

Project supervisor email *

james.garside@manchester.ac.uk

Title and objective of the project *

COMP25111 practical development

Number of students requested (with justification if more than 1)

*

Preferably 2, if suitable candidates are available. There are likely to be a number of (somewhat) independent practical exercises based around an automated system. (TBC)

Start date, end date, total duration (weeks) *

Maximum duration, from as early as practical.

The benefit to the School *

Greater completeness of COMP25111 redevelopment for AY17. A more acceptable course.

The benefit to the student *

Some degree of design and much implementation work in an eLearning environment. Some programming practice leading to a real (and therefore, necessarily, robust!) product.

The opportunity to gain a deeper insight into the education process and appreciate the needs of a range of students. Making a positive contribution to the education of future Computer Science students over the next decade and more.

Skills needed by the student. *

These will vary according to the way the intended practical work develops. It is clear that background understanding of the subject area is important, which will include principles such as virtual memory management, process scheduling and interaction, interrupt and exception handling and so forth.

The willingness to help design and develop inspirational ideas for the presentation of such topics is important, as is the willingness to experiment with novel ways of teaching. Student feedback on how the 'ideal' course might run is important in this initiative. A critical attitude to exercise development will be needed.

It is envisaged that there will be increased computer-based interaction in future practicals. This implies the development of systems which will accept, 'assess' and give feedback on 'exercises' automatically; although the emphasis is intended to be on learning there would also be some record-keeping involved so communications with a data recording system is needed. The implementation of particular 'exercises' is liable to require some form of programming skills: a convenient approach may be presentation via a web browser although other options are certainly possible. The interface design is clearly also an important issue.

Details of the work that the student would do *

Assist in the design and do the majority of the implementation work of 'laboratory exercises' for a revamped COMP25111 (Operating Systems) second year module.

Defining precise specifications is part of the work. There will be multiple, diverse concepts to illustrate. A possible example may serve to illustrate the current thinking.

Process scheduling: Imagine a display showing the state of a number of processes in a multi-tasking system. The number of processes (and processors?) may be varied and the behaviour of processes specified at a primitive level (e.g. `busy(<time>); input(<device>); wait(<time>); etc.`). An animated and interactive display could show the behaviour of the system as processes execute, stall, wait, signal etc. Perhaps a problem could be set up automatically - ideally different for each student - where they had to demonstrate their understanding of what they are seeing. The result could be assessed according to some rules and a mark recorded - and feedback generated - 'instantly' without human intervention.

Hopefully, individual topics can be assembled into a portfolio with a common look-and-feel.

Infrastructure requirements and any required staff support other than the project supervisor *

No hardware requirements other than working computers; no time to build stuff in any case! Note potential interaction with computer down-time/reimaging.

Supervision arrangements throughout the duration of the project (named staff and dates covering the entire duration) *

J. Garside - hopefully some other(s) too.

Location of the project work (building/room) *

Pref. Kilburn; lab. immaterial. Results must run on any of our (Unix) systems from next year.