In-Course Assessment Brief

CPS Programme Academic Year 2008/2009

Module:	**Introduction to Open Source Technologies**
Assignment 1:	Coursework report.
Centre:	Computing, Telecommunications and Networks
Module Co-ordinator:	Richard Kay
Set Date:	See ECMS My Course on the intranet for details
Submission Deadline:	
Assessment Weighting:	
Submission Method:	Submitted through the IT Helpdesk on level 3.
Nominal time to complete this assignment:	20 Hours
Brief Assessment Details	Students will individually agree topics to research with the module coordinator, either from a list provided by the module coordinator, or a subject of similar relevance and interest chosen by the student and agreed with the module coordinator. Choice of selected topic must occur by 26 June. This topic will involve a case study concerning a major open source software development management transition or fork. Students will write a 2000 word report on the topic chosen.
Individual Assessment:	Individual assessment. The work you submit shall be your own and not the product of collaboration with anyone else. Plagiarism will be penalised. Group assessments. Members of the group shall be listed in the assessment.

If you should fail this module you will be permitted to be re-assessed on up to three occasions. If you fail to attend or to submit work for re-assessment at the next opportunity you will be deemed to have exhausted one of the opportunities.

Learning Outcomes to be Assessed:

1. Understand how the OS community operates and to obtain and provide support for OS products.
2. Have a working knowledge of OS licenses, community norms, development and support processes.

Assessment Details:

Students will individually agree topics to research with the module coordinator, either from a list provided by the module coordinator, or a subject of similar relevance and interest chosen by the student and agreed with the module coordinator. Choice of selected topic must occur by 26 June 2009. This topic will involve a case study concerning a major open source software development management transition or fork. Suitable studies will include one the following, or an event of similar interest and significance suggested either by the student or module coordinator.

a. The transition of Linux kernel development between BitKeeper and Git source code control systems occurred relatively recently. What were the technical, community, personal and licensing drivers behind this change. How can the success or failure of this transition be measured ?

b. Keith Packard led the Xfree86 to X.org fork. What caused this fork, and what does this demonstrate concerning bazaar and cathedral style software engineering methodologies and what were the outcomes for open source graphics display driver development and capabilities ? Are any objective indications of these outcomes available concerning hardware support and if so which, and if not why not ?

c. Why did Apple choose a BSD kernel for their operating system ? What other choices did they have ? What other evidence of the use of BSD code within proprietary systems exists ? Do the companies using this code contribute any changes upstream and if so why? If they don't what benefits would they receive if they did and what costs would they incur ?

d. Research, compare and contrast the evolving relationship between development and stable Linux kernel versions and staging trees starting with the release of 2.4.0. Does the shorter period during which major changes can be merged during 2.6 point releases compared to the 2.5 development kernel prevent such major Kernel changes from occurring ? What is the effect of this on the numbers of users and developers involved in testing ?

Students will write a 2000 word report on the topic chosen.

Assessment Criteria:

See table below

Table of Assessment Criteria and Associated Grading Criteria

Assessment Criteria →	**1. Extent and quality of investigation and background research**	**2. Understanding demonstrated by student of issues concerning studied case topic**	**3. Quality of referencing of sources used.**	**4. Report style, presentation and quality of writing.**
Weighting:	**30%**	**40%**	**10%**	**20%**
Grading Criteria **0 – 29%**	Little evidence of independent work by student.	None demonstrated or unduly derivative of published sources	No evidence of judgement applied as to sources used.	Sloppy and disorganised.
30 – 39%	Standard sources used only. No attempt to go further.	Largely derivative work demonstrating little understanding if any.	Unclear indication of who wrote what within report.	Poorly presented work. Minor effort evident but too little.
40 – 49%	Some relevant materials accessed but investigation weak.	Enough understanding of background demonstrated to pass but large gaps.	Some indication but significant parts of report including sections of referenced sources inappropriately cut and pasted.	Some effort evident but very limited quality of student's own writing.
50 – 59%	Fair selection of relevant materials found	Fair understanding of issues raised by subject investigated.	Fair use made of referenced and relevant materials, within students own integrative work.	Fair effort evident, but with significant shortcomings.
60 – 69%	Good selection of materials and issues uncovered and raised but incomplete.	Good understanding of most issues clearly demonstrated.	All materials appropriately used and clearly referenced but some errors or limited gaps.	Good, well presented report. Clear writing style.
70+%	Excellent coverage of topic with all important sources of information found.	Full and critical understanding of software development and community issues clearly demonstrated.	Full and clear references based on concise and critical use of sources.	Excellent presentation and writing. Fully authentic report.
Checklist				