

Examen de *Environnements et Outils de Développement*

Note : Vous avez 1h30 pour la première partie, 1h30 pour la deuxième. Vous avez droit à tous documents. La première partie est à rédiger sur papier, la deuxième à l'ordinateur. Pour rendre la deuxième partie vous devez **envoyer un mail à zack@pps.univ-paris-diderot.fr** ayant en pièce jointe une **archive tar** du contenu requis, et comme **objet : examen ed6 2013-1**. Vous pouvez consulter le Web pour la deuxième partie mais toutes

formes de communication avec les autres sont interdites.

Première partie

- Exercice 1 (Le processus de compilation)**
1. Décrivez le processus de compilation typique du langage C. Détaillez les différentes phases de "compilation" (au sens large), et expliquez les relations entre eux.
 2. Expliquez l'erreur de liaison typique `undefined symbol`.
 3. Expliquez la différence entre liaison statique et liaison dynamique.

Exercice 2 (Preprocessor) Le fichier `hello.c` contient le code suivant :

```
#include <stdio.h>
#define INCR(x) ((x)++)
#define MULT(x, y) x * y
#ifdef DEBUG
#define INFO(s) printf("INFO: %s\n", s);
#else
#define INFO(s)
#endif
void hello(void) {
    INFO("entering hello")
    printf("Hello, World!\n");
    INFO("leaving hello")
}
```

Exercice 3 (Make) 1. Dans le cadre de Make, expliquez les notions de : règle, cible, prérequis, et commande.

2. Considérez le Makefile suivant :

```
foo: main.o high.o low.o utils.o
```

```
high.o: high.c low.h utils.h
        gcc -c -o high.o high.c low.h utils.h
low.o: low.c utils.h
        gcc -c -o low.o low.c utils.h
utils.o: utils.c utils.h
        gcc -c -o utils.o utils.c utils.h
clean:
        -rm -f foo main.o high.o low.o utils.o
```

donnez le diagramme de dépendances correspondant (i.e. un graphe où chaque noeud

Deuxième partie

Exercice 5 (Édition de texte) Téléchargez le fichier : http://upsilon.cc/~zack/stuff/le_grand_meaulnes.txt. Ce fichier contient les sept premiers chapitres du roman *Le Grand Meaulnes* d'Alain-Frédéric Leiris. Mécanisme : il y a eu quelques petites erreurs qui se sont

glissés dans ce fichier. Votre but est de les corriger. Des indications vous sont données entre `/*` `*/`, vous soumettrez une version corrigée de ce texte en suivant les indications et en veillant à toutes les supprimer.

Exercice 6 (Make) Récupérez l'archive `tar` disponible à <http://upsilon.cc/~zack/stuff/ed6-exam-2013-1.tar.gz>.

Écrivez un `Makefile` pour le projet `C` contenu dans l'archive. L'exécution de `make` doit produire un exécutable qui marche et lie tous les fichiers `*.c` ensemble. À noter : pour la phase de liaison vous devrez lier avec les bibliothèques `libncurses`, `libnsl`, et `libtermcap`.

Comme d'habitude pour un `Makefile` correct, l'exécution de `make` après des changements doit minimiser le nombre d'objets recompilés. Vous devez expliciter toutes les règles (i.e. l'utilisation de règles implicites prédéfinies est interdite) et minimiser la répétition d'information.

Exercice 7 (Git) Produisez un repository `Git` dont l'historique rassemble le plus possible à