

高知工業高等専門学校		開講年度	令和02年度 (2020年度)	授業科目	アセンブリ II
科目基礎情報					
科目番号	I4002	科目区分	専門 / 必修		
授業形態	講義	単位の種別と単位数	履修単位: 1		
開設学科	SD 情報セキュリティコース	対象学年	4		
開設期	後期	週時間数	2		
教科書/教材	参考書: 「情報処理技術者テキスト プログラミング入門 CASLII」 実教出版, 教材: Raspberry Pi				
担当教員	岡村 修司				
到達目標					
1. 基本的なデータ構造やアルゴリズムをアセンブリ言語で記述し、デバッグできる。 2. 分割したソースファイルから、実行可能ファイルを生成する仕組みを理解し、生成することができる。 3. 与えられた仕様を理解し、フローチャートで記述することができる。 4. makeの仕組みを理解し、実行可能ファイルを生成することができる。					
ルーブリック					
	理想的な到達レベルの目安	標準的な到達レベルの目安	未到達レベルの目安		
評価項目1	基本的なデータ構造やアルゴリズムをアセンブリ言語で記述し、デバッグできる。	基本的なデータ構造やアルゴリズムをアセンブリ言語で記述できる。	基本的なデータ構造やアルゴリズムをアセンブリ言語で記述できない。		
評価項目2	分割したソースファイルから、実行可能ファイルを生成する仕組みを理解し、生成することができる。	分割したソースファイルから実行可能ファイルを生成する仕組みを理解している。	分割したソースファイルから実行可能ファイルを生成する仕組みを理解できていない。		
評価項目3	与えられた仕様を理解し、フローチャートで記述できる。	与えられた仕様を理解できる。	与えられた仕様を理解できない。		
評価項目4	makeの仕組みを理解し、分割したソースファイルから実行可能ファイルを生成できる。	分割したソースファイルから実行可能ファイルを生成できる。	分割したソースファイルから実行可能ファイルを生成できない。		
学科の到達目標項目との関係					
教育方法等					
概要	前期で学んだアセンブリIの内容を基礎とし、様々なデータ構造やアルゴリズムをアセンブリ言語で記述する。複雑なプログラムを記述する過程で、C言語のポインタ操作への理解を深め、フローチャートを読むだけでなく、記述することを目指す。				
授業の進め方・方法	基本的に演習形式で行う。スライドを用いて、学習内容の説明を行う。学習内容をまとめたプリントを配布するので、これを参考にプログラミングを行う。プログラミングを通じて学習内容を理解する。レポートを作成することで、学習内容を確認し、技術文書の作成法を学ぶ。授業はセキュリティ演習室のLinux環境で行う。				
注意点	試験の成績を70%、平素の学習状況等(課題)を30%とし、総合的に評価する。成績評価は中間と期末の各期間の評価の平均とする。事前学習として、配布するプリントを読み、授業内容を理解しておくこと。授業を受けても理解できない内容は、質問するなどして、理解するための努力を行うこと。コーディング中は他の学生と相談してもよいが、各自主体的に取り組むことが重要である。事後学習として、授業で学んだ内容をレポートにまとめ、指定された期日までに提出する事。 アセンブリIの内容をよく理解している事。				
授業計画					
	週	授業内容	週ごとの到達目標		
後期	3rdQ	1週	ガイダンス make (1)	授業の進め方と評価方法を理解する。 makeの仕組みを理解し、説明できる。	
		2週	make (2)	makeによるプログラム開発が行える。	
		3週	基本的なプログラム (8)	2進数出力処理のフローチャートを作成し、プログラムを記述できる。 makeで実行可能ファイルを生成する。	
		4週	基本的なプログラム (9)	10進数出力処理のフローチャートを作成し、プログラムを記述できる。 makeで実行可能ファイルを生成する。	
		5週	基本的なプログラム (10)	16進数出力処理のフローチャートを作成し、プログラムを記述できる。 makeで実行可能ファイルを生成する。	
		6週	基本的なプログラム (11)	2進数入力処理のフローチャートを作成し、プログラムを記述できる。 makeで実行可能ファイルを生成する。	
		7週	基本的なプログラム (12)	10進数入力処理のフローチャートを作成し、プログラムを記述できる。makeで実行可能ファイルを生成する。	
		8週	基本的なプログラム (13)	16進数入力処理のフローチャートを作成し、プログラムを記述できる。 makeで実行可能ファイルを生成する。	
	4thQ	9週	基本的なプログラム (14)	文字数カウント処理のフローチャートを作成し、プログラムを記述できる。makeで実行可能ファイルを生成する。	
		10週	基本的なプログラム (15)	文字列比較処理のフローチャートを作成し、プログラムを記述できる。	
		11週	基本的なプログラム (16)	文字検索処理のフローチャートを作成し、プログラムを記述できる。 makeで実行可能ファイルを生成する。	
		12週	基本的なプログラム (17)	度数分布表出力処理のフローチャートを作成し、プログラムを記述できる。makeで実行可能ファイルを生成する。	

	13週	基本的なプログラム (18)	多分岐処理のフローチャートを作成し、プログラムを記述できる。 makeで実行可能ファイルを生成する。
	14週	基本的なプログラム (19)	リスト処理のフローチャートを作成し、プログラムを記述できる。 makeで実行可能ファイルを生成する。
	15週	基本的なプログラム (20)	整列処理のフローチャートを作成し、プログラムを記述できる。 makeで実行可能ファイルを生成する。
	16週		

モデルコアカリキュラムの学習内容と到達目標

分類	分野	学習内容	学習内容の到達目標	到達レベル	授業週	
専門的能力	分野別の専門工学	情報系分野	プログラミング	代入や演算子の概念を理解し、式を記述できる。	4	
				プロシージャ(または、関数、サブルーチンなど)の概念を理解し、これらを含むプログラムを記述できる。	4	
				変数の概念を説明できる。	4	
				データ型の概念を説明できる。	4	
				制御構造の概念を理解し、条件分岐を記述できる。	4	
				制御構造の概念を理解し、反復処理を記述できる。	4	
				与えられた問題に対して、それを解決するためのソースプログラムを記述できる。	4	
				ソフトウェア生成に必要なツールを使い、ソースプログラムをロードモジュールに変換して実行できる。	4	
				与えられたソースプログラムを解析し、プログラムの動作を予測することができる。	4	
				主要な言語処理プロセッサの種類と特徴を説明できる。	4	
				ソフトウェア開発に利用する標準的なツールの種類と機能を説明できる。	4	
				プログラミング言語は計算モデルによって分類されることを説明できる。	3	
				主要な計算モデルを説明できる。	3	
				要求仕様に従って、標準的な手法により実行効率を考慮したプログラムを設計できる。	4	
			要求仕様に従って、いずれかの手法により動作するプログラムを設計することができる。	4		
			要求仕様に従って、いずれかの手法により動作するプログラムを実装することができる。	4		
			要求仕様に従って、標準的な手法により実行効率を考慮したプログラムを実装できる。	4		
			システムプログラム	形式言語の概念について説明できる。	3	
				オートマトンの概念について説明できる。	3	
				コンパイラの役割と仕組みについて説明できる。	3	
形式言語が制限の多さにしたがって分類されることを説明できる。	3					
正規表現と有限オートマトンの関係を説明できる。	3					

評価割合

	試験	平素の学習状況	相互評価	態度	ポートフォリオ	その他	合計
総合評価割合	70	30	0	0	0	0	100
基礎的能力	50	20	0	0	0	0	70
専門的能力	20	10	0	0	0	0	30
分野横断的能力	0	0	0	0	0	0	0