תוכן תרגול

مبואז לסינכרוניזציה ב-POSIX
 Orcoro בTextWriter

mutex □
Semaphore □

condition variable □

דוגמה: מיומש מנעולים קודאים-כותבים.
מבוא לסינכרוניזציה ב-לינוקס

- הצור ב-'סינכרון הקשורי' ייצוג של מונעหาย (enburg)
- הון חゾי בכול מערכות הפועלת
- לינוקס מספקת מגוון רחב של מונגים לסינכרון
- הון חゾי בכול מערכות הפועלת
- לינוקס מציעה עצום בسفرירטים ובגרעינים
- בחרוגו זה נצאי את עיקר מונגי הסינכרון
- горף במחמת המונגוס (API) הון חゾי
 putStrLn POSIX ב- \textit{Linux Threads}

- \textbf{ POSIX-1003.1b conditioned variables}
- \textbf{ POSIX-1003.1b monitors}

...אירח עיבור חוטני (לא בינ תחילתיים)

- \textbf{ mutexes}
- \textbf{ condition variables}
- \textbf{ monitors}

...איבת ניינ למשמע מעるために כללי, כי שרגאשה בהמשר
מנעול,UTH: mutex

מנעול אסプロкров לתוח אחד בדים לקהימי ב
לעוז (לעוז)

כל תוח אוחר המبصر לקהימי במגעך יחסוך עד א chịu
המעשלишוחר

洛克 התוח המזוחיק ימגעך אמור לשחזר אתו (核实על
המעשל)

מנעול,UTH: mutex המשמש בדרך-כלל להגנה על גישה
לנתונים משותפים, בחר קרוע קוד קריטי, לע,核实ת
המעשל בכניסהלקטש הקיריטי ושחזרו בسوف.
include <pthread.h>

int pthread_mutex_init(pthread_mutex_t *mutex, const pthread_mutex_attr_t *mutexattr);

int pthread_mutex_destroy(pthread_mutex_t *mutex);

POSIX

Linux

mutex (1)
פעולים על מעתולים (2)

- **mutex**
  ```c
  int pthread_mutex_lock(pthread_mutex_t *mutex);
  ```

- **נסיגת מעתול**
  ```c
  int pthread_mutex_trylock(pthread_mutex_t *mutex);
  ```

- **שחרר מעתול**
  ```c
  int pthread_mutex_unlock(pthread_mutex_t *mutex);
  ```
mutex על מקטולי (3)

فرقטרים:

- המקטולי על מבוטע המקולע

- למזרח את התוכנות h-mutex
- למזרח את התוכנות h-mutexattr

- בירית המתחד
- במקטול h-mutex - NULL
- הממחישים לגזרה h-mutex במקטול "בודק שגיאות", כדי למלוןizado
- הממחישים לגזרה h-mutex במקטול "בודק שגיאות", כדי למלוןizado

- man pages
- הפריטיםقيل לבוחר את סוג h-mutex

ערך מותזר: 0 בטצלחו, ערך אחר בכסף
## סוכרים ב-

### POSIX

**mutex סוכרים של מניעלי**

<table>
<thead>
<tr>
<th>סוכר המגע &quot;&quot;</th>
<th>סוכר המגע &quot;&quot;</th>
<th>נעלת חוורת &quot;&quot;</th>
<th>pozix녹ורסי</th>
<th>pozix녹ורסי</th>
</tr>
</thead>
<tbody>
<tr>
<td>מובטח</td>
<td>מובטח</td>
<td>DEADLOCK</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
</tr>
<tr>
<td>מובטח מניפה נעל &quot;&quot;</td>
<td>雕塑יר 녹르סי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
</tr>
<tr>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורسي</td>
</tr>
<tr>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
</tr>
<tr>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
</tr>
<tr>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
<td>pozix녹ורסי</td>
</tr>
</tbody>
</table>
mutex

Let's define a mutex system:

```c
pthread_mutex_t m;
int count;

void update_count() {
    pthread_mutex_lock(&m);
    count = count * 5 + 2;
    pthread_mutex_unlock(&m);
}

int get_count() {
    int c;
    pthread_mutex_lock(&m);
    c = count;
    pthread_mutex_unlock(&m);
    return c;
}
```

1. When defining and initializing thealyse_to_count() method:
   - Define a mutex for synchronization.

2. When accessing thealyse_to_count() method:
   - Lock and unlock the mutex for thread safety.

Comment: This is a basic example of using mutexes in C. It shows how to protect shared variables.

POSIX/Linux
סמספורים כלליים

סמספור הממוסע ב-POSIX: Linux Threads

- לכל סמספור יש מנוה, תור ומונע שט🏼פקייה לולא על הקוד של הפעולה של הסמספור ( 있었다: השם של הסמספור)
- השתייה פועל ומופיעה על הסמספור:
  - wait: אם המונה גדול מ-0, מקטינה אתו ב-1, אוחזר החוזה נכס להמתנה בתור.
  - בחראזא הקטנה גמ אם המונה קטן מ-0.
- post: אם תור הממחייה לא ריק, מוציא אתו ומטרח את החוזה הרקשון בתור.
  - אוחזר המגדל את המונה ב-1.
- בחראזא רקון ל-postッシュ
- signal

תור הממחייה הון (FIFO) לפי עדיפות, לכלור חות עדיפות
- גובחה⚽︎ איצ מחתו לפי חות עדיפות נמוכת

سمعCharset בוטי POSIX

"שכורה בוטי POSIX" סמספורים כלליים

סמספור הממוסע ב-POSIX: Linux Threads

- לכל סמספור יש מנוה, תור ומונע שט🏼פקייה לולא על הקוד של הפעולה של הסמספור ( есть השם של הסמספור)
- השתייה פועל ומופיעה על הסמספור:
  - wait: אם המונה גדול מ-0, מקטינה אתו ב-1, אוחזר החוזה נכס להמתנה בתור.
  - בחראזא הקטנה גמ אם המונה קטן מ-0.
- post: אם תור הממחייה לא ריק, מוציא אתו ומטרח את החוזה הרקשון בתור.
  - אוחזר המגדל את המונה ב-1.
- בחראזא רקון ל-postッシュ
- signal

תור הממחייה הון (FIFO) לפי עדיפות, לכלור חות עדיפות
- גובחה⚽︎ איצ מחתו לפי חות עדיפות נמוכת

سمعCharset בוטי POSIX

"שכורה בוטי POSIX" סמספורים כלליים

סמספור הממוסע ב-POSIX: Linux Threads

- לכל סמספור יש מנוה, תור ומונע שט🏼פקייה לולא על הקוד של הפעולה של הסמספור ( есть השם של הסמספור)
- השתייה פועל ומופיעה על הסמספור:
  - wait: אם המונה גדול מ-0, מקטינה אתו ב-1, אוחזר החוזה נכס להמתנה בתור.
  - בחראזא הקטנה גמ אם המונה קטן מ-0.
- post: אם תור הממחייה לא ריק, מוציא אתו ומטרח את החוזה הרקשון בתור.
  - אוחזר המגדל את המונה ב-1.
- בחראזא רקון ל-postッシュ
- signal

תור הממחייה הון (FIFO) לפי עדיפות, לכלור חות עדיפות
- גובחה⚽︎ איצ מחתו לפי חות עדיפות נמוכת

سمعCharset בוטי POSIX

"שכורה בוטי POSIX" סמספורים כלליים

סמספור הממוסע ב-POSIX: Linux Threads

- لكل سمسفور יש מנוה, תור ומונע שטמסות לולא על הקוד של הפעולה של הסמספור (есть השם של הסמספור)
- השתייה פועלת ב_smsفسיט על הסמספור:
  - wait: אם המונה גדול מ-0, מקטינה אתו ב-1, אוחזר החוזה נכס questão להמתנה
לдумать בתור.
  - בחרזא הקטנה גומ אם המונה קטן מ-0.
- post: אם תור הממחייה לא ריק, מוציא אתו ומטרח את החוזה הרקשון בתור.
  - אוחזר המגדיל את המונה ב-1.
- בחרזא רקון ל-postッシュ
- signal

תור הממחייה הון (FIFO) לפי עדיפות, לכלור חות עדיפות
- גובחה⚽︎ איצ מחתו לפי חות עדיפות נמוכת

سمعCharset בוטי POSIX

"שכורה בוטי POSIX" סמספורים כלליים

סמספור הממוסע ב-POSIX: Linux Threads

- לכל סמספור יש מנוה, תור ומונע שטמסות לולא על הקוד של הפעולה של הסמספור (есть השם של הסמספור)
- השתייה פועלת ב_smsفسיט על הסמספור:
  - wait: אם המונה גדול מ-0, מקטינה אתו ב-1, אוחזר החוזה נכס questão להמתנה
לдумать בתור.
  - בחרזא הקטנה גom אם המונה קטן מ-0.
- post: אם תור הממחייה לא ריק, מוציא אתו ומטרח את החוזה הרקשון בתור.
  - אוחזר המגדיל את המונה ב-1.
- בחרזא רקון ל-postッシュ
- signal

תור הממחייה הון (FIFO) לפי עדיפות, לכלור חות עדיפות
- גובחה⚽︎ איצ מחתו לפי חות עדיפות נמוכל
#include <semaphore.h>

size_t sem_init(sem_t *sem, int pshared, unsigned int value);

int sem_destroy(sem_t *sem);

POSIX

Linux
面白וה על סמספים (2)

ビوزוט על סמספי נ:

ירוזוט `wait` על סמספי:

int sem_wait(sem_t *sem);

ירוזוט `post` על סמספי:

int sem_post(sem_t *sem);

גרסא לא-חוסמת של `wait`:

int sem_trywait(sem_t *sem);

אם המונה של הסמספי לא יותר מגודל מ-0, הוזר newPosition

קריאת ערך מונה הסמספי:

int sem_getvalue(sem_t *sem, int *sval);
פונקציות על סמפורות (3)

פרמטרים:

- **sem** – הסמפורה עצמה המבוצעת בפעולה.
- **pshared** – אםعرן גודל מ-0, מציין שהסמפורה יולד ליחה.
- **value** –عرן החתולה שלมงคล הסמפורה.
- **sval** –เลข בלוקים בו יאוחסן ערךมงคล הסמפורה.

**ערך מוצמד: 0 베테לחה, (1-) בכישוף**

الفונקציות sind: **sem_getvalue()** ו- **sem_wait()**.
העורות על סמפורים

- סמפור המאוחזר על מונה בכרך 1 נkerja סמפור בינאר.
- סמפור בינאר יוכל לשמש ולהנהג על קטע קריטי (ליצור
  המיעום הדדי בין החוטים הניגשים).
- סמפור בינארי אינו מנעול mutual, משום שלך חוט יוכל
 لبצע post על סמפור, גמ לא比亚 ייצע על הסמפור
  קוד לשלו (יאן "בלולט" על הסמפור).

POSIX

ב- Linux
משתנה מסוים

נתבון ב明天 מימ貸. נרצהلسכנין את פעולות ההכonso
ורוצאה המחזור.andr 대한 ידכ פלואיז איבר מותר רק אם החר
אנון רך.

ב значение דו יש צוור לסכנין בוי ותית המבצועים ההכנו
אתם כדי לשמור על תקינות מבנה הגתנו, וכככוי לאיור
כדי להבנה תצאות איבר מותר רק.

משתנה תנאי (condition variable) — מפעיל לאות לחתיני
לא지원 כלשהו, כפי שמתואר "י"המחכים.

משתנה תנאי تمديد קשור ל-mutex מעומ.
mutex + 0 או יצרת משטני תנאי ≈ מוניור.

צורה זו חותי POSIX Linux
פעモノ על-unstyledית שהאד (1)

```c
#include <pthread.h>

int pthread_cond_init(pthread_cond_t *cond,
                      pthread_condattr_t *
                      cond_attr);
```

حفاظה העшла אים ויהי יוטו-Smithים של הותרת

```c
int pthread_cond_destroy(pthread_cond_t *
cond);
```

int pthread_cond_wait(pthread_cond_t *cond,
        pthread_mutex_t *mutex);
الفუונט על משתני ת네י (3)

вшחרור חוט ממתני על משתנה תנאי:

```c
int pthread_cond_signal(pthread_cond_t *cond);
```

אם אין חוט שמתתני الكرע על משתנה התנאי Ethics – הפונקציה לסרת החפעה (ואינה נזרקת הלאה).

אם כל אחת מתמטיים מתומן על משתנה התנאי Ethics – אוחד החוטים

המתכינים (ങรงות לא מורכבות) המסיקים להמתין על משתנה התנאי. החותה

ה""ל עובר להמתין על ה- mutex ויחזור לפוניקלה (ייבחר את ה- Mesa mutex (.isAuthenticated מצהש את ה- (pthread_cond_wait()
פונקציות של התחינת הממחינים על המשטחים התנאי

שחורות כל התחיית הממחינים על המשטחים התנאי:

```
int pthread_cond_broadcast(pthread_cond_t *cond);
```

ה intéressant הוא (אוyna יכתוב הלאת). אם לא יש לי חותם הממחין על המשטח התנאי החסס – הפעולה

אם קיים חותם ממחין על המשטח התנאי החסס – כל החותים

המחינים המפרשים למחינים על המשטח התנאי והפרים להמחין

לע ה-utex. התותים יזרז לפעולה בדה אחר זה (בเธอ כלשהו,

לא דווקא חום) לאחר סיעת מחדה את ה-utex.
פונקציות על משטחי חנה (5)

**פרמטרים:**
- `cond` - משטח התנאי עלוי מבצעת הפונקציה
- `cond_attr` - מgeries את תכונות משטח התנאי
- `mutex` - המנעול עליו משייך משטח התנאי הביצוע

**ערך מחודר:** 0 בעצלת, ערכ אתר בכישול

**הfonקציות:** `wait`, `broadcast`, `signal`, `init`

הצלו.
כโปּר בִּיןֵי אִירוּ וּמַשְׁתַּחֵתָה תָּנֵא (1)

הדרר הפשוטה ביוור לְקַשַּׁר בִּיןֵי אִירוּ הַמֶּֽזָּחִין קְיוּםָ מְצַב

רצוי להשלו למשנתה תנא'义乌 בַּכָּדוּ בָּדְלֵקַמּוּ

أدָלְמַתָּמוּ לְאִירוּ (הָחוּט שָׁרוֹרָה לְהָתָּכֵּדֵמְרָךְ קְאַשור יְושָׁב

הַמְצַבְּ הָרָצוּי)ः

if <לְאַ מַחְקִים הַמְצַבְּ הָרָצוּי

pthread_cond_wait (<הָחוּט שָׁרוֹרָה לְהָתָּכֵּדֶמְרָךְ קְאַשור יְושָׁב

הָמְשֶׁר הָקְדוֹד

أدָלְמַתָּמוּ לְאִירוּ (הָחוּט שָׁרוֹרָה אֵת הַמְצַבְּ הָרָצוּי וּמְסַמְּמָי לְחוּטִי

הַמְצַבְּ הָרָצוּי)ः

if אֶפֶֽשֶר לֹוְּרָיָֽהּּ אָכְלִי דֵּרוּיוֹ <מַחְקִים הַמְצַבְּ הָרָצוּי

pthread_cond_signal (<הָחוּט שָׁרוֹרָה לְהָתָּכֵּדֶמְרָךְ קְאַשור יְושָׁב

);
pthread_mutex_t qlock;
pthread_cond_t notEmpty;
/* ... initialization code ... */

void enqueue(item x) {
    pthread_mutex_lock(&qlock);
    /* ... add x to queue ... */
    pthread_cond_signal(&notEmpty);
    pthread_mutex_unlock(&qlock);
}

item dequeue() {
    pthread_mutex_lock(&qlock);
    if <queue is empty>
        pthread_cond_wait(&notEmpty, &qlock);
    /* ... remove item from queue ... */
    pthread_mutex_unlock(&qlock);
    /* .. return removed item */
}
קשור בינ' איורע ומתחנה תנא' (3)

פעולת signal לא מאפרשת לחוט הממחלק לحماית מיז.

החות הממחלק זריך התיחלה לפניyat ה-<mutex י mexico

(Mesa סונגו)

יexion שלפניך שהחות הממחלק י ulaş לפניyat ה-<mutex י אופֶד

יסף להכנס לחוט נסף שישנה את הנתונים 크' שיימצֶב

הרצויくる לא מתכֵּים.
כשתッチ ביני איור והמשתנה תניי (4)

ג在網, لنפי הזיזה של התו ריק ודוח hỏוט t1 קורא (-)

המשתנה enqueue ()

cעת הוט t2 קורא (-)

המשתנה signal, מכניס אישר לתоор ומ yazılı t1

cעת מграни הוט t3, קורא (-)

המשתנה dequeue ()

cעת t3 נסם מהמשתנה למגעול בתחליל הקוד

המשתנה enqueue ()

cעת t2 מתחבר לאח המגעול ומסים את (-)

המשתנה dequeue ()

cעת t3 מכבה את המגעול, רכスポ, מוציא אישר ומסים

המשתנה enqueue ()

cעת t1 מכבה את המגעול, ממני לבצע את הקוד ומנסה להזיז

אייב מתור ריק!
קישור בינ'אירוע ומפגטה תניי (5)

مامה נבעת הבעיה بدומה?

.mcק שמתאacios על האירוע לא בדך אחרי InterruptedException-

cejץ ניטן לפתור את הביעה?

"ע"بذיקה חזרת של תניי האירוע והמפגטה располגת לפנים הצוור.

لدודהמה:

while <queue is empty>
    pthread_cond_wait (…)

האם הבעיה הכלל לكرות עם אם המנעל היא חוג (טור?

(FIFO

鹳! אם 3 ת'היי קרוא ל-().dequeueومתוכה למגע לפנים/page-2

הגייל ל--

signal
ימ😠ויטו 묿ונב גנוזח

־ לופי סגנון Hoare (בדירש הוזה), החותם שמהוי. müzik

־ Signal גוס מוזהר על המ '_', ומעביר אתו לרות הממחור. ■

— בסופו זה מצב ההתחוות לא משותפה, והחותם הממחור מ,strongש לארה

הפרשה במצבי הרשים.

מצד שני, בסגנון Hoare החותם שיביציל Signal לע

כי שסיים לחות אחר לתיאוד

—which על התיאודמות שלו הממחור עד שהמחוזר משוחזר.

האםbnbויא שזאינונ בדונהות של התיאור יוכל ליקוט אם

?Hoare ה- he בסטון monitor

לא! כי 11 היא מקבל את המנועול לפעני 3

מעורוט הפעולה - תרגול 7 2003 (8) ארצ'י 2003
מ늘ול קוראים-כותבים

🎮ผสมרכ עב שטי פעלת: קיריאו וכרבה

☐ מסר חוטים קוראים יכלים לגשת למשאות בו-צמיגת

☐ חוט כותב צירっと ישנה בלעדית למשאות

☐ בשקפים הבאים נדחים צד ניתה לאמות מנסול

☐ קוראים-כותבים אמאפעת הסתתית והארה ראיות אין לשלמת בسفرה למטרה זו.
#include <pthread.h>

int number_of_readers;  
	pthread_cond_t readers_condition;

int number_of_writers;  
	pthread_cond_t writers_condition;

mutex_t global_lock;

void readers_writers_init() {
    number_of_readers = 0;
    pthread_cond_init(&readers_condition, NULL);
    number_of_writers = 0;
    pthread_cond_init(&writers_condition, NULL);
    pthread_mutex_init(&global_lock, NULL);
}
void reader_lock() {
    pthread_mutex_lock(&global_lock);
    while (number_of_writers > 0) 
        wait
    pthread_cond_wait(&readers_condition, &global_lock);
    number_of_readers++;
    pthread_mutex_unlock(&global_lock);
}

void readers_unlock() {
    pthread_mutex_lock(&global_lock);
    number_of_readers--;
    if (number_of_readers == 0)
        pthread_cond_signal(&writers_condition);
    pthread_mutex_unlock(&global_lock);
}
void write_lock() {
    pthread_mutex_lock(&global_lock);
    while ((number_of_writers > 0) || (number_of_readers > 0))
        pthread_cond_wait(&writers_condition, &global_lock);
    number_of_writers++;
    pthread_mutex_unlock(&global_lock);
}

void write_unlock() {
    pthread_mutex_lock(&global_lock);
    number_of_writers--;
    if (number_of_writers == 0) {
        pthread_cond_broadcast(&readers_condition);
        pthread_cond_signal(&writers_condition);
    }
    pthread_mutex_unlock(&global_lock);
}
הסרונות שלìmימש

כל עוד המנשא צל להקוראים, קורא חדש שמגי עזילך
להיכנס וישקף כתבים שלגינו לפניו.

הסיר הועונת והעבך הספרות שלכותבם.
לא יתייחד עם האמ הקוראים ואה נApiController הקטוע
הקוריא.

תלוי מידיציל_LTVO לטופוס ראשו את המנעול.

האם ואיך אפ prevState לפותר בעיות אלו?