CSCI 322, Winter 2015, Midterm Exam
Solutions

1. The semaphore operations `increment` and `decrement` go by many names, depending on their function. Put each of the following words into the correct box:

P	V	signal	wait	post	lock	unlock

increment			
V	signal	post	unlock

decrement		
P	wait	lock

2. **FALSE** The *cache* is fast memory designed to speed up concurrent processing.
3. **TRUE** *False sharing* occurs when two variables share the same cache line.
4. **TRUE** If `B` is *At-Most-Once*, then `<await(B);>` can be implemented as `while (not B) skip;`
5. **FALSE** A *unconditionally fair* scheduler guarantees that, for an eligible statement `<await(B) S;>`, if `B` becomes true infinitely often, then `S` is guaranteed to execute.
6. As a function of `n`, how many different orders can the following program write out the numbers? Assume writing a number is atomic. $n!$

```
process foo[i = 1 to n] {
  write(i);
}
```

7. A *safety property* means: something bad does not happen.
8. A *liveness property* means: something good eventually happens.
9. *Partial correctness* means: the final state is correct.
10. *Total correctness* means: the final state is correct and the program terminates
11. The *read set* of a process is: the set of all variables read by a process.
12. The *write set* of a process is: the set of all variables written to by a process.
13. Two processes are *independent* means: the write set of each is disjoint from the read and write sets of the other.
14. Give the justifications for lines in the following proof. Remember to give the **line numbers that justify each line** whenever a line is inferred from previous lines. Assumptions for conditional or indirect proofs can just be marked A.

    Prove: $((P \vee Q) \Rightarrow R) \Rightarrow (P \Rightarrow R)$

1.	$(P \vee Q) \Rightarrow R$	Assumption
2.	$\neg(P \Rightarrow R)$	Assumption
3.	$P \wedge \neg R$	2,Equivalent by Axiom
4.	P	3,Simplification
5.	$\neg R$	3,Simplification
6.	$\neg(P \vee Q)$	1,5,Modus tollens
7.	$\neg P \wedge \neg Q$	6,Equivalent by Axiom
8.	$\neg P$	7,Simplification
9.	$P \Rightarrow R$	4,5,Indirect proof
10.	$((P \vee Q) \Rightarrow R) \Rightarrow (P \Rightarrow R)$	Conditional proof

15. What should replace $\boxed{A}$ and $\boxed{B}$ in the following proof?

    Prove: $\{x > 0\}$ `y = 2*x` $\{y > x\}$

1.	$\boxed{A}$	$\boxed{B}$
2.	$(x > 0) \Rightarrow (2x > x)$	Logic and arithmetic
3.	$\{x > 0\}$ `y` $= 2 *$ `x` $\{y > x\}$	1, 2, Consequence

    (a) $\boxed{A}$: $\{2x > x\}$ `y = 2 * x` $\{y > x\}$

    (b) $\boxed{B}$: Assignment Axiom

16. Here is part of a proof. Without knowing what the omitted lines or their justifications are, can you determine what should go in for $\boxed{A}$ and $\boxed{B}$?

1.	$\cdots$	
2.	$\boxed{A}$	
3.	$\cdots$	
4.	$\boxed{B}$	
5.	$\{x = 0\}$ `if(x >` y`)y =` `x` $\{y \geq 0\}$	2, 4, if-then

    Can be in either order:

    (a) $\boxed{A}$: $\{(x = 0) \wedge (x > y)\}$ `y = x` $\{y \geq 0\}$

    (b) $\boxed{B}$: $((x = 0) \wedge (\neg(x > y))) \Rightarrow (y \geq 0)$

17. We want to prove that the following precondition and assignment:

$$\boxed{\{x >= 4\}\langle \texttt{x = x - 4 } \rangle}$$

    does not interfere with the precondition of

$$\boxed{\{x >= 0\}\ \langle \texttt{x = x + 5 } \rangle\ \{x >= 5\}}$$

    (a) What do we have to prove?
    $\{(x \geq 0) \wedge (x \geq 4)\}\ \langle$`x = x - 4`$\rangle\ \{x \geq 0\}$

    (b) Does it interfere? **No**

18. The inference rule for the `if-then` statement looks like this:

$$\frac{\{P \wedge C\}\ \texttt{S}\ \{Q\}, P \wedge \neg C \Rightarrow Q}{\{P\}\ \texttt{if (C) then S}\ \{Q\}}$$

    Assume the `if-then-else` statement looks like:

```
if (C) then S1 else S2
```

    What would the `if-then-else` rule look like?

$$\frac{\{P \wedge C\}\ \texttt{S1}\ \{Q\}, \{P \wedge \neg C\}\ \texttt{S2}\ \{Q\}}{\{P\}\ \texttt{if (C) then S1 else S2}\ \{Q\}}$$

19.
```
co < await ( x > 5 ) x = x - 5; >   # line A
// < await ( x == 1 ) x = x + 8; >  # line B
oc
```

For the above program, for what initial values of `x` does this program terminate, assuming scheduling is fair? What are the corresponding final values of `x`? Explain your answer.

Case 1: If $x = 1$ then line B executes, making $x = 9$, then line A executes, making $x = 4$.

Case 2: If $x = 6$ then line A executes, making $x = 1$, then line B executes, making $x = 9$

Case 3: $x \leq 5$ but $x \neq 1$, then neither line executes and the program does not terminate.

Case 4: $x > 5$ but $x \neq 6$, then line A executes, but $x \neq 1$ so line B does not execute and the program does not terminate.

20. In the three processes below, SAB, SBA, SBC, and SCB are all semaphores initialized to zero.

process PA
```
while true:
  SAB.signal()
  SBA.wait()
  A
```

process PB
```
while true:
  SBA.signal()
  SBC.signal()
  SAB.wait()
  SCB.wait()
  B
```

process PC
```
while true:
  SCB.signal()
  SBC.wait()
  C
```

Rewrite the processes to use only three semaphores (initialized to anything you like), but obey the same synchronization as before.

Initialization
```
SAB = semaphore(0)
 SB = semaphore(0)
SCB = semaphore(0)
```

process PA
```
while true:
  SAB.signal()
  SB.wait()
  A
```

process PB
```
while true:
  SB.signal()
  SB.signal()
  SAB.wait()
  SCB.wait()
  B
```

process PC
```
while true:
  SCB.signal()
  SB.wait()
  C
```