UNIVERSITÉ GRENOBLE ALPES
Licence de Sciences et Technologie - Parcours INM, MIN, Min-Int.

Contrôle Continu UE INF401 : Architectures des ordinateurs

Mars 2024, durée 1 h 30
Document : 1 A4 R/V personnel manuscrit autorisé ; calculettes et téléphones portables interdits.
La plupart des questions sont indépendantes, si vous avez du mal avec l'une, passez à la suivante.
Tant que possible, indiquez bien tous les détails et justifiez vos réponses.
Le barème est donné à titre indicatif.

1 Numération, opération en base 2 et en complément à 2 (6 points)

Rappel important : en machine, les entiers relatifs ($\mathbb{Z}$) sont représentés selon la méthode dite du complément à 2 (C2), il ne faut pas confondre cette méthode de représentation avec l'opération dite de complémentation. Pour les entiers naturels, la base 2 est utilisée.

Questions :

(a) Donner la représentation sur 16 bits de l'entier naturel 5847 et de l'entier relatif -47. **(2 points)**

(b) Donner la représentation binaire et la valeur décimale des 2 entiers relatifs suivants donnés sous forme hexadécimale : $(1024)_{16}$ et $(FACE)_{16}$. **(2 points)**

(c) Effectuer, en posant l'addition comme habituellement et en écrivant toutes les retenues, l'opération binaire suivante sur 1 octet :

$$\begin{array}{rr} & 0101\ 1001 \\ + & 0100\ 1011 \\ \hline = & \end{array}$$

Donner les 2 interprétations usuelles possibles de cette addition selon que les octets sont des entiers naturels ou relatifs. En déduire, la valeur des indicateurs (Z, N, C et V) en expliquant le sens de ces indicateurs. **(2 points)**

2 Tours de Hanoï (8 points)

Cet exercice a pour objectif de résoudre le problème des tours de Hanoï en itératif, pour 5 niveaux.

Remarque : la solution s'étend à un nombre quelconque de niveaux.

```
Lexique :
  int tours[3]={31,0,0};
  int avant[3]={2,0,1};
  int apres[3]={1,2,0};
  int tr,max,min,top;
```

```
Debut
00. tr=1;
01. tant que tours[2] != 31 faire :
02.     si tours[avant[tr]] < tours[apres[tr]] alors
03.               max=apres[tr];  min=avant[tr];
04.     sinon  max=avant[tr]; min=apres[tr]; fin si
05.     afficher(max, ''->'', min)
06.     top = 16
07.     tant que  (tours[max] | top) != tours[max] faire
08.            top=top>>1
09.     fin tant que
10.     tours[max]=tours[max]-top; tours[min]=tours[min]+top;
11.     tr = tours[tr]
12. fin tant que
Fin
```

Dans l'algorithme, toutes les variables et tous les éléments de tableau sont des entiers relatifs sur 1 mot. Les tableaux sont placés en mémoire et pré-initialisés selon les valeurs données. Les variables sont placées dans les registres 5 (tr), 6 (max), 7 (min), 8 (top). Les affichages sont prévues avec les fonctions `EcrChaine`, `EcrZdecimal32` et `ALaLigne` correspondant aux fonctions du fichier `es.s` étudiées en cours et en TP. Le fonctionnement des fonctions d'`es.s` est rappelé en annexe du sujet. Ligne 7, le symbole "|" indique un `ou` bit à bit, le symbole "»1" indique un décalage arithmétique à droite d'1 bit.

Vous allez maintenant traduire le programme en assembleur `ARM` en vous basant sur le squelette de code donné ci-dessous.

```
  .data
       @ déclaration des chaînes de caractères
       @ déclaration des tableaux initialisés
  .text
  .global main
main: push {lr}
      @ votre programme
  pop {lr}
  bx lr
```

Questions :

(d) Donner le code `ARM` avec balign si nécessaire pour ajouter une chaine de caractères "->" dans la zone de données avec un accès dans la zone text. **(1 point)**

(e) Donner le code `ARM` pour la définition du premier des 3 tableaux initialisés (tableau `tours` zone data et zone text.) **(1 point)**

(f) Donner le code `ARM` pour l'affichage ligne 5. **(1 point)**

(g) Donner le code `ARM` pour la première des 2 affectations ligne 10. **(1 point)**

(h) Donner le code `ARM` pour la boucle interne lignes 7-9. **(2 points)**

(i) Donner le code `ARM` pour la conditionnelle lignes 2-4. **(2 points)**

3 Codage Quoted-Printable (6 points)

`Wikipédia dixit` : Quoted-Printable (QP) est un format d'encodage de données codées sur 8 bits, qui utilise exclusivement les caractères alphanumériques imprimables du code ASCII (7 bits).

En effet, les différents codages comprennent de nombreux caractères qui ne sont pas représentables en ASCII (par exemple les caractères accentués), ainsi que des caractères dits "non-imprimables".

L'encodage Quoted-Printable permet de remédier à ce problème, en procédant de la manière suivante :

— Un octet correspondant à un caractère imprimable de l'ASCII sauf le signe égal (donc un caractère de code ASCII entre 33 et 60 ou entre 62 et 126) ou aux caractères de saut de ligne (codes ASCII 13 et 10) ou une suite de tabulations et espaces non situées en fin de ligne (de codes ASCII respectifs 9 et 32) est représenté tel quel.

— Un octet qui ne correspond pas à la définition ci-dessus (caractère non imprimable de l'ASCII, tabulation ou espaces non suivies d'un caractère imprimable avant la fin de la ligne ou signe égal) est représenté par un signe égal, suivi de son numéro, exprimé en hexadécimal.

Enfin, un signe égal suivi par un saut de ligne (donc la suite des trois caractères de codes ASCII 61, 13 et 10) peut être inséré n'importe où, afin de limiter la taille des lignes produites si nécessaire. Une limite de 76 caractères par ligne est généralement respectée, ceci afin d'assurer la compatibilité avec certains logiciels de messagerie limités dans la longueur de ligne qu'ils peuvent gérer.

Par exemple, le caractère "é" est encodé en latin-9 par un octet valant 233, qui s'écrit "E9" en hexadécimal. Le caractère "é" sera donc représenté par "=E9" sous le format Quoted-Printable. Si l'on utilise plutôt l'encodage UTF-8, le caractère "é" est encodé par deux octets de valeur hexadécimales C3 et A9, que le format Quoted-Printable représente par "=C3=A9". `Fin de l'extrait.`

Remarques ou compléments d'explications :

— derrière le signe égal (=), le code est toujours donné en hexadécimal avec 2 chiffres ou lettres en majuscule, même pour les nombres inférieurs à 16.

— la limite de 76 caractères (affichables) par ligne encodée permet de limiter la taille des lignes des textes encodés par Quoted-Printable pour la visualisation dans les messageries ou ailleurs. Ainsi la visualisation de textes encodés avec le format Quoted-Printable ne nécessite pas de gestion d'un affichages spécial pour les lignes longues.

— la limite à 76 caractères (affichable), pour les lignes encodées avec l'insertion d'un passage à la ligne "=AD' (et les autres lignes similaires se terminant par "=AD"), compte le caractère "=" parmi les 76 caractères affichables, mais pas les caractères de passage à la ligne (codes ASCII 10 et 13, en hexadécimal, les codes A et D). Ces lignes de 76 caractères (affichables) peuvent donc occuper jusqu'à 78 octets dans les fichiers.

Questions :

Un fichier `FInit` est composé avec tous les octets possibles (placés dans l'ordre croissant, en commençant par l'octet 0, interprétés comme des octets d'entiers naturels donnés en base 2). Il est ensuite encodé selon le format décrit par le texte de Wikipédia pour être transmis par messagerie (avec la limite de 76 caractères par ligne). Nous appelons ce fichier encodé `FEncoded`.

(j) Comment le fichier `FInit` apparait-il avec un éditeur hexadécimal comme hexdump (donner le début et la fin de l'affichage) ? **(0.5 point)**.

(k) Quelle est la taille précise du fichier initial `FInit` ? **(0.5 point)**

(l) Donner le début (première ligne) et la fin du fichier encodé `FEncoded`. **(1.5 point)**

(m) Quelle est la taille précise du fichier encodé `FEncoded` ? Expliquez votre résultat si nécessaire. **(1.5 point)**

(n) Tous les fichiers ayant une taille initiale identique à la taille du fichier initial `FInit` ont-ils la même taille sous forme encodée ? Pourquoi ? Si les tailles peuvent varier, dans quel intervalle ? **(1 point)**

(o) Le fichier initial et le fichier encodé ont-ils la même taille si l'ordre des octets du fichier initial est modifié ? Pourquoi ? Si les tailles peuvent varier, dans quel intervalle ? **(1 point)**

ANNEXES

Fonctions d'entrée/sortie

Nous rappelons les principales fonctions d'affichages du fichier `es.s` :

- `bl ALaLigne` provoque un passage à la ligne dans l'affichage.
- `bl EcrChaine` affiche la chaîne de caractères dont l'adresse est dans `r1`.
- `bl EcrZdecimal32` affiche l'entier relatif représenté sur 32 bits dont le code est dans `r1`.

Principales instructions du processeur ARM

Code	Nom	Explication du nom	Opération	Remarque
0000	`AND`	AND	et bit à bit	
0010	`SUB`	SUBstract	soustraction	
0100	`ADD`	ADDition	addition	
1000	`TST`	TeST	et bit à bit	pas `rd`
1010	`CMP`	CoMPare	soustraction	pas `rd`
1100	`ORR`	OR	ou bit à bit	
1101	`MOV`	MOVe	copie	pas `rn`
	`Bxx`	Branch	branchement conditionnel	xx = condition
	`LDR`	LoaD Register	lecture mémoire	
	`STR`	STore Register	écriture mémoire	

L'opérande source d'une instruction `MOV` peut être une valeur immédiate notée `#5` ou un registre noté `Ri`, `i` désignant le numéro du registre. Il peut aussi être le contenu d'un registre sur lequel on applique un décalage de `k` bits ; on note `Ri, DEC #k`, avec `DEC` $\in$ {`LSL`, `LSR`, `ASR`, `ROR`}.

Principaux codes conditions du processeur ARM

cond	mnémonique	signification	condition testée
0000	EQ	égal	Z
0001	NE	non égal	$\overline{Z}$
1010	GE	$\geq$ dans Z	$(N \wedge V) \vee (\overline{N} \wedge \overline{V})$
1011	LT	$<$ dans Z	$(N \wedge \overline{V}) \vee (\overline{N} \wedge V)$
1100	GT	$>$ dans Z	$\overline{Z} \wedge ((N \wedge V) \vee (\overline{N} \wedge \overline{V}))$
1101	LE	$\leq$ dans Z	$Z \vee (N \wedge \overline{V}) \vee (\overline{N} \wedge V)$