2025年度
コンピュータとプログラミング
中間テスト 解答例

1

(1)(d)

- 1命令を表す機械語コードは可変長
- 1命令の実行でのripレジスタの増分は様々
- 機械語コードは64の倍数ではないアドレスにも配置可能

(2)(b)

- 4つのフラグは全部rflagsレジスタの中
- call命令で積まれるのはリターンアドレス(call命令の次の命令のアドレス)
- 商が64ビットに収まらないと例外が発生して実行終了

(3)(a)

- nop命令に対しても機械語コードを生成

2

(1) 4100 ($4*16^3 + 4*16^0 = 4100$)

(2) −120
（1を減算してビット反転して符号を負に）
- 0xff88 → 0xff88 − 1 = 0xff87
  → 0x0078 = 120 → −120

(3) 0x16d ($1*16^2 + 6*16^1 + 13*16^0 = 360$)
- 0x016dでもよい

(4) 0xfdf8
（符号を正にしてビット反転して1を加算）
- −520 → 520 = 0x208 → 0xfdf7 → 0xfdf8

3

(1) 0x950c
- 0x000000000000950cでもよい
- ボロー(繰り下がり)の1を引く必要あり

(2) 0xfe23ba67
- rbxレジスタの値の1が立っているビットの場所を反転する

(3) 0x4433221100000000
- 想定する環境はlittle endianなので, メモリアドレスが小さいほうにあるバイトが, 読み込んだ整数の下位のバイトに来る

(4) 4
- 3で割る関数fを5回呼び出して, 999→333→111→37→12→4

4

(1)(a) mov X(, %rcx, 8), %rdx
　　(b) jg L1 または jge L1

(2) pop %rbx

(3) rep stosq

5

(1) **`grep network config.txt | wc -l > ./result.txt`**

- `./` はなくてもよい

(2) and $0xff, %rax と shr $56, %rax

(3) 2506命令

- L1に入る前に3命令
- L1からLf直前までの5命令を500回繰り返して2500命令
- 最後にcmp命令とjg命令とcall命令の実行で3命令

6

- たとえばこれ(15命令)
  - もっと少ない命令数で記述できるかもしれない
- 甘めに採点します. 細かい部分にミスがあっても, だいたい合っていれば加点します

```
maxseqlen:
          mov $0, %rcx
          mov $0, %rax
L1:       cmp $0, %rdi
          jz Lf
          test $1, %rdi
          jz L2
          inc %rcx
          cmp %rax, %rcx
          jle L3
          mov %rcx, %rax
          jmp L3
L2:       mov $0, %rcx
L3:       shr $1, %rdi
          jmp L1
Lf:       ret
```