Mark Randles
2006-03-23

**Homework #2**

1a. The advantages to synchronous communications are reliable delivery and it is memory safe. However, synchronous communications block the sending and receiving threads until the message has been sent. Asynchronous communication does away with this drawback but introduces it's own. Not all asynchronous communications are memory safe, since you could deallocate the memory holding the message before it's delivered. Another drawback is that the message could be left in the queue indefinably, the receiving process never asking for the message.
1b. Automatic buffering is the more user-friendly type of buffering, abstracting the buffering of data from the calling thread. However, explicit buffering has advantages, such as limiting the memory used by the buffer. However, explicit buffering requires more setup and initial overhead.
1c. The disadvantages of send-by-copy are data coherency and overhead. The data can be quickly become out of data because there exists multiple copies. Also, every time a value is passed you incur an overhead to allocate the new memory and copy over the data. However, send-by-reference also has a draw back in that there exists only one copy of the data, and any changes made it a function could effect the program in negative ways.
1d. A fixed-sized message is easier for the system to allocate and process, since it knows the absolute size of the message (ease in memory allocation) and less complicated algorithms to receive/send the message (processing). For the programmer, a fixed message size is limiting only if it has not been abstracted over. Also, depending on the application, it could increase the amount of memory overhead and processing time.

2. The value printed by line A is 20, because the child increments the global variable `value`.

3.

```
int main() {
      int lastid = 0;

      // create child 3
      lastid = fork();
      if(lastid == 0) {
            // create grandchild
            int grandchild = fork();
            if(grandchild == 0)
                  cout << “grandchild” << endl;
            else {
                  wait(grandchild);
                  cout << “child3” << endl;
            }
            exit();
      } else
            wait(lastid);
```

```
	// create child 2
	lastid = fork();
	if(lastid == 0) {
		cout << “child2” << endl;
		exit();
	} else
		wait(lastid);

	// create child 1
	lastid = fork();
		cout << “child1” << endl;
		exit();
	} else
		wait(lastid);

	// do the parent output
	cout << “parent” << endl;

	return(0);
}
```

4. Multithreaded processes share global variables and heap memory.

5a. The implication of alpha=0 is that Tau(n) will be the same every iteration, in this case 100ms.
5b. The implication of alpha=0.99 and Tau(n)=10 will be that Tau(n+1) = 10 for every iteration.

6a. SFJ is a specialized priority system, where the priority of a thread is the time left till completion.
6b.
6c. A FCFS is a specialized priority system, where the priority of a thread is the time that the job entered the queue.
6d.

7. **See attached sheet.**

8a. ((n - 1) * q) + ((n - 1) * s) = t
8b. n=100, t=1, s=0.001
((100 – 1) * q) + ((100 – 1) * 0.001) = 1
(100 – 1) * q = 1 – ((100 – 1) * 0.001)
q = (1 – ((100 – 1) * 0.001)) / (100 – 1)
q = (1 – (99 * 0.001)) / 99
q = (1 – 0.099) / 99
q = 0.901 / 99
q = 0.009101

n=100, t=1, s=0.01
((100 – 1) * q) + ((100 – 1) * 0.01) = 1
q = (1 – ((100 – 1) * 0.01)) / (100 – 1)
q = 0.000101 / 99
q = 0.00000102