux candidats pour faire fonctionner cet environnement, en particulier sur des points non exigibles des programmes de MP2I et MPI (aide à l'invocation des compilateurs, rappels sur le système de modules en `OCaml` lorsque le sujet fournissait plusieurs modules, utilisation des scripts de compilation fournis, aide à l'accès aux outils de détection d'erreurs tels que l'address sanitizer, l'activation des avertissements du compilateur, etc.).

Afin de maintenir le bon niveau de préparation constaté cette année, *le jury encourage de nouveau toutes les formations à proposer à leurs élèves, durant leur scolarité, un environnement de travail adapté au programme de MP2I et MPI*, permettant au minimum d'aborder les points suivants :

— l'accès à un shell dans un terminal permettant d'invoquer les compilateurs mais également d'autres commandes, et de manière générale permettant de se familiariser avec le fonctionnement d'une ligne de commandes ;

— le fonctionnement d'un système POSIX, en particulier quant à la gestion des processus, des fils d'exécution, des flux standard et leur redirection, éventuellement avec des tubes ;

— la familiarisation avec des outils de compilation usuels (`make` ou `dune`). Leur usage fera toujours l'objet d'un rappel et aucune compétence spécifique n'est attendue, mais avoir déjà rencontré ces outils permet une plus grande efficacité.

Le jury rappelle l'indication suivante des programmes de MP2I et MPI : « Bien que ces notions soient indépendantes du système d'exploitation, le système Linux est le plus propice pour introduire les éléments de ce programme. »

Remarques concernant `OCaml`

Le jury maitrise la bibliothèque standard du langage mais il n'exige pas que les candidats utilisent d'eux-mêmes des fonctions de haut niveau, par exemple celles provenant du module `List`. L'évaluation est par exemple indifférente à l'utilisation, qui n'est ni valorisée ni sanctionnée, de `List.fold_left` (et autres fonctions similaires). Le jury est surtout attentif à la clarté du code produit, à sa correction et à la maitrise de ce code par les candidats.

Il est pertinent de réfléchir quelques instants à la manière de concevoir un code, pour éviter que le résultat soit inutilement compliqué et soit source d'erreurs difficiles à détecter et corriger.

Certains sujets invitaient à utiliser dans un même programme des aspects fonctionnels et des aspects impératifs du langage, ce qui a déstabilisé plusieurs candidats ou bien a conduit à la rédaction de codes très compliqués. Le jury rappelle qu'un `match` ou un `if` sont des expressions comme les autres en OCaml et qu'il est parfaitement admis d'utiliser ces constructions partout où une expression est permise par le langage (membre droit d'un `let`, corps d'une boucle). Il est cependant impératif de bien connaitre les priorités des constructions et des opérateurs et de savoir parenthéser correctement (avec les mots-clés `begin-end` ou avec des parenthèses). Le jury conseille de systématiquement introduire ce parenthésage en présence conjointe de la construction `if then else` et de l'opérateur point-virgule (`;`), les erreurs de priorité à ce niveau ont posé des problèmes à de très nombreux candidats, engendrant souvent une importante perte de temps.

Les candidats doivent savoir proposer un programme `OCaml` complet, qui compile dans son intégralité et pas uniquement ligne par ligne dans un REPL. Une aide à la compilation a systématiquement été fournie. Le jury accepte si nécessaire le double point-virgule (`; ;`) pour séparer les phrases, même si ce dernier ne fait pas partie du langage à proprement parler ; il le propose aux candidats quand cela aide parfois à mieux cerner certaines erreurs de syntaxe difficiles à situer d'après le message du compilateur. Sur ce dernier point, le jury attire l'attention des candidats sur le fait que `;` est un opérateur binaire et non pas un terminateur d'instruction.

Les candidats doivent savoir identifier les erreurs de syntaxe dans leurs programmes, celles-ci se situent souvent bien avant la ligne à laquelle est indiquée l'erreur.

Remarques concernant `C`

La gestion de la mémoire est un aspect maitrisé par de nombreux candidats, mais les oublis d'allocations avec `malloc` et surtout les oublis de libération avec `free` ont été trop fréquents. Il est toujours pertinent de s'interroger sur la politique d'allocation de la mémoire, en particulier lorsque le sujet invite à programmer une petite API manipulant une structure de données, dont on se sert ensuite. Avoir une politique claire d'allocation *et de libération* est essentiel.

Lors de la préparation des candidats, il peut être utile d'indiquer comment compiler un programme avec `gcc -g -Wall -fsanitize=address` et d'interpréter la sortie en cas d'erreur de segmentation ou de libération oubliée. La compilation séparée a fait l'objet de rappels, mais le jury souligne que celle-ci doit être reconnue par les candidats qui sont donc supposés être un minimum familiers avec ce procédé.

Remarques concernant `SQL`

Le niveau de maitrise est très hétérogène parmi les candidats interrogés. Il est globalement satisfaisant. Le jury attire cependant l'attention sur le fait que plusieurs candidat semblent avoir fait l'impasse sur ce langage. Ces candidats ont tenté de construire des requêtes à partir d'exemples extraits de la documentation mise à disposition sur les machines. Il en a résulté des prestations décevantes :

— cette stratégie a permis de construire les requêtes les plus simples, au prix d'une énorme perte de temps ;

— elle conduit fréquemment à utiliser des éléments hors programme et non compris des candidats. En particulier, certains candidats ont tenté d'utiliser en `SQLite` des fonctions trouvées dans le manuel de `MariaDB`, sans comprendre pourquoi le moteur refusait la requête.

Les sujets proposent des requêtes de niveaux différents. Le jury s'attend à ce que les requêtes les plus faciles puissent être traitées par tous les candidats, notamment en ce qui concerne :

— l'usage des fonctions d'agrégation avec ou sans `GROUP BY` ;

— l'usage de jointures ;

— l'usage de `LIMIT` et `OFFSET`.

Le jury félicite les candidats qui ont réussi à construire des requêtes pour quelques questions particulièrement alambiquées.

Conclusion

Le jury a été globalement été très satisfait par l'ensemble des prestations des candidats. Ces derniers ont été très bien préparés aux modalités de cette épreuve. Le jury tient à nouveau à féliciter l'ensemble des candidats ainsi que leurs enseignants pour leur engagement.

Pour les sessions suivantes, il attire l'attention sur l'importance de la compréhension et de la maitrise du cours, sur la nécessaire prise de recul quant aux notions abordées pendant l'année et sur l'intérêt d'une pratique très régulière sur machine tout au long du cursus.