

情報セキュリティ大学院大学  
情報セキュリティ研究科（博士前期課程）情報セキュリティ専攻  
2013年度特待生選抜試験問題

1次選考（筆記試験）

10:00～11:30

- (1)
- I 情報数学 A
  - II 情報数学 B
  - III 通信ネットワーク
  - IV 情報システム
  - V ソフトウェア
- (2)
- 小論文

【注意事項】

1. 指示があるまで、この問題冊子を開いてはならない。
2. この問題冊子の本文は全部で12ページある。落丁、乱丁があれば申し出ること。
3. (1)、(2)のいずれかを選択し、答案を作成せよ。ただし、技術系の研究テーマを希望する受験者は(1)を選択すること。
4. (1)を選択した受験者は、上記I～Vの5項目から2項目を選択し、解答すること。5項目中どの2項目を選択してもよい。  
(2)を選択した受験者は、与えられた課題について、2000字以上3000字以内の小論文を作成すること。
5. 解答用紙は計3枚（(1)用解答用紙2枚、(2)用解答用紙1枚）配布される。  
(1)を選択した受験者は、「筆記試験(1)用解答用紙」を、選択した項目ごとに1枚ずつ使用すること。必要があれば裏面を使用してよい。筆記試験(2)用解答用紙には何も記入しないこと。  
(2)を選択した受験者は、「筆記試験(2)用解答用紙」1枚のみを使用すること。筆記試験(1)用解答用紙には何も記入しないこと。  
同一受験者が(1)、(2)両方に解答した場合、いずれの解答用紙も無効となるので注意すること。
6. 解答用紙の指定欄に、選択した項目名（「ローマ数字+科目名」※(1)を選択した受験者）、受験番号（全受験者）を必ず記入すること。解答用紙の回収前に、これらを記入したかを必ず確認すること。
7. 問題冊子、解答用紙、計算・下書き用紙は持ち帰ってはならない。

# I 情報数学 A

(問1)  $\alpha > 0$  とする。  $n$  を 2 以上の整数とすると、次の不等式を証明せよ。

$$(1) \int_1^{n+1} \frac{dx}{x^\alpha} < 1 + \frac{1}{2^\alpha} + \frac{1}{3^\alpha} + \cdots + \frac{1}{n^\alpha}$$

$$(2) \int_1^n \frac{dx}{x^\alpha} > \frac{1}{2^\alpha} + \frac{1}{3^\alpha} + \cdots + \frac{1}{n^\alpha}$$

(問2)  $\alpha > 0$  とする。 次の無限積分が存在する条件を求めよ。

$$\int_1^\infty \frac{dx}{x^\alpha}$$

(問3)  $\alpha > 0$  とする。 次の級数の収束、発散を判定せよ。

$$\sum_{n=1}^{\infty} \frac{1}{n^\alpha}$$

(問4)  $a, b$  を  $0 < b \leq a$  を満たす定数とする。  $\alpha \geq 1$  の時、次の級数の収束、発散を判定せよ。

$$\sum_{n=1}^{\infty} \left( \frac{a}{n^\alpha} - \frac{b}{(n+1)^\alpha} \right)$$



## II 情報数学B

$p$  を奇素数とし、標数  $p$  の素体  $\mathbb{F}_p$  上のすべての  $2 \times 2$  行列のなす集合を  $M_2(\mathbb{F}_p)$  とかく。すなわち、

$$M_2(\mathbb{F}_p) = \left\{ A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mid a, b, c, d \in \mathbb{F}_p \right\}.$$

- (問1) 集合  $M_2(\mathbb{F}_p)$  から行列  $A$  を一様ランダムに選ぶとき、 $|A|^2 = 1$  である確率を求めよ。ただし、ここで  $|A|$  は行列  $A$  の行列式を表す。
- (問2) 集合  $M_2(\mathbb{F}_p)$  から行列  $A$  を一様ランダムに選ぶとき、 $A^2 = I$  である確率を求めよ。ただし、ここで  $I$  は単位行列  $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$  を表す。



## Ⅲ通信ネットワーク

以下のケース1からケース3は、IP ネットワークで障害が発生した3つのケースにおいて、それぞれネットワークワークコマンドを実行した場合の画面表示である。3つのケースの障害原因は互いに異なるものとして、画面表示から各ケースの障害原因を推定し解答用紙に記述せよ。

### (ケース1)画面表示1

```
C:¥>ping 10.1.2.3

Pinging 10.1.2.3 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.2.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)
```

### (ケース2)画面表示2

```
C:¥>ipconfig

Windows IP Configuration

Ethernet adapter ローカル エリア接続:

    Connection-specific DNS Suffix:
    Autoconfiguration IP Address: . . . :169.254.0.2
    Subnet Mask . . . . . :255.255.0.0
    Default Gateway . . . . . :
```

### (ケース3)画面表示3

```
C:¥>netstat -s

IPv4 Statistics

Packets Received           =562255
Received Header Errors     =0
Received Address Errors    =9
Datagrams Forwarded        =0
Unknown Protocols Received =0
Received Packets Discarded =247685
Received Packets Delivered =314561

(以下省略)
```



## IV情報システム

次の4つの問いについて、各々5行程度で答えよ

- (問 1) 最近のプロセッサは、処理性能向上のためにスレッド(thread)という単位での制御を有しているものが多い。その機構の概要を述べ、また、スレッドとプロセスとの差異について述べよ。
- (問 2) ページング(paging)とは何か、また、その目的について述べよ。
- (問 3) プロセッサの持つ動作モードに特権モード(privileged mode, または、kernel mode, supervisor mode)がある。特権モードとは何か、また、それを設ける目的について述べよ。
- (問 4) 複数のプロセッサで主記憶を共有する主記憶共有マルチプロセッサ(shared memory multi-processor)構成では、各プロセスはどのプロセッサ上でも実行可能であり、性能向上等に有用である。しかし、キャッシュに、ある工夫が必要である。その工夫とはどういうものか、また、それが必要となる理由について述べよ。





# Vソフトウェア

(問 1) 複数のプロセス間でリソース共有を行うため、Dijkstra は、下記の  $\text{down}(s)$ ,  $\text{up}(s)$ <sup>1</sup> という二つの操作を提案した。a から c の空欄に入るものを解答用紙に記せ。

$\text{down}(s)$ : s が  より大きいかわ調べ、大きければその値を  減少して実行を継続する。  
それ以外の場合は、プロセスはブロック状態に入る。

これらの処理は ひとつの切り離せない処理として実行する。

$\text{up}(s)$ : ブロックされているプロセスがあれば、その中から一つを選び、実行可能状態にする。

それ以外の場合は、s を  増加させ、実行を継続させる。

これらの処理は ひとつの切り離せない処理として実行する。

(問 2) s のことを何というか?

(問 3) プロセス間のリソース共有問題の有名な例として、食事をする哲学者の問題がある。5 人の哲学者が丸テーブルに座り、各哲学者の前にはスパゲッティの皿が置かれ、皿と皿の間には 1 本のフォークが置いてある。スパゲッティを食べるためには、哲学者は左右 1 本ずつのフォークをつかむ必要がある。哲学者は食べることで瞑想することで生活をしている。哲学者が空腹になると、左右から 1 本ずつ 2 本のフォークをつかんだ後、しばらくスパゲッティを食べ、その後フォークを置いて瞑想にふける。これを繰り返す。問題は決まった作業だけを行い、中断状態を引き起こさないような哲学者用のプログラムをかけるかということである。

下記の例は、この問題に対する不十分な解決例である。なお、 $\text{think}()$ ,  $\text{eat}()$ 関数は別に定義してあるとする。

この解はなぜ不十分であるか述べよ。

```
#define N 5                                /* 哲学者の数 */

void philosopher(int i)                    /* i: 哲学者の番号 0-4 */
{
    while (TRUE) {
        think();                          /* 瞑想中*/
        take_fork(i);                     /* 左のフォークを掴む */
        take_fork((i+1) % N);             /* 右のフォークを掴む */
        eat();                             /* 食べる */
        put_fork(i);                      /* 左のフォークをテーブルに戻す */
        put_fork((i+1) % N);              /* 右のフォークをテーブルに戻す */
    }
}
```

<sup>1</sup> Dijkstra の論文ではオランダ語で P, V と記されている。ここでは分かりやすくするために  $\text{down}$ ,  $\text{up}$  に変えている。

(問 4) 下記にこの問題に対する解決策の一つを記す。d から k の空欄に入るものを解答用紙に記せ。

```

#define N 5                /* 哲学者の数 */
#define LEFT (i-1)%N      /* i の左の数 */
#define RIGHT (i+1)%N    /* i の右の数 */
#define THINKING 0        /* 瞑想中 */
#define HUNGRY 1          /* フォークを掴もうとしている*/
#define EATING 2          /* 食事中*/

typedef int semaphore;
int state[N];              /* 各人の状態を表す*/
semaphore mutex=1;        /* 危険領域に対する排他制御 */
semaphore s[N];           /* 哲学者一人につき一つのセマフォ*/

void philosopher(int i)   /* i: 哲学者の番号 0-4 */
{ while (TRUE) {
    think();              /* 瞑想中*/
    [ d ];                /* 2本のフォークを掴む 又はブロック*/
    eat();                /* 食べる*/
    [ e ];                /* フォークをテーブルに戻す */
}

void take_forks(int i)
{ down([ f ]);           /* 危険領域に入る */
  state[i]= HUNGRY;
  test(i);               /* 2本のフォークを掴もうとする */
  up([ g ]);
  down([ h ]);           /* 2本のフォークを掴めなかった時、ブロックする*/
}

void put_forks(int i)
{ down([ i ]);
  state[i]= THINKING;
  test(LEFT);           /*左隣りが食事ができるか見る */
  test(RIGHT);          /*右隣りが食事ができるか見る */
  up([ j ]);
}

void test(int i)
{ if (state[i] == HUNGRY && state[LEFT] != EATING && state[RIGHT] != EATING) {
    state[i] = [ k ];
    up(&s[i] ); }
}

```

## 小論文

現在、政府が日本国内で生活する個人に番号を割り振り、税金と社会保障に関する個人情報を一元的に管理して利用しようとする制度の導入が検討されている。このような制度の導入に関する問題点に関して、自分自身のこれまでの経験にも即しながら、2000字以上3000字以内で小論文を作成せよ。

