

Durée : 2h00

Ce document contient 4 pages.

Consignes générales :

Le barème donné est indicatif. Les exercices sont indépendants et peuvent être traités dans le désordre. **Il est recommandé d'essayer de traiter toutes les questions**, même si vous ne les terminez pas.

Les documents sont interdits, sauf une feuille A4 manuscrite recto-verso. Photocopies et documents imprimés interdits. Tout appareil électronique (*e.g.* ordinateur portable, téléphones, clés USB, *etc.*) interdit. Toute tentative de fraude entrainera une convocation devant la Section Disciplinaire de Grenoble-INP.

Le sous-répertoire docs contient les documents PDF supports du cours, ainsi que les documentations de l'architecture Intel.

Le code que vous écrirez devra être correctement présenté et commenté. La clarté du code et le respect des conventions d'écriture (*coding-style*) seront pris en compte dans la notation.

Dans tous les exercices portant sur l'assembleur Intel, on demande de traduire **systématiquement** du code C en assembleur, comme le ferait un compilateur. Vous ne devez donc pas chercher à optimiser le code écrit et **vous devez placer et lire systématiquement les variables locales dans la pile d'exécution comme vu en TP**, sauf indication contraire dans les questions.

Pour chaque ligne de C traduite en assembleur, vous recopierez en commentaire la ligne en question avant d'implanter les instructions assembleur correspondantes. Vous indiquerez aussi la position par rapport au registre `%rbp` (ou `%ebp` pour le code 32 bits) de chaque variable locale, ainsi que les registres ou les adresses dans la pile contenant les paramètres au début des fonctions implantées. Tous les commentaires additionnels sont bienvenus.

Tout le travail demandé est à rendre dans les fichiers fournis ou que l'on vous demande explicitement de créer : le correcteur ne regardera pas les autres fichiers que vous pourriez ajouter. On fournit des tests basiques pour vous aider à mettre au point vos programmes. On ne vous demande pas d'en ajouter, mais vous pouvez modifier les fichiers distribués pour ajouter vos propres tests si besoin. Pour compiler vos programmes, vous utiliserez le `Makefile` fourni. Si vous voulez nettoyer le répertoire pour tout recompiler à partir de zéro, il suffit de taper la commande `make clean`.

A la fin de l'épreuve, vous devez fermer proprement votre session en cliquant sur l'icône « Sauvegarder et terminer l'examen ». Une fois déconnecté, vous ne pourrez plus vous reconnecter et votre code sera rendu automatiquement. **Attention, vous ne devez surtout pas vous déconnecter via le menu système, au risque de perdre votre travail.** On recommande de sauvegarder régulièrement votre travail grâce à l'icône « Sauvegarder l'examen sans quitter » présente sur le bureau.