

- Home

help2002

Schedule

Recordings (via LMS)

Project-2

Unit outline

C textbooks

OS textbooks

Information resources

Extra reading

Working effectively

Past projects

Recent student feedback

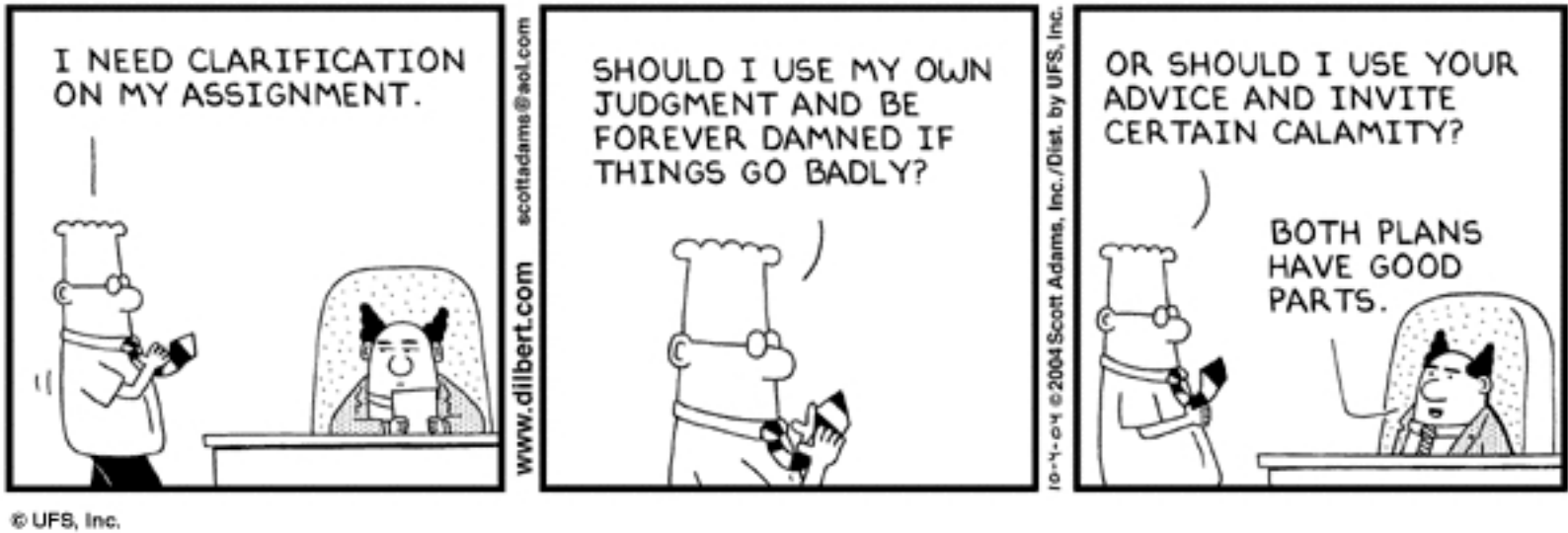
Department of Computer Science and Software Engineering

CITS2002 Systems Programming

Project 2 Clarifications

Back to [project 2](#).

This page provides any clarifications, more detailed explanations, big hints, or corrects any errors identified after the project description was released. This page will be updated with responses to any common problems or misunderstandings posted on [help2002](#) or asked in laboratory sessions.



12th October

Each **file** in your output tarfile should have the same modification time and permissions (modes) as the original file from which it was copied (coming from one of the input tarfiles).

You **do not** have to preserve the modification times or permissions (modes) of any **directories** in the output tarfile. Simply give directories the default permission of 0700 (an octal value), and just let their modification time remain at the time when the (new) directory was created. (it's possible to preserve these attributes for directories but, because it was not clearly stated in the project that this should be done, it will not be changed or defined at this late stage. If you already implemented this, do not change it, leave the code there).

2nd October

Some students are having difficulties creating input tarfiles to test their projects. The shellsript [create](#) emulates a simple (fictitious) workload over 3 days, and makes backup copies of each day's work using *tar*. Copy the shellsript to a new directory, and run it with *sh create*

```
prompt> sh create
```

You may then use the 3 input tarfiles to test your project. Note that, even when using the same shellsript, different implementations of *tar* (on different systems) will build identical tarfiles but may display (list) their results differently: [output-linux](#) and [output-macos](#). NOTE - these examples are listing the *input* tarfiles, not the final *output* tarfile.

29th September

Your output tarfile needs to retain the directory structure of its inputs - the sequence of directories (the pathname) containing each file is just as important in a backup as the file's name itself. The output tarfile should not just contain all files in a single 'top-level' directory, otherwise the clearly different files:

```
monday/java-project/Makefile
and
tuesday/python-project/Makefile
```

would incorrectly be 'collapsed' to a single *Makefile* [\[1\]](#), [\[2\]](#).

26th September

A helpful constant is MAXPATHLEN, which defines the maximum possible total length of any file or directory name. It typically has a value of 1024 (bytes). You can use the constant if your code has: *#include <sys/param.h>* [\[1\]](#).

23rd September

You **must not** use the *system()* function or the *popen()* function [\[1\]](#).

Why you may not use *system()* or *popen()* - [\[2\]](#).

22nd September

The entries in each input *tar* archive will only be regular files or directories (in particular, no hard or symbolic links).

Chris McDonald.