

21

מדינת ישראל

[illegible][illegible][illegible]

```
struct queue_head *queue_get(struct queue, root *root) {
```

```
struct node head = head, *next;
```

lock (& root \rightarrow head-lock);

```
while (root->head != NULL) {
```

~~pthread~~ - wait (empty);

3

head = root \rightarrow head

$\text{root} \rightarrow \text{head} = \text{head} \rightarrow \text{next}$

```
unlock(&root → head_block);
```

```
return head;
```

void queue_ptr (struct queue_head *hex, struct queue_root *root)

```
lock (&root → +all -lock);
```

if (root == tail) == Null

3

root \rightarrow Lead: hely

$$\text{new_next} = \text{null}$$

3

return tail->next = new;

```

    rest->tail = new; }

```

```
pthread_cond_signal(&empty);
```

```
unlock (& root->tail_block);
```

3

אלה 2:

1. תהליך של תרגום מ-RNA ל-protein. זהו תהליך שבו ה-RNA משמש כתבנית ליצירת ה-protein. תהליך זה מתרחש בציטופלזמה או בגוף הריבוזומי.
2. תהליך של תרגום מ-DNA ל-RNA. זהו תהליך שבו ה-DNA משמש כתבנית ליצירת ה-RNA. תהליך זה מתרחש בגרעין.
3. תהליך של תרגום מ-RNA ל-RNA. זהו תהליך שבו ה-RNA משמש כתבנית ליצירת ה-RNA. תהליך זה מתרחש בגרעין.

3: alic

```
struct my {
    mutex m
}
struct c {
    cond c; none;
}
```

```
win_lock(-)
{
    unix_lock(m)
}
win_unlock(-) {
    unix_unlock(m)
}
```

```
win_signal(c) {
    unix_lock(m);
    flag = 1;
    while (none > 0)
        { none--; }
    unix_signal(c);
    unix_unlock(m);
}
```

```
win_wait(c) {
    unix_lock(m);
    if (flag == 0)
        { none++; }
    unix_wait(c);
}
unix_unlock(m);
```

```
win_reset(c) {
    flag = 0;
}
```

1: alic

אם המערכת היא מונו-ליטית, כלומר, כל התוכנות והמערכת עצמה רצים על אותו ליבה, אז אין צורך במונitors. אבל אם המערכת היא מולטי-ליטית, כלומר, יש לה מספר ליבות, אז צריך מונitors כדי לשמור על קונסיסטנטיות. המונitors הם מבנים שבהם כל משתנה משותף נמצא בתחתיתם, ויש להם פונקציות של lock וunlock. כל פעם שאנחנו רוצים לשנות משתנה, אנחנו לוקים את המונитор, עושים את השינוי, ופונקציה של unlock. זה מבטיח שרק תוכנית אחת תוכל לשנות את המשתנה בכל פעם.