Workflows with job dependencies

What is a workflow?

By "workflow" we are referring to a set of jobs, perhaps with differing resource requirements, where one job depends on the outcome of another. Some circumstances to consider are:

- One job processes the output of a previous job (a dependency)
- A different job should be started based on whether a previous job succeeded or failed
- A task must be performed at the end of a job, even if the job is cancelled
  (for example, copying data back from $PBS_MEMDISK)

You can instruct a job to start only after a dependency has been met with:

```
qsub -Wdepend=<type>:jobid> nextjob.q
```

Dependency types include:

- after (after the previous job has started)
- afterok (after the previous job has finished, only if exit code = 0)
- afternotok (after the previous job has finished, only if exit code is not 0)
A workflow is normally implemented as a script which submits a series of jobs, setting up dependencies between them.

But for the moment, we'll assemble our workflows at the command line.

Exercise
Using our wordcount example, make jobs for two samples and a job to compare them:

```bash
$ cat myjob1.q
#!/bin/bash
#PBS -l walltime=5:00
#PBS -l nodes=1:ppn=1
#PBS -l mem=1gb

RUNDIR=${SCRATCH}/tut3/
mkdir -p $RUNDIR
cd $RUNDIR
cp /share/apps/examples/arrayjob/* .
sleep 60
python ./wordcount.py sample-1.txt > wc1.out

$ cat myjob2.q
#!/bin/bash
#PBS -l walltime=5:00
#PBS -l nodes=1:ppn=1
#PBS -l mem=1gb

RUNDIR=${SCRATCH}/tut3/
mkdir -p $RUNDIR
cd $RUNDIR
cp /share/apps/examples/arrayjob/* .
sleep 60
python ./wordcount.py sample-2.txt > wc2.out

$ cat compare.q
#!/bin/bash
#PBS -l walltime=5:00
#PBS -l nodes=1:ppn=1
#PBS -l mem=1gb
#PBS -j oe

RUNDIR=${SCRATCH}/tut3/
cd $RUNDIR
grep -H "" wc1.out wc2.out | sort -rn -k2 > all_words.txt
```

We need to capture the job ID of each:

```bash
$ j1=$(qsub myjob1.q)
$ j2=$(qsub -Wdepend=afterok:$j1 myjob2.q)

$ qsub -Wdepend=afterok:$j2 compare.q
```

The "sleep" lines in each script are to help watch as they progress through the queue.

Use "qstat -u $USER" to watch as the jobs change from state "H" through "Q", "R" and then "C".
Array job dependencies

You can also set a dependency on some or all of the tasks in a job array:

```
$ qsub -Wdepend=afterokarray:1234567[] nextjob.q
$ qsub -Wdepend=afternotokarray:1234567[] nextjob.q
$ qsub -Wdepend=afteranyarray:1234567[] nextjob.q
$ qsub -Wdepend=afterokarray:1234567[][5] nextjob.q
```

The final form means "run nextjob.q after 5 tasks from array job 1234567 have completed with exit status 0"