Concurrent Systems Paper 3
OS Foundations Paper 10

2000
JMB
Outline Solution. p10q8

a)(i)

```
carpark : monitor
    export enter(), exit()
    count : integer := ∅
    notfull : condition
  enter()
    if count ≥ N then wait (notfull)
    (raise entry barrier)
    count := count + 1
  exit()
    count := count - 1;
    signal (notfull);
    raise exit barrier
```

8

a) (ii)

```
spaces : semaphore := N
enter()
    wait (spaces)
    raise entry barrier
exit()
    signal (spaces)
    raise exit barrier
```

4

b)(i) semaphore implementation

there must be <u>atomic</u> operations

build on TAS, CAS
Reads clear etc.

```
wait (sem)
    if value > 0
      then decrement value
      else queue process
signal (sem)
    if any process queued, free one process
    else increment sem
```

sem: value
queue

4

(ii) need a monitor lock, typically a semaphore
need a queue for each condition variable.
on wait (condition) must cause monitor lock to be freed and process queued. On signal (condition) must let caller exit before signalled proceeds. 4