

開講期間	配当年	単位数	科目必選区分
3年前期	3	2	選択（教職「情報」は選択）
担当教員			
國持 良行			
添付ファイル			

講義概要	C/C++などの高位言語を機械語に翻訳する処理系をコンパイラと呼び、PythonやPHPなどのコードを解釈実行する処理系をインタープリタと呼ぶ。この講義ではUNIXの環境下で構文解析部分を生成するflex/bison (lex/yaccの後継)を用いることを前提にして、コンパイラやインタープリタについて学び、簡単な処理系を段階的に作成する演習を行なう。また、オートマトンと形式言語理論にもとづく字句解析や構文解析についても理解する。		
授業計画	1	言語処理系の導入・基本情報技術者試験の問題 概要：基本情報技術者試験(FE)の問題を例題にして、コンパイラの概要を学習する。また、Linux(WSL)、VS Code、gcc、g++、make、flex、bisonなどのパッケージをインストールする。 演習：FEの過去問を解く(AL①)。インストール作業をする(AL②③)。 準備学習：テキストの1.1節と1.2節を予習する。 課題：インストール作業のまとめと確認をする(AL④)	
	2	形式言語の基本 概要：形式言語の基本的な記法と定義、チョムスキー階層(0型～3型文法)、文法と導出を学習する。 演習：語の導出過程、言語の演算を計算する(AL①②③)。 準備学習：テキストの2.1節を予習する。 課題：数式処理システムGAPのSemigroups Packageを調べてみよう(AL④)	
	3	正規言語(3型言語)と有限状態オートマトン 概要：有限(状態)オートマトン(Finite Automata)の定義、記法、図式表現などを学ぶ。決定性(DFA)と非決定性(NDFA)の等価性を理解する。 演習：正規(0型)文法と有限オートマトンの等価性、DFAとNDFAの変換を演習する(AL①②③)。 準備学習：テキストの2.3節を予習する。 課題：DFAとNDFAの等価性の証明を調べる。数式処理システムGAPのAutomata Packageを調べてみよう(AL④)	
	4	正規表現 概要：正規表現(Regular Expressions)の定義、有限(状態)オートマトンとの等価性を理解する。 演習：正規表現からオートマトン、オートマトンから正規表現への変換(AL①②③)。 準備学習：テキストの2.3節を予習する。 課題：字句解析ソフトウェアflexを調べてみよう(AL④)	
	5	flexの演習 概要：flexは正規表現をもとにテキストの字句解析処理を行う。makeコマンドについても学ぶ 演習：正規表現をflexに与え、テキスト処理の演習を行う(AL①②③)。 準備学習：テキストの3章を予習する。 課題：makeコマンドを使ってプログラムの開発効率を上げよう(AL④)	
	6	文脈自由(2型)文法と文脈自由(2型)言語 概要：文脈自由言語(Context-free Languages, CFL)は、文脈自由文法で生成される言語である。また、プッシュダウンオートマトン(Push-down Automata, PDA)で受理される言語と一致する。 構文木(導出木)やBNF記法についても学ぶ。 演習：構文木を書く、最左/最右導出の演習を行う(AL①②③)。 準備学習：テキストの4.1章を予習する。 課題：曖昧性、決定性、CFLのサブクラス(Simple Languages, DCFLなど)について調べよう(AL④)	
	7	構文解析の手法 概要：文脈自由文法のサブクラスであるLL(k), LR(k)文法について学ぶ。また、文脈自由文法の標準形であるチョムスキーの標準形について学ぶ。構文解析アルゴリズムであるCYK法を実装する。 演習：一般の文脈自由文法を標準形に変換する(AL①②③)。 準備学習：テキストの4.2節と4.3節を予習する。 課題：グライバッハの標準形、ポンプの補題、パルクの定理、semi-linearについて調べよ(AL④)。	
	8	gccプログラミング — CYK法の実装 概要：CYK法をC言語で実装する。C言語の復習をする。 演習：CYK法で与えられた語の構文解析を行う(AL①②③)。 準備学習：テキストの4.4節を読んで、プログラムを作成する 課題：4.5節上昇型構文解析、4.6節LRパーサを調べよう(AL④)	
	9	上昇型構文解析 概要：上昇型構文解析であるLR(k)法について学ぶ 演習：LR(k)法によって構文解析の演習を行う(AL①②③)r 準備学習：4.7節を読んで上昇型構文解析を予習する。 課題：BNFで書かれた文法の構文解析の演習問題を解く(AL④)	
	10	gccプログラミング — アセンブリ言語 概要：アセンブリ言語(ディレクティブ、命令形式、)を学ぶ。 演習：アセンブリ言語プログラミング(AL①②③) 準備学習：参考文献の計算機アーキテクチャの関連箇所を予習する。 課題：アセンブリ言語の応用例を調べる(AL④)	

	11	gccプログラミング ― 計算機アーキテクチャ 概要：計算機アーキテクチャの概要、CPU、レジスタ、命令体系、メモリを復習する。 演習：基本情報技術者試験のハードウェア問題を解く (AL①②③) 準備学習：参考文献の計算機アーキテクチャの関連箇所を予習する。 課題：CPU命令、GPU命令を調べてみよう (AL④)
	12	gccプログラミング ― アセンブリ言語 概要：アセンブリ言語(ディレクティブ、命令形式、)を学ぶ。 演習：アセンブリ言語プログラミング (AL①②③) 準備学習：参考文献の計算機アーキテクチャの関連箇所を予習する。 課題：アセンブリ言語の応用例を調べる (AL④)
	13	flexとbisonによるアセンブリ言語プログラムの生成 概要：字句解析器 flex、構文解析器 bison、Makeファイル、BNF記法、スタックマシン。 演習：式を構文解析して、アセンブリ言語プログラムを生成し、実行結果を確認する (AL①②③) 準備学習：テキストの該当部分を予習する。 課題：授業で作成したプログラムのデバッグを行い、完成させる (AL④)
	14	インラインアセンブラ 概要：インラインアセンブラ、C言語、gcc. rdtsc命令 演習：プログラムを作成し、実行結果を確認する (AL①②③) 準備学習：テキストを該当部分を予習する。 課題：授業で作成したプログラムのデバッグを行い、完成させる (AL④)
	15	まとめのテスト 概要：授業のまとめと総括を行う。 演習：授業で作成したコンテンツやプログラムのまとめとテストを行う (AL③) 準備学習：作成物を整理整頓しておくこと。 課題：総括をもとにできなかった問題をレポートする (AL④)
	16	定期試験
授業形態	講義とノートパソコンを使う演習を行う。 アクティブラーニング：①:14回, ②:14回, ③:15回, ④:15回, ⑤:0回, ⑥:0回	
達成目標	1) オートマトンと形式言語理論の基礎を理解している。 2) 数式処理システムを使い、半群、モノイド、オートマトンの取り扱いができる。 3) 字句解析や構文解析のアルゴリズムを理解し、その実装ができる。 4) アセンブリ言語の簡単なプログラミングができる。 5) flexやbisonを使い、アセンブリ言語へのコンパイルができる。	
評価方法・フィードバック	演習 (40%)：小テストとレポート(プログラム作成)。 定期試験 (60%)：知識・思考を問う問題、授業内で使用した処理系に関する問題を出題する。 原則として、レポート・小テスト等のフィードバックは次回以降の授業内で実施する。	
評価基準	秀(1～5)：100～90、優(1～5)：89～80、良(1～4)：79～70、可(1～3)：69～60、不可：59以下。	
教科書・参考書	教科書：プリントを配布する 参考書：John Levine「flex & bison: Text Processing Tools」O'REILLY(Kindle版) 参考書：Brian Ward「スーパーユーザーなら知っておくべきLinuxシステムの仕組み」インプレス (2022/3/8) 参考書：大崎 博之「独習アセンブラ 新版」翔泳社：新版 (2021/9/8)	
履修条件	「OS(UNIX)」「プログラミング実践演習2」に合格していることが望ましい(ファイル・ポインタ・構造体・共用体の知識が必要)	
履修上の注意	ネットワークに接続可能なパソコンを持参すること。	
準備学習と課題の内容	①初回までにシラバスを読み授業項目や目的を理解しておくこと。 ②毎回3時間以上授業外に予習と復習をして授業に臨むこと。	
ディプロマポリシーとの関連割合(必須)	知識・理解:40%, 思考・判断:20%, 関心・意欲:20%, 態度:10%, 技能・表現:10%	
DP1 知識・理解		
DP2 思考判断		
DP3 関心意欲		
DP4 態度		
DP5 技能・表現		