

Langages : interprétation et compilation

Université Paris 8 – Vincennes à Saint-Denis
UFR MITSIC / L3 informatique

Séance e (TP) : Programmation MIPS

N'oubliez pas :

- Les TPs doivent être rendus par courriel au plus tard la veille de la séance suivante avec “[liec]” suivi du numéro de la séance et de votre nom dans le sujet du mail, par exemple “[liec] TPe Rauzy”.
- Quand un exercice demande des réponses qui ne sont pas du code, vous les mettez dans un fichier texte **reponses.txt** à rendre avec le code.
- Le TP doit être rendu dans une archive, par exemple un tar gzippé obtenu avec la commande `tar czvf NOM.tgz NOM`, où **NOM** est le nom du répertoire dans lequel il y a votre code (idéalement, votre nom de famille et le numéro de la séance, par exemple “rauzy-tp”).
- Si l’archive est lourde (> 1 Mo), merci d’utiliser <https://bigfiles.univ-paris8.fr/>.
- Les fichiers temporaires (si il y en a) doivent être supprimés avant de créer l’archive.
- Le code doit être proprement indenté et les variables, fonctions, constantes, etc. correctement nommées, en respectant des conventions cohérentes.
- Le code est de préférence en anglais, les commentaires (si besoin) en français ou anglais, en restant cohérent.
- **N’hésitez jamais à chercher de la documentation par vous-même sur le net!**

Dans ce TP :

- Compilation “à la main” de petits programmes Python en MIPS.

Exercice 0.

Récupération des fichiers nécessaires.

1. Pensez à organiser correctement votre espace de travail, par exemple tout ce qui se passe dans ce TP pourrait être dans `~/liec/se-tp/`.
2. Récupérez les fichiers nécessaires depuis la page web du cours, ou directement en ligne de commande avec `wget https://pablo.rauzy.name/teaching/liec/seance-e_tp.tgz`.
3. Une fois que vous avez extrait le dossier de l’archive (par exemple avec la commande `tar xzf seance-e_tp.tgz`), renommez le répertoire en votre nom (avec la commande `mv liec_seance-e_files votre-nom`). Si vous ne le faites pas tout de suite, pensez à le faire avant de rendre votre TP.
4. Vous êtes encouragé à tester systématiquement votre code après chaque modification de code, cela vous aidera à repérer les erreurs au plus tôt et donc le plus précisément possible.

Exercice 1.

Programmes.

1. Structures de contrôle.
→ Recodez en MIPS l’algorithme d’affichage du triangle de Sierpinsky qui se trouve dans le fichier “sierpinsky.py”.
2. Fonctions et appels de fonctions.
→ Recodez en MIPS le jeu “plus petit ou plus grand” qui est implémenté dans le fichier “jeu.py”.
3. Allocations mémoire et listes chaînées.
→ Recodez en MIPS le programme de calcul de moyennes qui est implémenté dans le fichier “moy.py”.
4. Si il vous reste du temps, avancez sur votre projet en n’hésitant pas à me solliciter si besoin :).