

Informations générales sur le CFS

Revision: 1.43

Le CFS est une épreuve individuelle en quatre heures et sur ordinateur comptant pour le contrôle continu transversal (CCT)¹ de licence. Le CFS vise à tester les capacités d'un étudiant à réaliser, seul mais aidé d'un ordinateur, des tâches informatiques mettant en œuvre des connaissances et des savoir-faire en algorithmique, programmation, système, etc.

Ce document décrit l'organisation générale des CFS. Il est publié sur le site de licence bien avant le premier CFS afin que tous les étudiants puissent étudier comment se déroulera cette épreuve.

1 Conditions générales

Les programmes et autres fichiers demandés au cours de cette épreuve seront à fournir via *Delivery_Builder*². Les étudiants composeront sur les machines de second cycle sous des comptes étanches et propres. Les noms de connexion et mots de passe seront conservés. Ces comptes procurent l'environnement standard de licence ainsi, bien sûr, que tous les utilitaires et documentations classiques d'Unix. Quelques autres utilitaires particuliers au CFS pourront être fournis ; ils seront alors décrits dans les énoncés. Les étudiants n'auront pas accès au contenu de leurs précédents comptes personnels, il est donc conseillé de **maîtriser l'environnement standard de licence**.

Delivery_Builder est paramétré pour n'accepter les copies que jusqu'à l'heure de fin de l'épreuve. **Aucune copie ne pourra être rendue en retard**. Vous disposerez d'un délai de grâce d'une minute pour soumettre votre travail lorsque la fin de l'épreuve sera annoncée. Vous pouvez par contre, soumettre autant de réponses que vous le souhaitez à chaque question mais seule la dernière sera prise en compte. Il est conseillé, si vous n'avez jamais utilisé *Delivery_Builder*, de vérifier que vous êtes bien enregistré ainsi que vous familiariser avec. En particulier, **veillez à posséder votre mot de passe³ pour le ouêbe afin de pouvoir soumettre votre copie via *Delivery_Builder* pendant l'épreuve**.

Pour des raisons de fiabilité, un démon archivera le contenu de vos répertoires toutes les quelques minutes. Toujours pour des raisons de fiabilité mais également pour garantir le caractère personnel du travail, un autre démon enregistrera la suite de tous les caractères qui seront frappés au clavier pendant l'épreuve.

Les comptes seront étanches en ce sens que tous les moyens les plus usuels de communication entre utilisateurs seront inactivés pendant la durée des CFS. Les machines étant spécialement configurées, il n'y aura pas de libre service depuis la veille du premier CFS jusqu'au lendemain du dernier CFS.

Les étudiants seront convoqués à l'une des quatre sessions de CFS. Les quatre populations seront statistiquement homogènes afin de pouvoir comparer leur performance (moyenne et écart-type) sur les divers sujets de ces quatre sessions. L'homogénéité est, par exemple, obtenue par classement de tous les étudiants sur la base de leurs notes de DTH et de partiels dans les modules du semestre. Les étudiants convoqués à la *i*-ème session sont ceux qui sont classés avec un rang égal à *imodulo*4.

Les étudiants seront convoqués à l'une des 4 sessions dans l'une des 7 salles de TP de second cycle de l'UFR (bâtiment 31). La convocation sera affichée sur le site de la licence et au secrétariat des étudiants.

¹<http://www.infop6.jussieu.fr/licence/2003/cct/>

²http://www.infop6.jussieu.fr/licence/2003/delivery_builder/

³<http://www.infop6.jussieu.fr/licence/auth/>

Tous les étudiants recevront, au début d'une session, les instructions leur permettant de lire l'URL de l'énoncé de l'épreuve. Les épreuves durent quatre heures, les étudiants ne pourront quitter l'épreuve avant ces quatre heures (prenez un bouquin pour patienter au cas où). L'épreuve commence à une heure précise (9h le matin, 13h30 l'après-midi), il est déconseillé d'arriver en retard. Les étudiants ayant oublié leur nom de connexion ou leur mot de passe, ne sachant pas se connecter ou soumettre via *Delivery_Builder* ne pourront s'en prendre qu'à eux-mêmes (l'assistance du CCE essaiera néanmoins de les dépanner le plus vite possible).

Dans la mesure des places disponibles et seulement 30 minutes après le début de l'épreuve, les étudiants dispensés de CFS pourront suivre l'épreuve pour s'entraîner ; ils devront céder leur place si des étudiants régulièrement convoqués arrivent particulièrement en retard. Ces étudiants surnuméraires recevront une « note potentielle » leur permettant de juger de leur niveau. Cette note potentielle n'est donnée qu'à titre indicatif et n'a strictement aucune valeur légale : elle ne saurait être reportée, capitalisée, transmise par héritage, etc.

L'épreuve consiste à élaborer des programmes ou des fichiers de données qui seront soumis via *Delivery_Builder* qui ramassera le contenu du répertoire de travail.

Le comportement des fichiers à produire est spécifié dans les énoncés. Les fichiers seront principalement notés vis-à-vis du respect du comportement qu'ils doivent exhiber. En conséquence, le choix des langages et techniques de programmation est totalement libre (Ada, C, C++, Caml, Java, Perl, Python, Scheme, sh, Tcl, etc.) à condition qu'ils soient disponibles sur les machines du centre de calcul des étudiants. Si vous souhaitez d'autres langages compatibles des configurations du CCE, veuillez nous en faire part le plus vite possible mais au moins un mois avant les épreuves. Le système d'exploitation (Linux) est néanmoins imposé. L'aspect esthétique des programmes ainsi que leur vitesse ou leur consommation en mémoire pourront être pris en compte.

Vos programmes seront comparés aux nôtres (en comportement, vitesse, consommation de mémoire) ou à ceux d'autres étudiants, ils seront tous exécutés sur vos fichiers de tests ainsi que sur les nôtres ou sur ceux d'autres étudiants. **Le processus est entièrement automatique, vous serez donc particulièrement vigilants et respecterez à la lettre les consignes.**

La notation est anonyme et mécanique. **Aucun traitement manuel de rattrapage ne sera opéré.**

Le barème est indiqué dans l'énoncé ainsi que la nature des tests que nous opérerons sur vos programmes. Un corrigé du CFS sera publié à l'issue des journées de CFS. Un compte-rendu de notation détaillé sera élaboré automatiquement pour chaque « copie » rendue et accessible sur le réseau.

Les salles de TP seront surveillées, les enseignants, parmi les surveillants, répondront aux questions d'interprétation du sujet. Il est bien sûr interdit de fureter dans les répertoires des autres candidats.

Il est rappelé qu'il est interdit d'apporter et/ou de consommer des nourritures ou boissons dans les salles d'ordinateurs, grignoter ou siroter sera cependant toléré dans le couloir.

2 Instructions complémentaires pour les CFS

Aucun document n'est autorisé mais tous les serveurs pédagogiques de l'UFR sont visitables ainsi que toutes les documentations en ligne. **Sachez manipuler des documentations informatiques !**

Il est imposé de créer un sous-répertoire nommé *cfs* (écrit exactement ainsi) directement dans votre *HOME* afin d'y mettre **tous** les fichiers de ce CFS (*HOME* est votre répertoire initial quelque chose comme */users/p6ip01/p6ip1/p6lip333*). C'est ce répertoire que vous indiquerez à *Delivery_Builder*. Ce répertoire doit être lisible par *Delivery_Builder*, pensez à faire :

```
chmod 755 cfs
```

Ne créez point de sous-répertoire pour les différentes questions du CFS, tous les fichiers doivent apparaître dans l'unique répertoire nommé *cfs*.

2.1 Construction d'exécutables

Pour chaque question où vous devez fournir un programme exécutable nommé *x.exe* (*x* sera spécifié par l'énoncé), il vous est demandé de fournir :

- les fichiers sources de ce programme qui devront être dans votre répertoire *HOME/cfs/* (où ils seront ramassés par *Delivery_Builder*)
- un fichier nommé *compil-x.sh* qui devra être un script écrit en *sh* dont le rôle est de créer le programme exécutable (*x.exe*) demandé.

Si l'on suppose, par exemple, que le script *compil-x.sh* doit construire le programme *x.exe* alors nous effectuerons (pour noter votre programme) la construction de ce programme de manière similaire à :

```
rm x.exe
bash < compil-x.sh
if [ -x x.exe ]
then echo Bravo ; exit 0
else echo Echec ; exit 1
fi
```

Notez que tout programme nommé *x.exe* est effacé avant d'être reconstruit. Il est bien entendu que, si la commande précédente ne construit pas le programme exécutable demandé à savoir *x.exe*, la question sera notée zéro.

Exemples

Le script (un simple fichier textuel) demandé se limite probablement à une seule ligne du genre, pour C :

```
gcc -Wall -ansi -pedantic -o x.exe monsource.c
```

pour Ada, deux cas sont possibles.

Premier cas, le nom de l'exécutable à produire est tout en minuscules et peut être considéré comme un identificateur légal pour Ada, alors, ceci suffit pour produire l'exécutable nommé *executable*, le fichier source étant *executable.adb* :

```
gnatmake executable
```

Autre cas : si le fichier source se nomme *monsource.adb* et si l'exécutable doit se nommer *x.exe*, alors ceci convient :

```
gnatmake -o x.exe monsource.adb
```

Pour Java, c'est un peu plus compliqué puisque le produit de compilation n'est pas directement exécutable. Mais ceci devrait faire l'affaire. Supposons que la classe à lancer soit *a.b.X* (la classe *X* du paquetage *a.b*) dans le fichier *a/b/X.class* dont le source en Java est dans le fichier *a/b/X.java*. Soit le script *x.sh* exécutable :

```
#!/bin/sh
java a.b.X "$@"
```

Alors le fichier *compil-x.sh* peut être :

```
# Compilons ...
```

```
javac a/b/X.java
# puis créons l(e script) exécutable
cp x.sh x.exe
chmod a+x x.exe
```

La technique utilisée pour Java est valable pour tous les langages mettant en jeu un interprète (sh, Perl, awk, sed, caml, etc.) Voici un exemple en Perl. Supposons que le script soit dans le fichier *x.pl* (même si la première ligne n'utilise pas la convention dite du *she-bang*) alors le script *compil-x.sh* pourra être :

```
cat > x.exe <<EOF
perl x.pl "$@"
EOF
chmod a+x x.exe
```

Attention en copiant/collant le code précédent, certains navigateurs ajoutent des blancs néfastes en début de ligne. La constante EOF marquant la fin du document en ligne (en jargon *Here doc*) doit impérativement commencer en première colonne.

Il est bien sûr important que vous connaissiez la ligne qui permet d'engendrer un programme exécutable. Observez que la présence de ce script est la rançon de la liberté offerte dans le choix du langage de programmation.

2.2 Conseils divers

Tous les fichiers requis doivent se situer directement dans le répertoire *HOME/cfs/* du compte anonyme que vous utiliserez. Aucun lien symbolique ou physique n'est autorisé.

Si vous modifiez ultérieurement un programme déjà ramassé par *Delivery_Builder*, resoumettez à *Delivery_Builder* la nouvelle version. Seule cette dernière comptera ! Au passage, faites le nécessaire pour connaître *Delivery_Builder* et être connu de *Delivery_Builder*.

Soyez sûr, le jour du CFS, de savoir vous connecter à un ordinateur : soyez sûr de vous rappeler votre mot de passe, apportez un crayon et du papier si cela aide votre réflexion.

2.3 Tests d'exécutables

Vos programmes seront exécutés dans un répertoire quelconque d'une machine quelconque. Ils seront exécutés dans un environnement confiné en temps CPU (30 secondes), en nombre d'octets produits (5 mégaoctets), en nombre de processus engendrés (100), en nombre de fichiers simultanément ouverts (100), en place mémoire occupée (100 mégaoctets). Les fichiers *core* sont limités à 1 kilo-octet.

Chaque test se verra allouer un quantum de temps qui ne saurait être dépassé. Il est par défaut de 30 secondes CPU mais pourra être allongé pour des tests volumineux. Lorsque le programme n'a pas livré toutes les réponses attendues dans le quantum, il est réputé boucler ou bloquer et sera éliminé avec la note zéro.

NOTA: *Pour vous aider, des binaires des solutions peuvent vous être fournis qui vous permettront de comparer, aussi exactement que vous le souhaitez, vos programmes avec les nôtres.*

2.4 Fin d'épreuve

Les réponses sont soumises à *Delivery_Builder* **qui sera arrêté, au plus tard, une minute après la fin officielle de l'épreuve**. Il ne sera donc plus possible de soumettre quoi que ce soit après. Soumettez donc régulièrement et avant les limites fatidiques !

Nous vous demandons de sortir des salles en ordre et rapidement afin de permettre l'installation d'autres étudiants.

3 Documentations

Aucun document personnel n'est permis. Vous ne pouvez utiliser que les documentations⁴ disponibles à partir des machines sur lesquelles vous travaillez.

⁴<http://www.infop6.jussieu.fr/licence/2003/cct/public/docs.php>