

1. Które z wymienionych funkcji związane są z obsługą procesów?

- |                                 |                                |                                 |                                 |                                 |
|---------------------------------|--------------------------------|---------------------------------|---------------------------------|---------------------------------|
| <input type="checkbox"/> read   | <input type="checkbox"/> open  | <input type="checkbox"/> exit   | <input type="checkbox"/> lseek  | <input type="checkbox"/> mkfifo |
| <input type="checkbox"/> write  | <input type="checkbox"/> close | <input type="checkbox"/> fork   | <input type="checkbox"/> creat  | <input type="checkbox"/> pipe   |
| <input type="checkbox"/> signal | <input type="checkbox"/> kill  | <input type="checkbox"/> execlp | <input type="checkbox"/> dup2   | <input type="checkbox"/> wait   |
| <input type="checkbox"/> dup    | <input type="checkbox"/> sleep | <input type="checkbox"/> execvp | <input type="checkbox"/> unlink |                                 |

2. Plik tekst.txt ma następującą zawartość: abcdefghijkl. Co zostanie odczytane z pliku?

```
int d;
char buf[20];

d = open("tekst.txt", O_RDONLY);
lseek(d, 4, SEEK_SET);
read(d, buf, 3);
```

3. Ile różnych wartości i ile razy każda z nich pojawi się na standardowym wyjściu?

```
int main(){
    pid_t pid1, pid2, pid3, pid4; .....

    pid1 = fork(); .....
    pid2 = getpid(); .....
    pid3 = getppid(); .....
    printf("%d\n", pid1); .....
    printf("%d\n", pid2); .....
    printf("%d\n", pid3); .....
}
```

4. Które z podanych sekwencji z całą pewnością nie pojawią się na wyjściu w wyniku wykonania poniższego programu, gdy wszystkie wywołania systemowe, które dojdą do skutku, będą poprawne?

```
main(){
    int d;
    char buf[10];
    mkfifo("fifo", 0600);
    printf("A");
    if (fork() == 0) {
        printf("B");
        d = open ("fifo", O_RDONLY);
        printf("C");
    }
    else {
        printf("D");
        d = open ("fifo", O_WRONLY);
        printf("E");
    }
}
```

ABCDE  
 ABCED  
 ABDEC  
 ABDCE  
 ADEBC  
 ADBEC  
 ADBCE

5. Jaki będzie wynik wykonania poniższego programu (co zostanie wypisane na standardowym wyjściu):

```
int main() {
    int pid;
    pid = fork();
    if ( pid == 0 ){
        sleep( 10 );
        printf("Koniec potomka\n");
    }
    else {
        sleep( 5 );
        kill( pid, 9 );
        printf("Koniec przodka\n");
    }
}
```

6. Proszę zaznaczyć (poprzez obrysowanie dookoła) sekwencje liter, które mogą pojawić się na wyjściu w wyniku wykonania poniższego programu, oraz przekreślić dwoma krzyżującymi się liniami te sekwencje, które z całą pewnością nie pojawią się na wyjściu.

```

struct BUF {
    long mtype;
    int mvalue;
};
#define msize sizeof(struct BUF) - sizeof(long)

main() {
    int id;
    struct BUF mb;

    id = msgget(IPC_PRIVATE, 0600|IPC_CREAT);
    if ( fork() == 0 ){
        printf("A");
        mb.mtype = 1; mb.mvalue = 10;
        msgsnd(id, &mb, msize, 0);
        printf("B");
        mb.mtype = 2; mb.mvalue = 20;
        msgsnd(id, &mb, msize, 0);
        printf("C");
        msgrcv(id, &mb, msize, 4, 0);
        printf("D");
        msgrcv(id, &mb, msize, -3, 0);
        printf("E");
    }
}
else{
    printf("F");
    msgrcv(id, &mb, msize, -2, 0);
    printf("G");
    mb.mtype = 3; mb.mvalue = 30;
    msgsnd(id, &mb, msize, 0);
    printf("H");
    mb.mtype = 4; mb.mvalue = 40;
    msgsnd(id, &mb, msize, 0);
    printf("I");
}
}

```

FABGHICDE      FABCGHIDE      FABCDGHIE      FABCDEGHI      FAGHIBUDE  
 ABFGHICDE      ABCFGHIDE      ABCDFGHIE      ABCDEFGHI      AFGHIBUDE

7. Które z podanych wyników mogą się pojawić na standardowym wyjściu w wyniku wykonania poniższego programu:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
x1=3	x1=3	x1=3	x1=3	x1=3	x1=3	x3=4	x3=4
x2=3	x2=3	x3=4	x3=5	x3=4	x3=5	x1=3	x1=5
x3=4	x3=5	x2=3	x2=5	x4=7	x4=8	x2=3	x2=5
x4=7	x4=8	x4=7	x4=8	x2=3	x2=8	x4=7	x4=8

```

main(){
    int semid, x = 2;
    struct sembuf operacja = {0, 0, 0};
    semid = semget(IPC_PRIVATE, 1, 0600);
    semctl(semid, 0, SETVAL, (int)0);
    if (fork() > 0){
        x++;
        printf("x1=%d\n", x);
        operacja.sem_op = +5;
        semop(semid, &operacja, 1);
        wait(NULL);
        printf("x2=%d\n", x);
    }
    else{
        operacja.sem_op = -2;
        semop(semid, &operacja, 1);
        x+=2;
        printf("x3=%d\n", x);
        semop(semid, &operacja, 1);
        x+=3;
        printf("x4=%d\n", x);
    }
}

```