

PROG5 - Projet

Guillaume Huard

Université Grenoble Alpes

Travail demandé

Réaliser le chargement d'un exécutable au format ELF

- programme en langage machine ARM en entrée
- lecture, au format ELF, de la structure du programme
- chargement du code aux adresses données

C'est la partie finale d'un éditeur de liens

Squelette fourni

Essentiellement à coder à partir de rien, on vous donne

- des fonctions de debug
- un exemple d'utilisation de `getoptlong`
- un simulateur ARM qui trace l'exécution de ses instructions
- une interface permettant un pilotage de ce simulateur

Code à écrire

Un chargeur est essentiellement constitué des étapes suivantes :

- lecture du fichier à charger (format binaire)
- création et l'initialisation des sections chargeables
- correction du contenu des sections contenant des adresses
- écriture du fichier résultant (format binaire)
- exemples et jeux de tests pour valider tout ça

Objectifs (programmation)

Logiciel de taille moyenne, développement à structurer le chargeur complet fait plus d'un millier de lignes

- environnements de développement croisés
- plusieurs composants (lecture, chargement, écriture, test)

Compétences acquises pendant le semestre à appliquer

- PROG : bas niveau, manipulation de la mémoire, outils
- ALM : langage machine, structure d'un programme

Objectifs (respect d'un cahier des charges)

Appréhender des documentations techniques volumineuses

- documentation technique de base de 80 pages environ
- le manuel de référence ARMv7 fait plus de 2000 pages
- chercher au bon endroit les informations
- respecter une specification

Objectifs (génie logiciel)

Apprendre à gérer le travail au sein d'un groupe de taille moyenne

- gérer le groupe, les compétences et le rythme de chacun
- diviser le travail
- suivre l'évolution
- intégrer l'ensemble
- tester le nouveau code, mais aussi l'ancien (non régression)

Page du projet

lien vers la page du projet sur la page de l'UE

- annonces, à consulter tous les jours
- ressources (sujet, code)
- documentation
- FAQ

Support distanciel

Support par ordre de préférence :

- sur le mattermost, sur le canal de PROG5
- par mail à
`im2ag-l3-info-hotline-s5@univ-grenoble-alpes.fr`

Réfléchir avant d'écrire

- arrive aux enseignants du projet
- temps de réponse selon leur disponibilités

Support présentiel

Dans les salles réservées au projet

- passage quotidien d'un enseignant jusqu'à la veille du rendu
- sauf mention contraire, passage durant le créneau 10h-11h30
- préparez vos questions
- en l'absence de questions l'enseignant repartira

Dépôt git

Un dépôt git sera créé sur votre groupe sur le gitlab de l'UGA :

<https://gricad-gitlab.univ-grenoble-alpes.fr>

Ce dépôt devra être utilisé pour tous les rendus. Tous les rendus devront se faire sur la tête de la branche master.

Modus operandi :

- connectez vous sur la plateforme avec vos identifiants Agalan (cela crée votre compte à la volée, si besoin)
- une fois les groupes formés, un script crée les projets et vous ajoute comme membres du projet adéquat
- à partir du 16/12, si vous n'avez pas été ajouté automatiquement, demandez à un autre membre de votre groupe ou à la hotline de vous ajouter

Jalons

Mi-projet, jeudi 19 décembre 2024

- audit de code, 20mn par groupe, planning à établir

Rendu du projet, jeudi 09 janvier 2024 à 12h

- à pousser sur le dépôt git fourni
- contenant du code qui compile
- contenant aussi tous les documents demandés

Dernière journée, vendredi 10 janvier 2024, soutenances

- 20mn de présentation/démo projetée
- *a priori* sans intervention des enseignants
- 20mn de questions

Evaluation

Présentation orale

- clarté
- respect du temps
- réponse aux questions

Quantité globale de travail

- avancement dans les étapes
- tests
- outils et automatisations

Qualité technique globale

- absence de bug
- qualité du code
- qualité des documents rendus

La soutenance

Préparer et répéter la soutenance

- préparer slides, discours, démo
- respecter le temps imposé
- ne rien oublier de ce qui a été fait
- répartir au mieux le temps de parole

Être positif

- ne rien cacher, mais ne pas insister sur les bugs, manques, ...
- structurer surtout autour de ce qui marche (intro, plan)
- conclure sur l'ensemble du travail réalisé