ملاحضة

قد يكون فيه بعض الأخطاء

جميع الأسئلة كانت اكمل الفراغ والجواب النهائي تضعه في هذي الصفحة

Final Cs227 boys 21/11/2022

Selection chapters are only 4,5 and 8

1	2	3	4	5
Parallelism	Signals	Many-to-One	Many-to-One	1.6
6	7	8	9	10
all threads	One-to-One	Deferred	Extending Thread class	Implementing the Runnable interface

1	2	3	4	5	6
7	8	9	10	11	12
13					

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	

Fill in the blank 10 mark

1. ……..implies a system can perform more than one task **simultaneously**
2. .....are used in UNIX systems to notify a process that a particular event has occurred.
3. ..... Many user-level threads mapped to single kernel thread
4. ……One thread blocking causes all to block
5. application is 80% parallel and tow cores the is speed up is ……..
6. **exec()** usually works as normal – replace the running process including .....
7. ……It also allows multiple threads to run in parallel on multiprocessors
8. …….cancellation allows the target thread to periodically check if it should be cancelled
9. Java threads may be created by ..... or .....

Fill in the blank 13 mark

```
do {
    flag[i] = false;
    turn = j;
    while (flag[j] && turn = = j);
            critical section
    flag[i] = false;
            remainder section
} while (true);
```

1. From figure is satisfy the three requirements .....(yes/no)
2. If no what is the requirement are not satisfy .....(**Mutual Exclusion, Progress, Bounded Waiting**)

Consider P_1 ,$p_2$and P_3 that require S_1 to happen before S_2 ...

```
P1:
  S1;
  signal(synch1);
P2:
  wait(synch1);
  S2;
  signal(synch2);
P3:
 wait(synch2);
 S3;
```

3. semaphore "**synch**1" initialized …….. and "**synch**2"is ……..

4.

do {

acquire lock

critical section

release lock

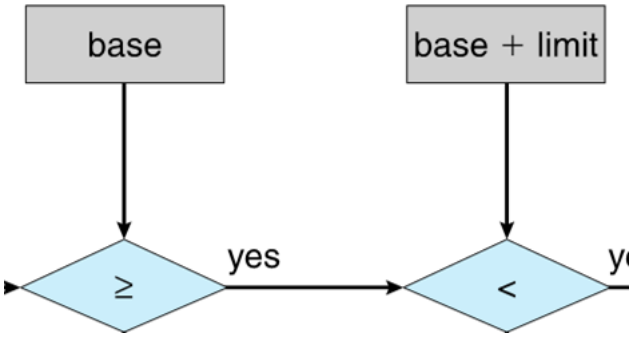
remainder section

} while (true);

5. **Priority Inversion** – Scheduling problem when lower-priority process holds a lock needed by higher-priority process
6. **wakeup** – remove one of processes in the waiting queue and place it in the ready queue

7. **Binary semaphore** – integer value can range only between 0 and 1
   Same as a **mutex lock**

Fill in the blank 17 mark



Base>= <base + limit

Logical and physical addresses are **the same in compile-time and load-time address-binding schemes;**

Calculating internal fragmentation

- Page size = 2,048 bytes
- Process size = 72,766 bytes
- 35 pages + 1,086 bytes
- Internal fragmentation of 2,048 - 1,086 = 962 bytes

Exercise 8)

Consider a logical address space of 256 pages with a 4-KB page size, mapped onto a physical memory of 64 frames.

a) How many bits are required in the logical address?

256 pages = 2^8 pages → we need 8 bits to represent the page number.
4KB = $2^2 \times 2^{10}$ bytes = 2^{12} bytes → we need 12 bits to represent the offset.
Hence, we need 8 + 12 = 20 bits for the logical address.

---

b) How many bits are required in the physical address?

64 frames = 26 frames → We need 6 bits to represent the frame number.
From (a), we concluded that we need 12 bits for the offset.
Hence, we need 6 + 12 = 18 bits for the physical address.

وبعض الأسئلة الحسابية