

Organisation des ordinateurs
Examen de seconde session 2019
Énoncés et solutions

Énoncés

- [2/20] 1. (a) Une machine est constituée de deux récipients opaques R_1 et R_2 entre lesquels sont réparties mille billes identiques. Chaque bille a une probabilité égale de se trouver dans R_1 ou dans R_2 , indépendamment des autres billes. La machine comprend également une balance de précision située sous R_1 , affichant en permanence le nombre de billes que ce récipient contient. Si la balance affiche 0, quelle quantité d'information fournit-elle sur l'état de la machine ? (Donner la réponse sous forme numérique, et justifier votre calcul.)
- [2/20] (b) Une station météorologique est équipée de deux capteurs : un thermomètre affichant la température par pas de $0,5^\circ$, et un pluviomètre renseignant la quantité de précipitations en millimètres avec un chiffre après la virgule. On suppose que la température est indépendante des précipitations, que la température affichée suit une distribution uniforme entre -20° et 40° , et que la valeur renseignée par le pluviomètre suit une distribution uniforme entre 0 et 100 mm. Quelle quantité de mémoire, exprimée en octets, est-elle nécessaire pour enregistrer le bulletin météorologique de cette station ? (Donner la réponse sous forme numérique, et justifier votre calcul.)
- [2/20] (c) Dans les circuits d'un ordinateur, pourquoi préfère-t-on utiliser des signaux discrets plutôt que continus ? Pourquoi choisit-on habituellement des signaux possédant deux valeurs nominales ?
- [2/20] 2. (a) Représenter le nombre -16 sur 8 bits par valeur signée et par complément à deux.
- [1/20] (b) Représenter le nombre $-1,25$ en virgule fixe par complément à deux, avec 4 chiffres avant la virgule et 4 chiffres après la virgule.
- [1/20] (c) Représenter le nombre 2^{-128} dans le format IEEE754 en simple précision, en exprimant le résultat en hexadécimal.
- [1/20] (d) Exprimer la valeur du nombre entier dont la représentation par complément à deux forme la suite de bits $b_3b_2b_1b_0$.
- [1/20] (e) Effectuer l'addition $(-2) + (-5)$ en représentant les nombres par complément à un sur 4 bits.
3. Écrire des fragments de code assembleur x86-64 réalisant les opérations suivantes. (On supposera chaque fois que la pile est initialement correctement

configurée.)

- [1/20] (a) Permuter les deux valeurs de 64 bits situées au sommet de la pile.
- [1/20] (b) Appeler 100 fois une fonction `f`, sans lui fournir d'argument, en employant la convention d'appel des systèmes *Unix*.
- [2/20] (c) Déterminer si le contenu du registre AL correspond au premier octet d'un caractère représenté sur 16 bits selon l'encodage UTF-8. À l'issue de cette opération, AH vaudra 1 dans le cas positif, et 0 sinon.
4. On souhaite programmer une fonction `count_ok(str)` comptant le nombre d'occurrences de la chaîne de caractères "ok" dans une chaîne `str`. Cette dernière contient des caractères encodés en ASCII, et est terminée par un zéro.
- [1/20] (a) Écrire, en pseudocode ou en langage C (au choix), un algorithme permettant de résoudre ce problème.
- [3/20] (b) Traduire cet algorithme en assembleur x86-64, en veillant à respecter la convention d'appel de fonctions des systèmes *Unix*.

Exemple de solutions

1. (a) Pour que la balance affiche 0, toutes les billes doivent se trouver dans R_2 , et la probabilité que cela se produise vaut

$$\left(\frac{1}{2}\right)^{1000} = \frac{1}{2^{1000}}.$$

La quantité d'information correspondante est donc égale à

$$\log_2 2^{1000} = 1000 \text{ bits.}$$

- (b) Il y a 121 valeurs possibles de la température, toutes équiprobables, et 1001 valeurs possibles pour la quantité de précipitations, également équiprobables. Ces deux données sont indépendantes. La quantité d'information totale vaut donc

$$\begin{aligned} \log_2 121 + \log_2 1001 &\approx 16,886 \text{ bits} \\ &\approx 2,111 \text{ Bytes.} \end{aligned}$$

- (c) Les signaux discrets permettent d'éliminer l'influence du bruit, à condition que celui-ci soit suffisamment petit. Les signaux binaires sont ceux qui tolèrent la plus grande quantité de bruit. Ils présentent également l'avantage d'être faciles à générer et à décoder.

2. (a) — Par valeur signée : 10010000.

— Par complément à deux : 11110000.

- (b) Il s'agit de la représentation par complément à deux sur 8 bits du nombre $-1,25 \times 2^4 = -20$, c'est-à-dire 11101100.
- (c) Pour obtenir une mantisse normalisée, l'exposant devrait être égal à -128 , ce qui n'est pas possible en simple précision. L'exposant sera donc fixé à -127 , et la mantisse sera dénormalisée et égale à

$$\frac{2^{-128}}{2^{-127}} = \frac{1}{2}.$$

On obtient donc la représentation

0 00000000 010000000000000000000000,

dont l'écriture hexadécimale est

0x00200000.

- (d) $-8b_3 + 4b_2 + 2b_1 + b_0$.

- (e)

$$\begin{array}{r} \boxed{1} \\ 1 1 1 \\ + 1 1 0 \\ \hline 0 1 1 \\ + 1 \\ \hline 1 0 0 \end{array}$$

3. (a) POP RAX
 POP RBX
 PUSH RAX
 PUSH RBX
- (b) MOV RBX, 100
boucle: CALL f
 DEC RBX
 JNZ boucle
- (c) MOV AH, 0
 AND AL, 0xE0
 CMP AL, 0xC0
 JNE fin
 MOV AH, 1
fin: ...

```

4. (a) unsigned count_ok(char *str)
    {
        unsigned nb;

        for (nb = 0; *str; str++)
        {
            if (str[0] != 'o')
                continue;

            if (str[1] == 'k')
            {
                nb++;
                str++;
            }
        }

        return nb;
    }

```

```

(b)          .text
count_ok:   PUSH    RBP
            MOV     RBP, RSP
            MOV     EAX, 0
boucle:     CMP     byte ptr[RDI], 0
            JE      fin
            CMP     byte ptr[RDI], 'o'
            JNE     continue
            CMP     byte ptr[RDI + 1], 'k'
            JNE     continue
            INC     EAX
            INC     RDI
continue:   INC     RDI
            JMP     boucle
fin:        POP     RBP
            RET

```