

Grille d'évaluation du projet de compilation

Pr Sid TOUATI
Université Côte d'Azur

1 Considérations pratiques à tenir en compte

Afin de faciliter la gestion des projets des étudiants, il vous est demandé de respecter chacun des points suivants.

- Le projet est à réaliser en binômes (sauf exception¹).
- Le projet doit être déposé dans l'endroit approprié de l'espace de cours sur moodle <https://lms.univ-cotedazur.fr>, une boîte de dépôt est prévue à cet effet. Pas de soumission acceptée par mél.
- Le rapport en PDF doit être soumis dans une boîte de dépôt différente de l'archive qui contient votre compilateur.
- Le projet doit être programmé et fonctionner sous Linux (distributions Fedora ou Redhat ou Debian, et non pas Ubuntu), avec les outils libres : `gcc`, `flex`, `bison`.
- Votre archive de projet de format **tar compressé** devra être nommée **Projet-Nom1-Nom2.tar.gz** ou **Projet-Nom1-Nom2.tgz**. Votre archive devra se décompresser dans un sous-répertoire nommé **./Projet-Nom1-Nom2**, elle ne doit pas se décompresser dans le répertoire courant.
- Votre projet doit pouvoir fonctionner en ligne de commande. Testez votre archive avant de la déposer. Vérifiez bien que vous n'avez oublié aucun fichier, et que votre système de **Makefile** fonctionne correctement. Il est fréquent que des étudiants travaillent avec des outils graphiques d'aide au développement logiciel, puis ils rendent des projets qui ne fonctionnent pas en ligne de commande.
- Les bits de permissions Unix de vos fichiers doivent être correctes : par exemple, les codes source C ou txt ne doivent pas être exécutables, etc. Vérifiez les permissions de vos fichiers avec la commande `ls -la`, qui permet aussi de vérifier l'absence de fichier caché. Si vous créez votre archive sur un autre système d'exploitation que Linux (par exemple sous Windows ou MacOS) il y a un fort risque que les permissions de vos fichiers soient incorrectes.
- Avant de soumettre votre archive, décompressez la et vérifiez que tout est OK avant de la soumettre. Mettez vous à la place du correcteur pour valider votre archive.

1. Pas de trinôme autorisé. Il est possible de faire le projet en monôme en cas d'impossibilité de binôme, mais cela n'est pas conseillé.

- Dans un binôme, un seul de vous deux doit soumettre l'archive et le rapport. Si vous soumettez une double version, l'enseignant peut les confondre.

2 Quelle distribution Linux favoriser ?

Vous pouvez installer Linux soit en *dual-boot* soit dans une machine virtuelle. La question récurrente des utilisateurs Linux est "Quelle est la distribution Linux idéale ?" Voici mes opinions :

1. Les distributions Fedora, Redhat et Debian sont mieux adaptées aux informaticiens et aux étudiants en science informatique : personnellement j'utilise la distribution Fedora pour tester vos projets. Je vous conseille les distributions Fedora ou Debian. Plus précisément, Fedora est réputée pour les stations de travail des scientifiques du numérique, et la Debian réputée pour les informaticiens programmeurs sous Linux. A vous de faire votre choix !
2. La distribution Ubuntu est destinée au large public : les utilisateurs de bureautique, les non informaticiens, etc. De plus, j'ai constaté en pratique des erreurs avec lex/yacc qui n'arrivent pas dans d'autres distributions. Il vous est déconseillé d'utiliser Ubuntu.
3. La distribution Redhat est bien adaptée pour le travail en entreprise et pour les serveurs en général (Linux pour les professionnels).
4. L'environnement WSL sous Windows qui émule un Linux n'est pas un vrai Linux (WSL est un hyperviseur Linux) : le noyau système qui tourne derrière reste Windows (celui qui a le réel contrôle sur la machine) ainsi que le système de fichiers et les processus système. Des commandes avancées système peuvent ne pas fonctionner et le système de fichiers n'est pas celui de Linux. Cependant cet environnement est toléré pour votre projet de compilation si vous souhaitez, à condition que le projet fonctionne sous Linux par la suite (c'est un risque technique que vous prenez si vous ne vérifiez pas avant).

Il y a d'autres systèmes libres Unix pour PC, notamment FreeBSD qui a une excellente réputation de stabilité, mais elle demande une connaissance technique avancée pour l'installer, la configurer et la maintenir.

3 Contenu de votre archive

L'archive doit contenir ce qui suit :

1. Un fichier `Readme.txt` ou `Lisezmoi.txt` qui donne vos indications à l'enseignant pour compiler et faire fonctionner votre compilateur, notamment préciser les arguments de commande. S'il y a des erreurs ou des problèmes, indiquez-le dans ce fichier.

2. Un fichier **Makefile** est vivement recommandé. Ainsi l'enseignant pourra exécuter **make** en ligne de commande, lex et yacc (ou bison) devront être exécutés pour générer le code C de votre compilateur, qui devra être compilé vers un code exécutable.
3. Tous vos codes sources et exemples de tests.
4. Les fichiers source de votre rapport (format word, ou libreoffice ou latex). La version PDF doit être déposée dans la boîte de dépôt du rapport, qui n'est pas celle de l'archive.
5. Votre archive ne doit pas contenir vos binaires exécutables, c'est une question de sécurité. C'est le correcteur qui devra compiler votre projet afin de générer le binaire exécutable.
6. L'usage traditionnel en science informatique pour générer un exécutable à partir de codes sources est de juste taper la commande **make**. S'il y a autre chose de spécifique à faire, il faut l'indiquer dans le fichier **Readme.txt** ou **Lisezmoi.txt**. Assurez-vous que lorsque votre archive est décompressée, il suffit de taper **make** pour générer un exécutable.

4 Contenu de votre rapport

Le rapport ne doit pas contenir un cours ou un rappel des généralités sur la compilation. Il doit contenir uniquement ce qui est nécessaire au message que vous souhaitez transmettre à l'enseignant. Il doit comporter une introduction qui décrit votre travail en général, ce que vous avez réussi à faire et vos échecs. Ensuite vous développez un peu, notamment vous devrez détailler les schémas de traduction de code (squelette de génération de code comme vus en TD et en cours). Votre rapport doit aussi comporter une conclusion qui décrit ce que vous avez appris, vos sentiments réels (pas inventés), vos pensées avec du recul (pas du blabla pour faire passer le rapport). Le rapport doit expliquer les difficultés rencontrées, les solutions proposées, les structures de données employées. Il doit également décrire comment le binôme s'est organisé et partagé le travail, qui a fait quoi exactement. Si vous avez reçu de l'aide de personnes extérieures (camarades, amis, site internet, forum, etc), indiquez-le. Le rapport doit aussi lister ce qui fonctionne et ce qui ne fonctionne pas dans votre compilateur. Le nombre de pages n'a pas d'importance, c'est la qualité et la précision du contenu qui comptent. Les enseignants savent détecter lorsqu'un étudiant essaye de remplir son rapport avec n'importe quoi.

5 Notation du projet

Le projet sera noté sur 20 points, 5 points pour la soutenance et le rapport et 15 points pour le compilateur. La note du projet comptera pour 50% de la note finale en compilation, l'examen terminal comptera pour les 50% restant.

Concernant la notation de votre compilateur (sur 15 points), ci-dessous quelques précisions :

- Si aucun code source n'est fourni dans l'archive de votre projet, la note sera 0.
- Si le code source de votre compilateur ne compile pas avec gcc, *i.e.* si l'enseignant n'arrive pas à générer l'exécutable de votre compilateur, la note sera en dessous de 7,5.
- Si votre compilateur s'exécute sur des programmes tests, mais il plante (*i.e.* il crache), sa note sera en dessous de 7,5.
- Si votre compilateur s'exécute sur tous les programmes tests sans planter, mais ne génère aucun code (ie votre compilateur est juste un parseur), la note sera en dessous de 7,5.
- Si votre compilateur s'exécute sur des programmes tests, mais génère des codes faux pour certains et des codes corrects pour d'autres, la note sera supérieure à 7,5.
- Si votre compilateur s'exécute correctement sur tous les programmes tests et qu'il génère du code correct pour tous les programmes tests, la note sera 15.

Les notes ci-dessus dépendront aussi de votre aptitude à répondre aux questions durant la démonstration. L'enseignant se réserve le droit de noter différemment les deux membres du binôme s'il a la conviction que le travail n'était pas équilibré. La note peut aussi être revue sensiblement à la baisse si le correcteur s'aperçoit que vos réponses orales ne sont pas en adéquation avec la réalisation du projet soumis.

6 Déroulement de la soutenance et de la démonstration pratique

6.1 Scenario

Chaque binôme passera une séance de 30 minutes avec l'enseignant avec le scenario suivant :

- Nous utiliserons vos PC personnels, nous fournissons le vidéoprojecteur ;
- Soutenance orale de 10 minutes + 5 minutes de questions : le temps de parole doit être équilibré au sein du binôme ;
- Le binôme devra utiliser le fichier archive remis (pas une nouvelle version améliorée la veille) ;
- Démonstration sous Linux de 10-15 minutes comme suit :
 1. Décompresser et compiler le projet devant l'enseignant ;
 2. Exécuter votre compilateur sur des exemples tests ;
 3. L'enseignant modifiera des exemples tests pour vérifier la robustesse de votre compilateur ;
 4. L'enseignant posera des questions techniques sur la réalisation du projet.

6.2 Soutenance

Comme tout contrôle noté, **la présence physique à la soutenance est obligatoire pour le projet de compilation**. Sinon la note du projet sera soit ABI en cas d'absence injustifiée, ou ABJ si l'absence est justifiée. Pour la soutenance, il est demandé de présenter votre travail oralement de façon concise avec des slides : les difficultés rencontrées et comment vous les avez traitées, quelles sont les structures de données employées dans votre compilateur, ce que fait au final votre compilateur et ce qu'il ne fait pas, etc. Il faut aussi expliquer comment vous êtes vous partagés le travail, qui a fait quoi exactement. La soutenance met en oral ce que vous avez mis par écrit dans votre rapport. C'est un moment d'échange intime où il faut être honnête avec l'enseignant. La durée sera de 10 min (5-10 slides) suivie de 5 minutes de questions. Il est recommandé de s'entraîner avant ! La démonstration pratique suivra la soutenance. L'enseignant peut aussi choisir qui d'entre vous deux devra faire la démonstration technique ou répondre aux questions, afin de vérifier que les deux étudiants du binôme ont réellement travaillé le projet.

Voici quelques conseils pour réaliser vos slides :

- Les slides doivent être numérotés, avec un premier slide de titre du projet, intitulé de votre formation, institution et noms des auteurs ;
- Un dernier slide de conclusion qui liste les idées ou points importants que vous souhaitez communiquer en dernier ;
- Un slide est un support pour votre discours, il ne doit pas être trop chargé avec du texte, sinon votre interlocuteur se concentrera sur votre texte écrit au lieu de vous écouter. Méfiez vous des images capturées sur écran avec un format pixelisé à résolution insuffisante, cela ne se zoom pas bien avec un vidéoprojecteur.
- Un bon schéma est souvent un meilleur support pour expliquer vos idées orales.

Je vous souhaite tout le succès.

Pr Sid TOUATI

A Avertissement et plagiat

Toute récupération de code ou de travail d'un autre étudiant (y compris des années précédentes) est considéré comme une fraude. Aussi, il est interdit d'utiliser des outils de génération automatique de code ou de texte (comme les logiciels de `chatGPT` ou copilot). Ceci constituerait une fraude condamnable par le conseil de discipline de l'université, car vous fournissez un projet noté qui n'est pas le fruit de votre travail.

Néanmoins, vous êtes autorisés à communiquer entre vous tous (tous les étudiants), à vous échanger des idées et des astuces, à vous entre-aider pour comprendre et progresser ensemble, à vous documenter sur internet, etc. Les

enseignants sont là aussi pour répondre à vos questions via le forum.

Lorsque vous aidez votre camarade, rappelez vous de l'adage « Donnez un poisson à votre ami, il se nourrira un jour. Apprenez-lui à pêcher, il se nourrira pour toujours. »