Minor I (6 Feb 06) CSL373:Operating Systems Time : 1100-1200

Answer the questions in the spaces provided on the question sheets. This is a **closed book and closed notes exam**. I donot answer any technical questions.

Name : Sample

If you feel that a reevaluation of a question is necessary } Please do the following: ① Write down "lucidly" why your case should be re looked at? ② Be prepared to answer the ensuing Q&A session.

Question:	1	2	3	4	Total
Points:	8	12	5	15	40
Score:					

1. (a) (2 marks) In class, we discussed copy-on-write for memory pages shared among multiple processes. We cannot apply this same concept blindly to process creation, but instead are forced to copy some parts immediately while other parts can be delayed. Knowing the components of general processes, which parts must be copied immediately, and which parts can be delayed and copied-on-write?

Stack space / Registers — ①

Entire Address Space — ①

(b) (2 marks) Can a multithreaded solution using multiple user-level threads achieve better performance on a multiprocessor system than on a single-processor system? Explain why.

In General NO ✓ ①

OS sees only one process and different threads cannot (will not) be scheduled on separate processors — ①

(LWP) → Answer can be partial Yes

Entry Number: ______________ Name : ______________________

(c) Consider a system using round-robin scheduling with a fixed quantum q. Every context switch takes s milliseconds. Any given process runs for an average of t milliseconds before it blocks or terminates.

i. (3 marks) Determine the fraction of CPU time that will be wasted because of context switches for each of the following cases

- t<q

$s/(t+s)$.

- t>> q (i.e. t is much greater than q)

t/q Context Switches will be Required.

$(t/q * s) / [t + (t/q * s)]$.

$= \frac{t}{qs} / [t + t/q \times s] =$

- q approaches 0

$q \to 0$

Efficiency $\to 0$

Waste $\to 100\%$.

ii. (1 marks) Under what conditions will the wasted fraction of CPU time be exactly 50

$q = s$ gives 50%.

Entry Number: ____________ Name : ____________

2. (12 marks) You have been hired by Save the Whale Organization to help the environment. Because unscrupulous commercial interests have dangerously lowered the whale population, whales are having synchronization problems in finding a mate. The trick is that in order to have children, three whales are needed, one male, one female, and one to play matchmaker literally, to push the other two whales together (I'm not making this up!). Your job is to write the three procedures Male(), Female() and Matchmaker(). Each whale is represented by a separate process. A male whale calls Male(), which waits until there is a waiting female and matchmaker; similarly, a female whale must wait until a male whale and a matchmaker are present. Once all three are present, all three return. Use semaphores to solve this problem.

```
Semaphore  male, female, ~~matchmaker~~ = 0
Semaphore  male-start = 0  male-end = 0,
Semaphore  female-start = 0, female-end = 0

male()                    female()                 matchmaker()
{                         {                        {
 ~~Signal~~ V(male);          V(female)                V(male)
 P(male-start);             P(female-st)             V(female)
 :                          ...                      P(male-start)
 P(male-end);               P(fem-end)               P(female-start)
}                         }                          match()
                                                     ~~Signal~~ P(m-e)
                                                   }  P(f-e.
```

(Certain soln)

wait (male)
wait (female)
signal (male-start)
signal (female-start).
match()
signal (male-end)
signal (female-end).
}.

Two semaphores are necessary.
One for mutual Exclusion betn male & female ①
One for mater and matchmaker ②

NOBODY has got this

Any whale can be a match maker as well as a ~~fem~~ mater
→ u have to protect against that also using ②.

Entry Number: ______________ Name : ______________________

3. (5 marks) We have discussed monitors in class. *Conditional Critical Region* is a similar language construct of the form " **region v when B do S**", where v is the name of the region, B is a Boolean expression and S is a statement. A process enters the critical region v only when B is true and no other processes are in region v.

Solve the bounded buffer problem using conditional critical regions.

B-f : shared record
pool : array [0..n-1] of item
count , in, out : integer
end;

region (buffer) when count < n.
do
begin.
pool [in] ← nextp.
in ← in+1
count++.

Same name

end.

region (buffer) when count >0
do begin.
nextc ← pool [out]
out ← (out+1) mod n
count --
end;

Entry Number: ______ Name : ______

4. (15 marks) Implement conditional critical regions using semaphores.

With each shared variable, mutex is associated

var x - mutex x-wait : semaphore
x- count x-temp : integer.

Mu Ex → Semaphore.

Boolean condn is false → waits on x-mutex
count keeps track of the processes waiting.

x-temp (to keep track of boolean condn being changed).

P(x- mutex).
if not B.
then begin
x - count --
V(x - mutex).
P(x-wait).

while not B
do
begin
x - temp ++.
if x - temp < x-count
then V (x-wait)
else V(x - mutex)
P(x-wait

end
x-count --
end
S;
if x -coun

ok

Text Book Solution

(check Stallings) Tanenbaum

ok