**THE UNIVERSITY OF HUDDERSFIELD**

School of Computing and Engineering

**ASSIGNMENT SPECIFICATION**

| Module details | |
| --- | --- |
| Module Code | CFS2101 |
| Module Title | Computer Organisation and Architecture |
| Course Title/s | Computer Science  Computer Science with Cyber Security  Computer Science with Game Programming  Computing  Software Engineering  Computer Systems Engineering |

| Assessment weighting, type and contact details | | |
| --- | --- | --- |
| Title | Machine and Assembly Language | |
| Weighting | 50% | |
| Mode of working for assessment task | Individual  Note : if the assessment task is to be completed on an individual basis there should be no collusion or collaboration whilst working on and subsequently submitting this assignment. | |
| Module Leader | Louie Qin | Contact details:louie.qin@hud.ac.uk |
| Module Tutor/s | Louie Qin, Minsi Chen | |

| Submission and feedback details | |
| --- | --- |
| Hand-out date | 17-11-2020 |
| How to submit your work. | Brightspace Submission |
| Submission date/s and times | 23:59 18-12-2020 |
| Expected amount of independent time you should allocate to complete this assessment | 50 hrs |
| Submission type and format | Source files |
| Date by which your grade and feedback will be returned | 01-02-2021 |

| Additional guidance information | |
| --- | --- |
| Your responsibility | It is your responsibility to read and understand the [University regulations regarding conduct in assessment](https://www.hud.ac.uk/registry/current-students/taughtstudents/conductinassessment/).  Please pay special attention to the assessment regulations (section 4) on Academic Misconduct.  In brief: ensure that you;   1. DO NOT use the work of another student - this includes students from previous years and other institutions, as well as current students on the module. 2. DO NOT make your work available or leave insecure, for other students to view or use. 3. Any examples provided by the module tutor should be appropriately referenced, as should examples from external sources.   Further guidance can be found in the SCEN Academic Skills Resource and UoH Academic Integrity Resource module in Brightspace.  If you experience difficulties with this assessment or with time management, please speak to the module tutor/s, your Personal Academic Tutor, or the School’s Guidance Team. ([sce.guidance@hud.ac.uk](mailto:sce.guidance@hud.ac.uk)). |
| Requesting a Late Submission | You are reminded to ‘back-up’ your work as late submission requests will not be given for lost work, which includes work lost due to hardware and software failure/s.  Late submission requests will only be approved if you can demonstrate genuine, unexpected circumstances along with independent supporting evidence (e.g. medical certificate) that may prevent you submitting an assessment on time.  Submit your request for Late Submission via MyHud/MyStudies within 2 working days of the due date.  Late submission requests, up to a maximum of 10 working days, but typically 1-5 working days, will be considered provided that there is appropriate evidence which clearly indicates reasons for the request.  You will have 5 working days after submitting a request to provide the evidence. Failure to submit evidence will result in the request being rejected and your work being marked as a late submission (see below).  If you are unable to submit work within the maximum late submission period of 10 days, contact the School’s Guidance Team. ([sce.guidance@hud.ac.uk](mailto:sce.guidance@hud.ac.uk)), as you may need to submit a claim for Extenuating Circumstances (ECs). |
| Extenuating Circumstances (ECs) | An EC claim is appropriate in exceptional circumstances, when an extension is not sufficient due to the nature of the request, or it concerns an examination or In-Class Test (ICT).  You can access the [EC claim form](https://www.hud.ac.uk/registry/current-students/taughtstudents/extenuatingcircumstances/) on the Registry website; where you can also find out more about the process.  You will need to submit independent, verifiable evidence for your claim to be considered.  Once your EC claim has been reviewed you will get an EC outcome email from Registry.  If you are unsure what it means or what you need to do next, please speak to the [Student Support Office](mailto:sce.student.support@hud.ac.uk) – Room SJ1/01  An approved EC will extend the submission date to the next assessment period (e.g July resit period). |
| Late Submission  (No ECs approved) | Late submission, up to 5 working days, of the assessment submission deadline, will result in your grade being capped to a maximum of a pass mark.  Submission after this period, without an approved extension, will result in a 0% grade for this assessment component. |
| Tutor Referral available | NO |
| Resources | * Please note: you can access free Office365 software and you have 1 Tb of free storage space available on Microsoft’s OneDrive – [Guidance on downloading Office 365](https://students.hud.ac.uk/media/universityofhuddersfield/content/documents/computingservices/office365/Office365-AppsDownloadGuide.pdf). |

**Disassembling Machine Language**

1. **Assignment Aims**

Machine language and instruction set architecture (ISA) are integral to the design and architecture of a processor. They also serve as an interface between higher-level programming languages and the underlying hardware platform.

This aim of this assignment is to further develop your understanding of ISA by tasking you to develop a disassembler for the y86 ISA.

1. **Learning Outcomes:**

On successful completion of this assignment, you will meet the following learning outcomes.

***Knowledge and Understanding Outcomes***

3. Describe how data are represented in computers

***Ability Outcomes***

4. Construct solutions to problems related to computer systems and their components

6. Develop a small computer with a minimal instruction set

1. **Assessment Brief**

You will be given a partially completed y86 disassembler written in C. Your aim is to complete its functionality so that it is able to disassemble the instructions defined the y86 architecture. The specification of all y86 instructions is available on Brightspace as part of the teaching materials.

To help you better manage your work, this assignment is further divided into the following three tasks. You will be given a starting source along with test files for each task. These files can be downloaded from Brightspace under the Assessment area.

**Task 1 Decode y86 Opcodes (20 pts):**

You are required to decode the opcode part of a list of given instructions. Each decoded opcode must be correctly displayed as its corresponding assembly mnemonic (i.e. symbolic name), e.g., in y86 the opcode 0x00 should be displayed as halt.

**Task 2 Decode y86 Instruction Operands (40 pts):**

Please note: The starting point of this task is based on the solution of task 1, i.e. your disassembler should be able to convert all opcodes into their corresponding mnemonic.

The objective of this task is to correctly decode the operands associated with an instruction. There are only three instructions with no operand, i.e. nop, halt and ret. Therefore, the majority will require the disassembling of operands. For example: the y86 machine code 0x6060 is ‘addl %esi, %eax’ in assembly.

**Task 3 Disassemble y86 Binary Machine Code (30 pts):**

This task is designated as a challenge task which is built on your solution to task 1 and 2. For this task, you will disassemble programs presented in y86 bytecode (machine code) and convert them into the corresponding y86 assembly.

You are provided with three test programs in both assembly and bytecode. Each program is stored in C as a contiguous array of unsigned char (bytes). The bytecode file reader has been written for you so you can dive straight into disassembling the bytecode. Your solution should emulate a basic von Neumann machine by disassembling a program based on the position and advancement of the program counter (PC).

Please note: Some test programs have data region. You do not have to decode the data region. You can check the disassembled code against the provided assembly code.

1. **Submission**

You must submit the source files for the completed tasks to the designated submission point on Brightspace by *23:59 Friday 18/12/2020*.

The source file for each task should be named in using the following convention:

* y86disasm-task1.c
* y86disasm-task2.c
* y86disasm-task3.c

Please note: You can upload more than one file per submission. You must make sure all source files are uploaded before confirming your submission. Multiple submissions are allowed. However, only the latest submission will be stored and marked.

1. **Marking Scheme**

The total attainable marks from this assignment is 100 which is further granulated based on the following table. Marks will also be awarded for partially correct solutions.

|  |  |
| --- | --- |
| ***Criteria*** | ***Available Marks*** |
| Task 1 Decode y86 Opcodes | 20 |
| Task 2 Decode y86 Instruction Operands | 40 |
| Task 3 Disassemble y86 Binary Machine Code | 30 |
| Overall program structure including variable naming, formatting and commenting | 10 |