GIF-1001: Examen 1

Nom :_____ **IDUL :**_____

Sommaire :

Question 1: (10 points) Les systèmes d'opération

#	Questions
1.1	Le BIOS est en voie d'être remplacer par un programme qui n'est pas qu'une simple version évoluée du BIOS. Quel est le nom de ce programme? (l'acronyme est accepté comme réponse) UEFI ou Unified Extensible Firmware Interface
1.2	Quelle partie du système d'opération est responsable de la gestion des valeurs de paramètres qui sont sauvegardés dans la mémoire de type C-MOS des ordinateurs de type PC? BIOS
1.3	Donnez le nom d'un système d'opération qui ne permet pas au processeur de faire du multitâches (acronyme accepté) DOS
1.4	Donnez deux noms de systèmes d'opération qui permettent de faire du mutitâches (acronymes acceptés) Windows, Mac OS, OS X, Linux, etc…
1.5	Nommez deux (2) types d'éléments de programmation qui interagissent avec le noyau d'un système d'opération? Des pilotes pour le matériel (« drivers »), des services pour les tâches générales, des éléments de gestion de l'usager (interface et message), de éléments de gestion des programmes

Question 2: (12 points) Langages de programmation

#	Questions
2.1	Le code que vous écrivez en langage natif est-il d'abord converti en code intermédiaire ou en code binaire? Justifiez votre réponse. En code intermédiaire d'abord. C'est le compilateur qui traite le code en premier et le compilateur fournit du code intermédiaire. Le code binaire est obtenu à partir du code intermédiaire qui est obtenu à partir du code écrit par l'usager.
2.2	À quoi sert un compilateur? À convertir le code écrit en langage natif, évolué ou assembleur en code intermédiaire (qui pourra être utilisé par l'éditeur de lien pour faire du code binaire)
2.3	Lorsque vous générez une application, lequel des deux ces programmes produit du code binaire : le compilateur ou l'éditeur de liens? L'éditeur de liens
2.4	Quel programme sert à déterminer comment sera utilisée la mémoire de l'ordinateur lors de l'exécution de l'application que l'on compile? L'éditeur de liens
2.5	Pourquoi l'exécution d'un programme interprété est-elle moins rapide que celle de la version compilée de ce même programme? Parce que l'ordinateur doit décoder le programme de façon logicielle plutôt que matérielle.
2.6	Nommez les 2 types de langage de programmation vus en classe et donnez un exemple de langage de programmation pour chacun de ces types (2 parmi les 3 suivants) Langage évolué : Java (ou n'importe quel autre exemple de langage évolué) Langage natif : C (ou Pascal, Basic etc i.e. n'importe quel autre langage natif Langage assembleur : assembleur pour ARM (ou assembleur pour 8086, MASM, etc.)
2.7	Quel type de code retrouve-t-on dans les fichiers qui font partie d'une librairie? Justifiez votre réponse. Code intermédiaire. Les librairies contiennent du code qui est compilé, mais ce n'est pas du code binaire qu'on peut utiliser dans le processeur. C'est donc du code intermédiaire.
2.8	Le langage assembleur ne permet pas au programmeur de faire abstraction du matériel. En est-il de même pour les langages natifs? Justifiez votre réponse. Non, on peut faire abstraction du matériel si on veut parce que le langage natif le permet
2.9	Pourquoi est-il avantageux de générer du code intermédiaire losrqu'on compile des programmes? Ça permet de créer du code qui est portable parce qu'il n'a pas été associé à du matériel par l'éditeur de liens, ça permet d'utiliser plusieurs langages de programmation pour se faire des librairies qu'on assemble comme on veut avec l'éditeur de liens, etc.
2.10	Pourquoi souhaite-t-on faire abstraction du matériel en programmation? Le matériel peut changer et le code n'a pas à être refait, pas de besoin d'être spécialiste en matériel pour

#	Questions
	travailler, etc.
2.11	Une machine virtuelle peut-elle exécuter du code interprété? Justifiez votre réponse. Oui. Une machine virtuelle étant un programme, il suffit que ce programme soit conçu pour décoder le code interprété et le convertir en instruction pour que ça soit possible.
2.12	Quel type de langage de programmation est de nature à faire plus d'abstractions que l'assembleur et moins que les langages évolués? Le code natif

Question 3: (28 points) Architecture des processeurs

#	Questions
3.1	Un compteurs de 16 bits peut-il être utilisé pour adresser tous les registres de 32 bits d'un bloc de mémoire de 32 kilo-octets? Justifiez votre réponse. Oui. Le compteurs peut adresser 2^{16} octets = 64 Ko tandis que la mémoire n'a que 32Ko. Le compteur pouvant identifier plus de registres que n'en contient la mémoire, il peut adresse tous les registres de celle-ci.
3.2	Un processeur qui n'aurait pas d'interactions avec le monde extérieur aurait-il un comportement imprévisible? Justifiez votre réponse. Non. Le comportement du circuit dépendrait uniquement du programme qu'Il exécute. Ce programme étant parfaitement connu à l'avance, on pourrait prévoir toutes les actions du processeur. Pas d'évèenements aléatoires donc comportement prévisible.

Utilisez les termes du tableau suivant pour associer les 12 descriptions qui suivent aux concepts et éléments auxquels elles se rapportent :

Bus d'instructions	**Bits**	**Données**	**Instructions**
Von Neuman	**Registres**	**Mots**	**Compteurs de programme**
Mémoire vive	**Bus de données**	**Harvard**	**Circuits logiques**
Non-et	**Octet**	**Accumulateur**	**Modèle élémentaire**
Complément 2	**Microprocesseurs**	**Assembleur**	**Autres**
Compteurs	**Tampons**	**Adresses**	**Mémoire non volatile**

Tableau des termes à copier au besoin dans la colonne « Concept » du tableau suivant

#	Description	Concept :

#	Description	Concept :
3.3	Élément électronique séparés du bus de données pour les architectures de Harvard	bus d'instructions
3.4	Élément électronique dans lequel se retrouvent les données produites par un programme	mémoire vive
3.5	Type d'architecture qui permet d'avoir un bus de données qui compte plus de bits que le bus d'instructions	Harvard
3.6	Il y en a autant dans une mémoire que le nombre de registres qui lui servent de cases mémoire.	Adresses
3.7	Utilisé pour mémoriser des valeurs	Registre
3.8	Permet de connaître l'adresse de l'instruction qui est en train d'être exécuté par un processeur.	compteurs de programme
3.9	Éléments qui ne peuvent pas se retrouver sur le bus d'instructions d'un processeur de type Harvard	Données
3.10	Registre au rôle central pour les processeurs du même type que les 8086 de Intel.	Accumulateur
3.11	Registres accompagnés d'un circuit logique et d'un tampon qui, sur commande, peut augmenter la valeur de son contenu par 1	compteurs
3.12	Élément électronique qui permet de conserver de l'information même lorsque l'ordinateur n'est pas sous tension.	mémoire non volatile
3.13	Type d'architecture qui accède aux données et aux instructions à l'aide d'un même bus.	Von Neuman
3.14	Représentation simplifiée de l'architecture d'un processeur	modèle élémentaire

Tableau des questions 3.3 à 3.14 et des réponses associées

Question 4: (4 points) Représentation des nombres

Répondez aux questions suivantes qui portent sur la représentation des nombres sachant que vous disposez des symboles « 0 » et « # » pour représenter des **nombres formés de 4 chiffres chacun** et les poids associés aux symboles sont tels que **« 0 » a un poids de 0 tandis que « # » a un poids de 1.**

#	Questions
4.1	Donnez la représentation à 4 chiffres des nombres suivants en utilisant les symboles 0 et # 0 =__0000_____ 3 =__00##_____ 6 =__0##0_____ 12 =__ ##00_____ 9 =__#00#_____ 4 =__0#00_____ 7 =__ 0###_____ 10 =_ #0#0_____
4.2	Considérant que « -2» se trouve à trois comptes de «0000» quand on utilise 4 chiffres pour représenter un nombre signé, indiquez comment vous représenteriez -2 à l'aide des symboles appropriés et justifiez votre réponse. 16 possibilités et 14 est à -2 de 16 et 14=8+4+2 ou ###0 d'où -2 qui vaut ###0 (on peut

#	Questions
	procéder par complément 2 -2 = __###0_____

Question 5: (20 points) Nombres en binaire

Répondez aux questions suivantes qui portent sur la représentation binaire des nombres.

#	Questions
5.1	Quel est le complément 2 de la représentation à 16 bits du nombre 0? (la réponse suffit) 0000 0000 0000 0000
5.2	Quel est le complément 2 du nombre écrit en binaire 111111111111? (la réponse suffit) 0000 0000 0001
5.3	Quel est le complément 2 de 10010110100? (la justification n'est pas requise) 01101001100
5.4	Trouvez la somme de ½ + 4 + 1/4 +2 et donnez la représentation binaire à 8 bits de cette somme en prenant soin d'utiliser un point pour séparer la partie entière exprimée avec 4 bits de votre réponse de sa partie fractionnaire représentée avec 4 bits 0110.1100
5.5	Donnez le complément 2 à 8 bits de 4 et donnez le résultat de la conversion de ce résultat en représentation à 16 bits équivalente (2 réponses requises) complément 2 : 11111100 représentation à 16 bits : 1111111111111100
5.6	Que faut il faire pour convertir un nombre positif représenté en binaire à l'aide de 8 bits en nombre équivalent représenté par 16 bits? On ajoute 8 zéros à la gauche du nombre de 8 bits (ex. 0000 0100 devient 0000 0000 0000 0100)
5.7	La représentation à virgule flottante à l'aide de 32 bits permet-elle de représenter plus de valeurs que la

#	Questions
	représentation binaire d'un nombre entier à l'aide de 32 bits? Jutifiez votre réponse. Non. La représentation à virgule flottante permet au mieux de représenter le même nombre de valeurs parce que ce nombre dépend du nombre de bits. Ne pas accorder de points s'il est indiqué que la représentation à virgule flottante couvre une plage plus grande que l'autre.
5.8	Quelle différence y a-t-il entre une retenue et un dépassement? Les retenues sont obtenues quand on dépasse le plus grand nombre qui peut être représenté par un des chiffres qui représentent une valeur tandis que les dépassements sont des retenues qui sont obtenues lorsqu'on dépasse la plus grande valeur qui peut être représentée par un ensemble de chiffre
5.9	Y a-t-il un avantage à préférer la représentation à virgule flottante vue en classe à la représentation binaire à 32 bits d'un nombre quand on sait qu'on peut vouloir représenter le résultat d'une division par zéro? Justifiez votre réponse Oui. La représentation à virgule flottante comprend une façon de représenter le résultat d'une division par zéro tandis que la représentation binaire n'en a pas.
5.10	Y a-t-il un avantage à utiliser un classement de type « big endian » plutôt que de type « little endian » lorsque toutes les valeurs à sauvegarder en mémoire se situent entre -128 et 127? Justifiez votre réponse. Non. Si toutes les valeurs se situent entre -128 et 127, chacune d'elles est représentable avec un seul octet. La question de savoir dans quel ordre il est préférable de sauvegarder les octets ne se pose donc pas puisqu'il n'y a qu'un seul octet à mettre en ordre. Donc pas d'avantage.

Question 6: (9 points) RISC vs. CISC

Répondez aux questions suivantes qui portent sur les RISC et les CISC.

#	Questions
6.1 2	Donnez 3 arguments qui jouent en faveur de l'orientation RISC lorsqu'on la compare à l'orientation CISC Permet de simplifier l'architecture du processeur, Jeu d'instructions plus simple donc plus facile à maîtriser Permet de réduire la consommation d'énergie grâce aux simplifications qui peuvent être apportées à l'électronique qui n'a pas à décoder des instructions complexes. D'autres réponses sont possibles
6.2 2	Est-ce qu'un processeur de type CISC peut aussi voir son architecture être de type Harvard? Justifiez votre réponse. Oui, le choix de disposer d'un jeu d'instruction complexe peut se faire sans égards à l'architecture du microprocesseur. D'autres réponses du genre sont possibles
6.3 2	Un processeur qui ne reconnaitrait pas d'instruction qui lui permettrait de faire une division serait-il de type RISC ou CISC? Justifiez votre réponse. RISC On peut penser que l'absence de l'instruction « division » résulte d'un désir de garder le nombre d'instructions aussi petit que possible comme le veut la philosophie RISC D'autres réponses du genre sont possibles
6.4 3	Donnez 2 caractéristiques techniques qui supportent l'idée qu'un processeur puisse être de type CISC plutôt que de type RISC. Beaucoup de registres au rôle interchangeable, instructions binaires qui contiennent des extensions, instructions toutes de la même longueur, nombre limité d'instructions. D'autres réponses sont possibles
6.5	Donnez 2 caractéristiques techniques qui supportent l'idée qu'un processeur puisse être de type ~~CISC plutôt que de~~ RISC La question n'a pas été corrigée Instructions de longueurs variables, instructions spécialisées, registres spécialisés

Question 7: (7 points) Processeurs ARM et Intel

Répondez aux questions suivantes qui portent sur les processeurs ARM et Intel

#	Questions
7.1	Donnez 3 caractéristiques techniques vues en classe qui font que les processeurs ARM se distinguent des processeurs de type x86 ou de leurs dérivés. Pas d'accumulateur, beaucoup de registres au rôle interchangeable, instructions de la même longueur, jeu d'instructions simplifiés (d'autres réponses sont possibles)
7.2	Le fonctionnement des processeurs Intel de type x86 repose-t-il sur l'emploi d'un accumulateur? Oui (justification pas requise)
7.3	Les processeurs de type ARM permettent-ils d'écrire une valeur qui ferait partie d'une instruction dans la mémoire sans qu'un registre soit impliqué dans l'opération? Justifiez votre réponse Non. L'écriture en mémoire implique toujours un registre pour les ARM
7.4	À quoi peuvent servir les bits « overflow » des registres de statut des processeurs ARM et Intel? Indiquer que le résultat d'une opération a dépassé la valeur maximale qui peut être représentée par un registre. Autres réponses possibles (ex. indiquer que le résultat de l'opération n'est pas valide)
7.5	Diriez-vous que les processeurs de type ARM d'Intel sont des CISC ou des RISC? Justifiez votre réponse. RISC. L'architecture des ARM respectent pas les grandes orientations des RISC puisqu'elle comprend plusieurs registres aux rôles interchangeables, peu d'instructions spécialisées, des instructions qui sont toutes de la même longueur (pour un même mode). D'autres justifications sont possibles.
7.6	Pourquoi les processeurs ARM sont-ils conçus de manière à supporter les jeux d'instructions ARM et Thumb? Dans certain cas, s'obliger à n'utiliser que des instructions qui sont longues de 32 bits conduit à un très grand gaspille de mémoire. En supportant le jeu d'instruction Thumb qui ne comprend que des instructions de 16 bits, on se donne la possibilité d'économiser bien de la mémoire. D'autres réponses sont possibles
7.7	La philosophie qui oriente la conception d'un processeur de type RISC a-t-elle un impact sur le jeu d'instructions que ce processeur reconnait? Justifiez votre réponse. Oui, la philosiphie RISC préconise l'usage d'un nombre restreint d'instructions et l'optimisation de celles-ci. L'orientation implique aussi des simplifications au niveau des registres qui font qu'on ne peut directement faire des opérations sur le contenu de la mémoire. D'autres réponses sont possibles

Question 8: (9 points) Notions liées aux outils

Répondez aux questions suivantes qui portent sur les notions associées aux outils utilisés pour les travaux pratiques en considérant que la mise en marche d'un environnement de programmation Eclipse/Android vous a demandé d'installer le programme "NDK-BUILD" qu'il vous faut appeler lorsque vous apportez certaines modifications à votre code et que vous voulez générer une application qui utilise ce code modifié.

#	Questions
8.1	À quoi sert le programme Eclipse? Editeur de code, gestion de fichier, interface usager pour faciliter l'appel des compilateurs. D'autres réponses sont
8.2	Quel type de m ILD? Les changemer es équivalentes possibles.
8.3	Que se produit ication et que vous utilisez les commande programme NDK-BUILD? Les changemer de intermédiaire associé au code natif n'aura pas été mis à jour.

Pas à faire, pas corrigé