1. Fie următoarele procese și timpii lor de execuție și de intrare (process, timp execuție, timp intrare): (P1, 8, 1), (P2, 4, 2), (P3, 3, 3), (P4, 7, 4). Determinați timpii medii de așteptare pentru First Come First Served (FCFS) și Shortest Job First (SJF).

2. Câte procese se vor crea în urma execuției secvenței de mai jos (se exclude procesul curent)? Toate apelurile fork se întorc cu succes.

```
fork();
if (fork() == 0)
            fork();
```

3. Fie secvențele de pseudocod de mai jos. Care din cele două abordări este optimă?

`mutex_lock(&mutex);`
`a++;`
`mutex_unlock(&mutex);`
	`spin_lock(&spinlock);`
	`a++;`
	`spin_unlock(&spinlock);`

4. În secvența de cod de mai jos, PAGE_SIZE reprezintă dimensiunea unei pagini, iar apelul mmap reușește. În urma rulării a 10 instanțe ale programului de mai jos se ocupă 11 pagini fizice în zonele de date (date inițializate, heap, readonly (.rodata), bss). Argumentați acest comportament.

```
const char x[PAGE_SIZE] = {1, };

int main(void)
{
        char *z = mmap(NULL,  PAGE_SIZE, PROT_READ, MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);
        printf("%c %c\n", x[0], z[0]);
        while (1)  /* wait forever */
                ;
        return 0;
}
```

5. Care dintre apelurile sigaction și sigemptyset va dura mai mult? sigaction actualizează rutina de tratare a unui semnal, iar sigemptyset initializeaza setul de semnale descris de structura sigset_t la 0.

6. Fie un sistem pe 32 biți cu 100MB RAM, 100MB de swap și pagini de 4KB. Spațiul de adresă virtual este împărțit 3GB programe utilizator / 1 GB kernel. Un program aflat în execuție rulează:

```
        void *ptr = malloc(1024 * 1024 * 1024);
```

Apelul se realizează cu success, returnând un pointer valid. Motivați de ce malloc se întoarce cu succes.
Ce se întâmplă și de ce, dacă se rulează:

```
        memset(ptr, 0, 1024 * 1024 * 1024);
```

1. Fie următoarele procese și timpii lor de execuție și de intrare (process, timp execuție, timp intrare): (P1, 8, 1), (P2, 4, 2), (P3, 3, 3), (P4, 7, 4). Determinați timpii medii de așteptare pentru First Come First Served (FCFS) și Shortest Job First (SJF).

2. Câte procese se vor crea în urma execuției secvenței de mai jos (se exclude procesul curent)? Toate apelurile fork se întorc cu succes.

```
fork();
if (fork() == 0)
            fork();
```

3. Fie secvențele de pseudocod de mai jos. Care din cele două abordări este optimă?

`mutex_lock(&mutex);`
`a++;`
`mutex_unlock(&mutex);`
	`spin_lock(&spinlock);`
	`a++;`
	`spin_unlock(&spinlock);`

4. În secvența de cod de mai jos, PAGE_SIZE reprezintă dimensiunea unei pagini, iar apelul mmap reușește. În urma rulării a 10 instanțe ale programului de mai jos se ocupă 11 pagini fizice în zonele de date (date inițializate, heap, readonly (.rodata), bss). Argumentați acest comportament.

```
const char x[PAGE_SIZE] = {1, };

int main(void)
{
        char *z = mmap(NULL,  PAGE_SIZE, PROT_READ, MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);
        printf("%c %c\n", x[0], z[0]);
        while (1)  /* wait forever */
                ;
        return 0;
}
```

5. Care dintre apelurile sigaction și sigemptyset va dura mai mult? sigaction actualizează rutina de tratare a unui semnal, iar sigemptyset initializeaza setul de semnale descris de structura sigset_t la 0.

6. Fie un sistem pe 32 biți cu 100MB RAM, 100MB de swap și pagini de 4KB. Spațiul de adresă virtual este împărțit 3GB programe utilizator / 1 GB kernel. Un program aflat în execuție rulează:

```
        void *ptr = malloc(1024 * 1024 * 1024);
```

Apelul se realizează cu success, returnând un pointer valid. Motivați de ce malloc se întoarce cu succes.
Ce se întâmplă și de ce, dacă se rulează:

```
        memset(ptr, 0, 1024 * 1024 * 1024);
```

În conformitate cu ghidul de etică al Catedrei de Calculatoare, declar că nu am copiat la această lucrare. De asemenea, nu am ajutat și nu voi ajuta pe nimeni să copieze la această lucrare.

Nume:..... Grupă: .....

Semnătură: .....

În conformitate cu ghidul de etică al Catedrei de Calculatoare, declar că nu am copiat la această lucrare. De asemenea, nu am ajutat și nu voi ajuta pe nimeni să copieze la această lucrare.

Nume:..... Grupă: .....

Semnătură: .....