

2017 年度 コンピュータリテラシー I

Tue Feb 28 12:17:05 JST 2017 年度版

Name:

このハンドアウトは再発行、再配布しません。大切に扱ってください。

ID:

注意事項

以下の注意事項を良く読み、指示に従うことを受講者に要求する。また、以下にある指示に従わない場合は減点等の対象となる。不明な点があれば、自ら担当教員に尋ねること。

- ・ 講義の出席や演習の参加、定期試験への参加も成績の評価に影響がある。不参加の場合は原則として、単位を与えない。特に、試験についてはすべて参加すること。そうでなければ不合格とする。
- ・ 提出物が 20 個以下の者は不合格になる。
- ・ 課題は原則として演習時間内に提出すること。指示がある場合にはそれに従うこと。
- ・ 宿題や Quiz は指定された締め切りを守ること。締め切り時間は指定された日の 20 時とする。
- ・ 課題と宿題は必須。Quiz と競争的課題は選択であり、付加的なポイントを得られる。課題と宿題と試験の結果を合計して満点であれば、最高得点になる。
- ・ File は指定されたディレクトリに指定されたファイル名で提出すること。
- ・ File やディレクトリは指定されたアクセスパーミッションに設定すること。
- ・ File の文字コードは指定されたものにすること。
- ・ レポート用紙は A4 の用紙を用いること。
- ・ 複数枚のレポートの提出の場合は必ずホッチキスで左上を綴じて提出すること。
- ・ 無駄な印刷をしないために印刷前にプレビューで確認すること。印刷物は責任を持って扱うこと。
- ・ レポート作成に利用した文献やウェブページが引用文献としてレポート内に明記されていなければ評価しない。
- ・ 引用文献における記述をそのまま転用したものや、転用の割合が高いと判断されるものは評価しない。
- ・ 他の学生のファイルの複製は採点しない。複製をレポートとして提出する、試験で不正を働くなどの不正行為を行った者には単位を与えない。また、定期試験における不正行為と同様に扱う。
- ・ 提出物や試験の配点はあくまでも予定であり、多少の変更はあり得る。
- ・ 期末試験は期末試験期間内に実施する。
- ・ リテラシーについての問い合わせは正規の大学のメールアドレスからのものか、正規のメールアドレス宛にCc がされているものに限り受け付ける。

講義と演習の予定						
日程	講義や演習のテーマ			課題等	得点	締切
						コメント
lec-1	4/10	コンピュータリテラシーとは何か。ディレクトリとアクセス制御。		レポート 1 Quiz	40 10	来週の演習の時間 演習最終日
	ex-1	4/10	User account and Login, Typing(1: Fundamental)	課題 1 宿題 1	10 10	当日 当日
	ex-2	4/12	Password, Keyboard, Mouse, X Window System(1), Emacs(1: Tutorial, Kanji), File(1), Typing(2: Exercise)	課題 2	10	当日
lec-2	4/17	UNIX、メールとニュース		課題 3 宿題 3	10 10	当日 2週間後
	ex-3	4/17	UNIX(1: Commands), Emacs(2: Point, Mark, Region, Buffer), News	課題 4 宿題 4	10 33	当日 2週間後
	ex-4	4/19	UNIX(2: Directory, Filtering, Re-direction), X Window System(2), Typing(3: C examples)			
lec-3	4/24	ネットワーク、WiFi		課題 5 宿題 5	20 35	当日 2週間後
	ex-5	4/24	How to Print, E-mail, Typing(4: LaTeX examples)	課題 6	10	当日
	ex-6	4/26	UNIX(3: Commands, Permissions, File system)			
lec-4	5/1	文字コード、日本語入力、正規表現		課題 7	10	当日
	ex-7	5/1	Convenient Tools (1: grep and sed)			
lec-5	5/8	ファイル管理、情報収集		課題 8 宿題 8	10 10	当日 2週間後
	ex-8	5/8	Convenient Tools (2: awk)	課題 9	10	当日
	ex-9	5/10	How to get information			
lec-6	5/15	ネットワークを利用する場合のエチケットと倫理		課題 10 宿題 10	10 10	当日 2週間後
	ex-10	5/15	Network Etiquette and Laws (Case study)			
lec-7	5/22	文書作成、LaTeX の基礎		課題 11 宿題 11	10 10	当日 当日
	ex-11	5/22	LaTeX(1: Sample, Preview)	課題 12 宿題 12	10 10	当日 2週間後
	ex-12	5/24	LaTeX(2: Fonts, Size, Equation, Figure)			
lec-8	5/29	文書作成、LaTeX の参照、作図 (Draw)		課題 13 宿題 13	20 10	当日 4週間後
	ex-13	5/29	Drawing Tools(tgif)	課題 14 宿題 14	10 10	印刷用紙 File
	ex-14	5/31	Examination(1: UNIX)	試験 1	50	
	ex-15	6/2	LaTeX(3: Tabular, Macro)	課題 15 宿題 15	10 10	当日 2週間後
				File		

日程	講義や演習のテーマ			課題等	得点	締切	コメント
lec-9	6/12	学術論文の作成, スライド		課題 16	10	当日	File
	ex-16	6/12	LaTeX(4: Slide)	課題 17	10	当日	File
	ex-17	6/21	PSTricks, Animation				
lec-10	6/19	著作権およびコンピュータやネットワーク利用に関する法律, セキュリティ		課題 18	10	当日	File
	ex-18	6/19	Copyrights(Case study), Octave	課題 19	10	当日	File
	ex-19	6/14	Making Graphs(Xmgrace, Gnuplot, Mathematica), Images(ImageMagick)	宿題 19	40	演習最終日	File
lec-11	6/26	情報管理					
	ex-20	6/26	File Compress, Web Browser Setting, FTP(1)	課題 20	10	当日	File
	ex-21	6/28	Examination(2: Network Etiquette and Laws, Copyrights)	試験 2	100		
lec-12	7/3	シェルそ の 1					
	ex-22	7/3	Shell Environment, X Window System Environment, FTP(2)	課題 22	10	当日	File
	ex-23	7/5	File Convert, Backup Method, R	課題 23	10	当日	File
				宿題 23	10	演習最終日	File
lec-13	7/10	シェルそ の 2					
	ex-24	7/10	Shell Commands, Shell Programming (1)	課題 24	10	当日 演習最終日	File File
	ex-25	7/12	セキュリティ	課題 25	10	当日 演習最終日	File File
	ex-26	7/19	Network Resources, Process Management, Internet	課題 26	10	当日 演習最終日	File File
				宿題 26	10	演習最終日	File
lec-14	7/24	簡単なシェルプログラミング					
	ex-27	7/24	Shell Programming (2), StarSuite SpreadSheet	課題 27	10	当日 演習最終日	File File
	ex-28	7/26	Make, スペルチエック, StarSuite Presentation	課題 28	10	当日 演習最終日	File File
				宿題 28	10	演習最終日	File
lec-15	7/31	期末試験の説明など(試験は試験期間に実施)		試験 3	100		
	ex-29	7/31	Trouble Shooting, How to Report Troubles	課題 29	10	当日 演習最終日	File File
	ex-30	8/1	E-mail, Calculator, Convenient Tools	課題 30	10	当日	File

目次

第 1 章 コンピュータリテラシーとは何か, ディレクトリとアクセス制御	40
1.1 コンピュータリテラシーとは	1
1.2 リテラシーを構成する3つの技能	8
1.2.1 テキストファイルの作成	8
1.2.2 文書の清書、印刷	9
1.2.3 コミュニケーション	9
1.2.4 その他	9
1.3 プロフェッショナルとしての倫理	9
1.4 第1回演習予告	9
1.5 第1回レポート宿題	10
1.6 Users	10
1.7 ファイル	10
1.7.1 ディレクトリ(フォルダ)	10
1.7.2 オーナーとパーミッション	12
1.8 参考資料	13
1.8.1 UNIX	13
1.8.2 ネットワークとセキュリティ	13
1.8.3 Emacs	13
1.8.4 コンピュータの基礎、一般的知識	13
1.9 Quiz	14
1.10 競争的課題	21
1.10.1 作品のエントリ	21
1.10.2 批評のエントリ	21
1.10.3 投票	21
1.10.4 参加賞	21
第 2 章 UNIX, Mail, News, Web	22
2.1 はじめに	22
2.2 UNIX	22
2.2.1 なぜ演習室はWindowsパソコンではなく、UNIXワークステーションか?	22
2.2.2 Mac OS X 端末	23
第 5 章 ファイル管理、情報収集	40
5.1 ファイル管理	5.1
5.1.1 ファイル名とワイルドカード	5.1.1
5.1.2 ファイル削除の注意	5.1.2
5.1.3 ファイルのリストア(復旧)	5.1.3
5.1.4 ヒストリ機能	5.1.4
5.1.5 ファイル名補完機能	5.1.5
5.2 情報の収集の方法	5.2
5.2.1 アプリケーションについての情報	5.2.1
5.2.2 個人所有の情報 - Home Directories	5.2.2
5.2.3 世界中から発信される情報 - Internet	5.2.3
5.2.4 図書館の利用	5.2.4
5.2.5 国立国会図書館	5.2.5
5.2.6 http://arxiv.org/	5.2.6
第 6 章 ネットワーク利用のエチケットと倫理	45
6.1 Network Etiquette = Netiquette	6.1
6.1.1 技術	6.1.1
6.1.2 マナー	6.1.2
6.1.3 アカウントロック	6.1.3
6.1.4 情報センターが管理しないコンピュータの利用について	6.1.4
6.1.5 アクセス制限	6.1.5
第 7 章 文書作成 LATEX の基礎	49
7.1 LATEX の基礎	7.1
7.1.1 LATEX とは	7.1.1
7.1.2 LATEX の処理の流れ	7.1.2
7.1.3 印刷までの手順	7.1.3
7.1.4 ソースファイルの構造	7.1.4
7.2 数式	7.2
7.2.1 2つの数式モード	7.2.1
7.3 簡条書	7.3
7.4 表	7.4
7.5 図	7.5
7.5.1 図のための空白の作り方	7.5.1
7.6 カスタマイズ	7.6
7.7 参考文献	7.7
第 8 章 文書作成 LATEX の参照、作図	54
8.1 相互参照	8.1
8.1.1 章、式、図表等の参照	8.1.1
8.2 labelの使い方	8.2
8.1.2 相互参照を有効にする	8.1.2
8.2.1 グラフィックエディタ	8.2.1
8.2.2 機能概要	8.2.2

8.2.2 <i>tgif</i> の起動	56	11.2.6 互換性	72	第 14 章 簡単なシェルプログラミング	84
8.2.3 起動時のウインドウ	56	11.2.7 PDF	72	14.1 シェルスクリプト	84
8.2.4 作図	56	11.3 Compress	72	14.1.1 対話処理とハッシュ処理	84
8.2.5 例えば四角形を描く	57	11.3.1 アーカイブ	72	14.1.2 シェルスクリプトーハッシュ処理	84
8.2.6 マウスボタンの意味	57	11.4 Backup Method	72	14.1.3 新しいコマンドを作る	84
8.2.7 作図のまとめ	57	11.4.1 リムーバブルメディア	72	14.1.4 コマンドの引数	85
8.2.8 保存	58	11.4.2 オンライнстレージ(Online Storage)	73	14.1.5 プログラムの出力を引数として受け取る	85
8.2.9 印刷	58	11.5 Web Browser Setting	73	14.2 C シェル(csh, tcsh)	86
8.2.10 終了	58	11.6 その他	74	14.2.1 foreach	86
8.3 LATEX への図の取り込み	58	11.6.1 ファイル通信機能 ftp	74	14.2.2 文字列修飾子	86
8.3.1 figure 環境	59	11.6.2 MagicPoint	74	14.2.3 ファイルテスト演算子	86
8.3.2 from dvi to ps	59	11.6.3 MATLAB	74	14.2.4 配列	86
第 9 章 学術論文の作成, スライド	60	11.6.4 Maxima	74		
9.1 学術論文	60	11.6.5 S-plus(統計ソフト)	74		
9.1.1 学術論文とは	60	11.6.6 オフィスツール	74		
9.2 科学技術文章の要件	60				
9.2.1 良い技術文章の必要性	60				
9.2.2 文を書く上で注意	61				
9.3 LATEX の論理構造命令	61				
9.3.1 <i>section</i> コマンド	61				
9.3.2 学術論文の論理構成と LATEX	62				
9.3.3 論理構成に便利な使い方	62				
9.4 文獻の参照	62				
9.4.1 BibTeX を用いた参照の方法	63				
9.4.2 thebibliography	63				
9.5 スライド Beamer	64				
9.6 LaTeX から PDFへ	65				
9.6.1 Type1 フォント	65				
9.6.2 フォントの埋め込み	65				
第 10 章 著作権およびコンピュータやネットワーク利用に関する法律, セキュリティ	66				
10.1 法律	66				
10.1.1 著作権関係の webpage	67				
10.2 Security	67				
10.2.1 セキュリティ維持の為のユーザの義務	68				
第 11 章 情報管理	69				
11.1 File Types	69				
11.2 File Convert	69				
11.2.1 画像ファイル	69				
11.2.2 音声ファイル	70				
11.2.3 メールへの添付ファイルについて	70				
11.2.4 PostScript	71				
11.2.5 EPS	71				
第 13 章 シェル(その 2)	80				
13.1 シェルの対話機能 2 — bashを中心にして	80				
13.1.1履歴(history)	80	1.4 タイピングの練習	92		
13.1.2 別名(alias)	81	14.1 タイピングの基礎知識	92		
13.1.3 クオーティングヒメタキャラクタ	81	14.2 タイピング練習をしてみましょう	92		
13.2 Bash シェルプログラミング	82	1.5 端末エミュレータ	92		
13.2.1 for	82	1.5.1 フォントサイズ	92		
13.2.2 文字列切り出し	82	1.6 課題 1	92		
13.2.3 ファイルテスト演算子	82	1.7 宿題 1	93		
13.2.4 配列	83	1.8 注意 1	93		
13.2.5 \$HOME/litteracy を作る	93	1.9 Unix コマンド(1) ファイルディレクトリ	93		

第 2 章 パスワード管理,X Window System, Emacs (1)	94	4.4 Window Manager	106	7.3 課題 7	117
2.1 パスワードの変更	94	4.5 UNIX(3)-パイプ, リダイレクト	106	8.1 表計算	118
2.1.1 パスワードの変更法	94	4.5.1 練習	106	8.1.1 Awk を使った簡単な足し算	118
2.2 マウスの操作	95	4.6 宿題 4(ターミナルモード)	107	8.1.2 Awk を使った簡単な計算	118
2.3 X ウィンドウシステム	95	4.7 GUI とターミナルモード	107	8.1.3 Awk を使った簡単な成績処理	119
2.3.1 X ウィンドウシステムの利用	95	第 5 章 印刷	108	8.2 課題 8	119
2.3.2 いろいろなウィンドウ、マウスの操作	95	5.1 印刷の方法	108	8.3 表の連結	119
2.3.3 練習	95	5.1.1 テキスト形式ファイルをポストスクリプト形式のファイルに変換する	108	8.4 宿題 8	119
2.4 Emacs(1)	96	5.1.3 PDF 形式のファイルの印刷	109	9.1 情報の入手	120
2.4.1 Emacs の編集モード	96	5.1.4 便利なオプション	109	9.1.1 情報検索	120
2.4.2 Emacs の基本操作	96	5.1.5 便利なツール	109	9.2 文字列検索	120
2.4.3 チュートリアル(Tutorial) の実行	97	5.1.6 ハンドアウトの印刷	110	9.2.1 課題 9	121
2.4.4 課題 2	97	5.1.7 Web Page の印刷(Mozilla Firefox)	110	第 6 章 UNIX の復習	112
2.5 日本語入力 (1)	97	5.1.8 課題 5	111	6.1.1 UNIX の命令の実行の方法	112
2.5.1 Emacs を利用した日本語入力	97	5.2 宿題 5(タイピングの練習)	111	6.1.2 ディレクトリの内容を出力する命令	112
2.5.2 変換方法	97	第 7 章 文字列検索と文字列置換	116	6.1.3 ネットワークやホストの管理情報	112
2.5.4 Sylpheed や Firefox, OpenOffice で日本語を入力する方法	98	7.1 文字列検索	116	6.1.4 ジョブの管理	112
2.6 Unix コマンド(2) アクセス制限管理	98	7.1.1 バイナリファイルを文字列検索する	116	6.1.5 ディレクトリの操作	112
2.6.1 \$HOME/literacy のアクセス制限	98	7.2 文字列置換	116	6.1.6 ファイルの操作	112
第 3 章 ファイル管理, Emacs(2), メール, ニュース	99	7.2.1 Emacs で文字列検索、文字列置換する	116	6.1.7 フィルタ	112
3.1 UNIX ファイル(1)	99	7.2.2 単語リストを作る	117	6.1.8 ネットワークを利用したログイン	113
3.1.1 ファイルに関するコマンド(命令)	99	7.2.3 tr を用いて置換する	117	6.1.9 その他の命令	113
3.1.2 演習	100	7.2.4 その他のツールで文字列置換する	117	6.1.10 ログインと X Window System	113
3.1.3 GUI を利用した File 管理	100	第 8 章 表計算	118	6.1.11 アクセス許可	114
3.2 Emacs(2) - 主な編集機能	100	8.1 表計算	118	6.1.12 chmod の練習	114
3.2.1 ポイント, マーク, リージョン(領域)	100	8.1.1 Awk を使った簡単な足し算	116	6.2 標準入出力とパイプ, リダイレクト	114
3.2.2 リージョンに対する編集	100	8.1.2 Awk を使った簡単な計算	116	6.3 UNIX のファイルシステム	115
3.2.3 バッファとレジスタ	100	8.1.3 Awk を使った簡単な成績処理	119	6.4 課題 6	115
3.2.4 矩形領域の編集	100	第 9 章 情報検索	120	6.5 宿題 6	115
3.2.5 課題 3	101	9.1 情報の入手	120	第 10 章 ネットワークのエチケットや倫理	122
3.3 メールとネットニュース	101	9.1.1 情報検索	120	10.1 Network Etiquette = Netiquette	122
3.3.1 メール・ニュースリーダ	101	9.2 文字列検索	120	10.2 課題 10	124
3.3.2 宿題 3	102	9.2.1 課題 9	121	10.3 宿題 10	124
3.4 LTeX(0) サンプルを使った練習	102	第 11 章 LTeX の基礎	125	第 11 章 LTeX の基礎	125
第 4 章 ディレクトリ, X Window System (2), プロセス管理, Window Manager	103	11.1 LTeX の文章の最小構成	125	11.1 LTeX のコンパイル	125
4.1 UNIX(2) ディレクトリ	103	11.2.1 英文のみの文書	125	11.2.1 LTeX のコンパイル	125
4.2 課題 4	103	11.3 LTeX(1)	125	11.2.2 LTeX の注意	125
4.3 X ウィンドウ (2)	104	11.3.1 LTeX の注意	125	11.3.2 特殊文字	125
4.3.1 X ウィンドウシステムの構成と操作	104	11.3.3 字体 (フォント = font) と文字の大きさ	126	11.3.3 字体 (フォント = font) と文字の大きさ	126
4.3.2 いろいろな X ウィンドウプログラム	105	11.3.4 ソースをそのまま出力する方法	126	11.3.4 ソースをそのまま出力する方法	126
4.3.3 練習	105	11.3.5 左寄せ・右寄せ・センタリング	126	11.3.5 左寄せ・右寄せ・センタリング	126
4.3.4 stty -a	106	11.4 数式モード, 参照, 図の挿入	126	11.4.1 数式を書く場合の注意	128
		11.4.2 Examples	128	11.4.2 Examples	128
		11.5 課題 11	129		
第 12 章 LTeX の練習	130				
7.1 文字列検索	116	12.1 LTeXtyping	130		
7.1.1 バイナリファイルを文字列検索する	116	12.1.1 練習 (1)	130		
7.2 文字列置換	116	12.1.2 練習 (2)	130		
7.2.1 Emacs で文字列検索、文字列置換する	116	12.2 課題 12(itemize)	131		
7.2.2 単語リストを作る	117	12.3 宿題 12(circuit.tex)	131		
7.2.3 tr を用いて置換する	117				
7.2.4 その他のツールで文字列置換する	117				

第 13 章 作図		132	第 21 章 試験: ネットワークを利用する場合のエチケットや倫理、著作権およびコンピュータやネットワーク利用に関する法律		157	
13.1 大学で利用できる主なグラフィックエディタ	132	18.4 民事訴訟(名誉毀損など)に関する事項	143	トや倫理、著作権およびコンピュータやネット		
13.2 gif の使い方 : 補足	132	18.5 犯罪的陳列に関する事項	144	ワーク利用に関する法律		
13.2.1 印刷	132	18.6 商行為に関する事項	144	21.1 競争的課題	157	
13.2.2 表示	132	18.7 個人情報の保護に関する事項	144	22.1 シェル: 計算機利用環境	158	
13.2.3 作図	132	18.8 その他	144	22.1 シェルとは	158	
13.3 絵を貼り込む	132	18.9 u-aizucourse.lit1 に質問しましょう	144	22.2 Shell 環境設定上の注意	158	
13.4 課題 13	133	18.10 課題 18	144	22.2.1 プロンプトを設定しましょう	158	
13.5 宿題 13	134	18.11 Octave の利用	144	22.3 Desktop 環境設定上の注意	158	
第 15 章 L^AT_EX の応用		135	18.11.1 行列計算	145	22.3.1 Gnome Desktop System	159
15.1 L ^A T _E X(Tabular)	135	18.11.2 プロット	145	22.3.2 Mac OS X	159	
15.1.1 複雑な表を書く	135	19.1 グラフを書くツール	147	22.4 作ってみましょう	159	
15.1.2 拡張版表組環境(tabularx)	135	19.2 Gnuplot でグラフを書く	147	22.4.1 初級コース(mytest)	159	
15.2 Tabular の練習	136	19.2.1 gnuplot のスクリプトファイル	147	22.4.2 会話的な処理(Interactive Processing)	159	
15.3 L ^A T _E X(Macro)	136	19.2.2 グラフをファイルに出力する	147	22.5 バッヂ処理	160	
15.3.1 マクロ	136	19.2.3 データのグラフをファイルに出力する	147	22.5.1 コマンドライン引数を利用する例	160	
15.4 課題 15	137	19.2.4 グラフを飾る	149	22.6 課題 22	161	
15.5 宿題 15	137	19.2.5 横軸を時間にする	149	22.7 宿題 22	161	
第 16 章 L^AT_EX スライド作成		138	19.2.6 対数スケール	149	22.3.2 Mac OS X	159
16.1 Beamer	138	19.3 課題 19	149	22.4 作ってみましょう	159	
16.1.1 サンプル 1 基本構造	138	19.4 Xmgrace の利用	149	22.4.1 初級コース(mytest)	159	
16.1.2 サンプル 2 タイトルページ	138	19.5 Mathematica の利用	150	22.4.2 会話的な処理(Interactive Processing)	159	
16.1.3 サンプル 3 箇条書き	138	19.5.1 グラフを書く	150	22.5 バッヂ処理	160	
16.1.4 サンプル 4 数式	139	19.5.2 描いたグラフを画像ファイルに保存する	150	22.5.1 コマンドライン引数を利用する例	160	
16.1.5 PDF を利用したプレゼンテーション	139	19.6 画像	150	22.6 課題 22	161	
16.1.6 Hyperref	139	19.6.1 表示	150	22.7 宿題 22	161	
16.2 課題 16	139	19.6.2 取込(Grab)	151	22.3.2 Mac OS X	159	
第 17 章 MH^APSTricks、アニメーション		140	19.6.3 画像を Desktop に貼り付ける	151	22.4 作ってみましょう	159
17.1 PSTricks	140	19.7 レポート作成	151	22.4.1 初級コース(mytest)	159	
17.1.1 サンプル 1 図形	140	19.7.1 宿題 19	151	22.4.2 会話的な処理(Interactive Processing)	159	
17.1.2 サンプル 2 グラフ	140	20.1 Compress ファイル圧縮	153	22.5 バッヂ処理	160	
17.1.3 サンプル 3 動く図形	140	20.2 課題 20	154	22.5.1 コマンドライン引数を利用する例	160	
17.1.4 サンプル 4 動く文字	141	20.3 ウェブブラウザ(Firefox)の利用法	154	22.6 課題 22	161	
17.2 アニメーション	141	20.3.1 Application	154	22.7 宿題 22	161	
17.2.1 Mathematica を利用する	141	20.3.2 Advanced タブ - Cache	154	22.3.2 Mac OS X	159	
17.2.2 Loop	142	20.3.3 lock ファイル	155	22.4 作ってみましょう	159	
17.2.3 パラノパラアニメを作る	142	20.4 FTP(file transfer program/file transfer protocol)	155	22.4.1 初級コース(mytest)	159	
17.3 課題 17	142	20.4.1 FTP の練習	155	22.4.2 会話的な処理(Interactive Processing)	159	
第 18 章 著作権およびコンピュータやネットワーク利用に関する法律		143	20.4.2 SFTP	155	22.5 バッヂ処理	160
18.1 Copyright 著作権などの法律	143	20.4.3 匿名 FTP	156	22.5.1 コマンドライン引数を利用する例	160	
18.2 著作権に関する事項	143			22.5.2 会話的な処理(Interactive Processing)	160	
18.3 肖像権に関する事項	143			22.6 課題 22	161	

25.1.7 組織のコンピュータやネットワークを守る	168
25.2 課題 25	168
25.3 宿題 25	168
第 26 章 ネットワークコンピューティング、プロセス管理、インターネット	169
26.1 Network の利用法	169
26.1.1 分散化されている計算機資源	169
26.1.2 異なる計算機資源	169
26.1.3 更新	169
26.1.4 NFS	170
26.1.5 リモートでの利用	170
26.1.6 ネットワークを監視するためのツール	171
26.2 プロセス管理	171
26.3 インターネット利用	172
26.3.1 注意事項	172
26.3.2 ウェブブラウザの利用	172
26.3.3 会津大学共用 WWW サーバの利用	173
26.4 課題 26	173
26.5 宿題 26	173
第 27 章 シェルプログラミング (2), 表計算 (OpenOffice Spreadsheet)	175
27.1 シェルの便利な使い方 – その 2	175
27.2 前回の復習	175
27.3 シェルプログラミング	175
27.3.1 作ってみましょう	175
27.3.2 チャレンジコース	175
27.4 課題 27	176
27.5 OpenOffice	176
27.5.1 OpenDocument Format	176
27.6 宿題 27	176
第 28 章 Make, スペルチェック, プрезентーション (OpenOffice Presentation), MagicPoint	177
28.1 make	177
28.2 課題 28 GNU-HELLO	177
28.3 Spell Check	177
28.3.1 spell	177
28.3.2 aspell	177
28.3.3 Enacs で ispell	178
28.3.4 Mac の辞書	178
28.4 プrezentーション	178
28.4.1 soffice の起動	178
28.4.2 presentation 利用開始	178
28.4.3 作業モード	178
28.4.4 プrezentーションの公開	179

28.5	MagicPoint	284.5	スライドショウ
28.6	宿題 28	179	
29	トラブル対処法	180	
29.1	トラブル対処法	180	
29.1.1	キーボードで入力ができない	180	
29.1.2	画面がおかしい	180	
29.1.3	ネットワークの具合が悪い	180	
29.1.4	ファイルを見ることができない	180	
29.1.5	印刷が暴走している	180	
29.1.6	コマンドが動作しない	180	
29.1.7	ファイルを誤って消してしまったら	180	
29.1.8	標準環境設定ファイル	181	
29.1.9	ISTC FAQ	181	
29.2	トラブルレポートの方法	181	
29.2.1	SSB(System support Base)	181	
29.2.2	緊急の場合	181	
29.3	課題 29	182	
29.4	HTML (Web ページを作る)	182	
29.4.1	HTML の書き方	182	
29.4.2	HTML の練習	182	
29.5	宿題 29(web-int)	182	
29.6	Maxima の利用	182	
29.6.1	多項式	182	
29.6.2	行列計算	183	
29.6.3	プロット	183	
30	E-mail, Calculator, その他の便利な道具	184	
30.1	電子メール (email)	184	
30.1.1	マナー	184	
30.1.2	再送	184	
30.1.3	添付	184	
30.1.4	添付ファイル	184	
30.2	課題 30	185	
30.3	電車	185	
30.4	便利な道具	186	
A	Sylpheed の設定	187	
A.1	はじめての利用	187	
A.2	メールアカウントの追加	187	
A.3	ニュースアカウントの追加	187	
A.4	迷惑メール対策	187	
B	CUI ベースのメール/ニュースアプリ	188	
B.1	/bin/mail, /usr/bin/mailx の使い方	188	
B.2	Emacs を使用したメール/ニュース	188	
B.3	その他のツール	188	

付録 C	タイピング練習 C 言語 example
C.1	100以下の自然数の和
C.2	100以下の正の偶数の和
C.3	$1+2+3+\dots+n \geq 500$ を満たすnとそのときの和
C.4	$1+2+3+\dots+n < 500$ を満たすときの和
C.5	500 400 300 200 100 0と表示する
C.6	$1+2+3+\dots+10$ と $1+2^2+\dots+10^2$
C.7	複素数 $e^{i\frac{\pi}{3}}$ の n乗の表示
C.8	フル型の利用
付録 D	演習参考資料 L ^A T _E X sample
D.1	Latex の基本例
D.2	フォントの例
D.3	章立ての例
D.4	箇条書きの例
D.5	作表の例
D.6	図の例
D.7	数式の例
D.8	グラフの例
D.9	文献参照の例
付録 E	演習参考資料 T _{GIF} Manual
E.1	四角形を描く.
E.2	円を描く.
E.3	文字を入れる.
E.4	円の線を太くし, 中を塗る.
E.5	矢印を描く.
E.6	矢印の線を太くし, ペンシルマークにする.
E.7	四角形をりサイズする.
E.8	円を移動する.
E.9	文字のフォントを変える.
E.10	その他
E.11	まとめ
付録 F	AINS の利用
F.1	大学の標準環境の利用方法
F.1.1	研究室の端末や個人の端末から利用する。
F.1.2	学外から利用する。
F.2	学内での個人PCのネットワーク
付録 G	自分のPCで自分のUNIX環境
G.1	大学のWSでWindowsやLinux
付録 H	AINS 利用規定

目次

5.5 Mozilla Firefox のファイルへ印刷	111
6.1 gdm login	113
6.2 Select Desktop	114
6.3 Pipe を繋ぐ	114
11.1 サブウインドウの名前	128
11.2 サブウインドウの名前	128
13.1 tgif import	133
17.1 pstricks0	140
17.2 pstricks-plot	141
18.1 $\sin(x)$ 関数	145
18.2 $z=\sin(r)r, r=\sqrt{x^2+y^2}$ の 3 次元プロット	146
18.3 $z=\sin(x)*\cos(y)$ の 3 次元プロット	146
18.4 $(x, y, z) = (t * \sin(t), t * \cos(t), t)$ の 3 次元プロット	146
19.1 Xmgrace	150
19.2 $\sin(x)$ 関数	150
19.3 3D プロット	150
19.4 Grab	151
20.1 Mozilla Firefox: Opening pdf	154
20.2 Firefox の Applications の確認、変更	154
20.3 キャッシュの設定	155
23.1 ヒストグラム	164
29.1 ISTC FAQ	181
29.2 $\sin(x)$ 関数	183
29.3 $z=\sin(x)*\cos(y)$ の 3 次元プロット	183
E.1 Mode メニュー	195
E.2 四角形を描く	195
E.3 文字を描く	196
E.4 円の線を太くする	196
E.5 円の中をぬる	196
E.6 矢印を描く	197
E.7 矢印を太くし、塗る	197
E.8 オブジェクトをリサイズする	197
E.9 オブジェクトを移動する	197
E.10 文字のフォントを変える	198
E.11 スプライン化	198
E.12 デフォルトの変更	198
1.1 ファイルの木構造(ディレクトリツリー)の概念図	11
3.1 サーバピクライアント	28
3.2 認証ネットワーク	33
4.1 漢字変換の動作	35
4.2 iiimscrt	36
4.3 gnome-terminal-set-character-encoding	37
4.4 gnome-terminal-select-input-methods	38
4.5 gnome-terminal-nihongo	38
5.1 図書館資料のリクエスト方法	44
5.2 国立国会図書館の電子図書館	44
5.3 arxiv (Grisha Perelman)	44
6.1 NAT の責任の範囲	48
8.1 図の参照の例	55
8.2 サブウインドウの名前	56
8.3 Choice ウィンドウ	57
8.4 メインメニュー	57
1.1 login screen	90
1.2 gnome desktop / mac os x logout	91
1.3 主に使用するキー	91
1.4 制御キー	91
1.5 指の割り当て	92
2.1 Unicorn ID Manager	94
4.1 ウィンドウの構成	104
4.2 X ウィンドウの画面構成例(Gnome Desktop)	105
5.1 Acrobat Reader の印刷オプション設定	109
5.2 Mozilla Firefox のページ設定	110
5.3 Mozilla Firefox の印刷器	110
5.4 Mozilla Firefox のページ設定	111

表 目 次

1.1 ディレクトリに関するコマンド. 各コマンドの厳密な文法や, 機能の詳細は, オンライスマニュアルや各種参考書を参照して下さい.	12
2.1 コマンド例 (1). 各コマンドの厳密な文法や, 機能の詳細は, オンライスマニュアルや Info を参照して下さい.	24
2.2 フィルタ. 各コマンドの厳密な文法や, 機能の詳細は, オンライスマニュアルを参照して下さい.	25
4.1 正規表現	37
6.1 1995 年までの各国の政府レベルの検討	46
6.2 最近の条約	46
6.3 日本の法律	47
8.1 表の参照の例	55
8.2 マウスボタンの意味	57
12.1 リダイレクトヒュイブ	78
11.1 特殊文字 1	125
11.2 特殊文字 2	126
11.3 アクセント記号	126
26.1 プロセスの属性	172

第1章 コンピュータリテラシーとは何か、ディレクトリとアクセス制御

会津大学では UNIX ベースの情報処理教育を行いますが、だからといって Windows を初めてとした Microsoft 製品の利用を制限しているわけではありません。むしろ、研究室では多くの Windows パソコンや Mac や Linux など多種多様なコンピュータを使用しています。また会津大学では Microsoft Open Value Subscription-Education Solutions Agreement(OVS-ES)という契約を Microsoft 社と契約していますので、大学の売店で格安で Office を購入し、各自所有のパソコンで利用できます。
<http://web-int.u-aizu.ac.jp/labs/istc/ipc/ssb/msca.html>
 に詳しい説明があるので参考にしてください。

1.1 コンピュータリテラシーとは

一般にはコンピュータによる「読み書きソロバン」をコンピュータリテラシーと呼びます。パソコンでワープロや表計算、簡易データベースなどのソフトを使いこなすことが現在のところその中身と考えられています。しかし、コンピュータ理工学部という看板を掲げている本学において、ワークステーションのネットワークシステムを使った新入生向け導入教育ではコンピュータリテラシーを次のように再定義しています。

研究、開発、事務処理などのように大量の情報を処理し文書(画像、音声も含む)を作成するという分野で働く職業人にとって、コンピュータネットワークとそれにつながれたワークステーションは標準の仕事環境となりつつあります。この環境下で自在に仕事をする能力をコンピュータリテラシーと呼びます。

特に重要なのは、自分で、解決の方法を見出す力です。そのためには、いろいろの道具が用意されています。それらの道具をいかにうまく使いこなすかが、重要です。

ワークステーションとコンピュータネットワークの発達で知的生産現場の環境はがらりと変わりつつあります。我々はネットワークを通じて最新の情報にアクセスし、また、そこから自分の研究成果を最新の情報として提供します。プロフェッショナル・コミュニケーション(学会、業界団体)の情報交換は電子メールによるものが主流になってきました。ワークステーションはこの情報ネットワークへの入口です。

- ・インターネットを利用して論文検索／論文投稿の例を探してみましょう。

また同時に、その言葉自身の意味からして当然なのですが、ワークステーションは情報生産・情報加工、研究開発、論文作成のツール群が完備した知的生産の工房でもあります。知識的生産の工房であり、情報ネットワークへの入口、それがワークステーションです。したがって、コンピュータリテラシーとはワークステーションを使いこなす能力のことである、と単純に定義することもできます。

1.1.1 オフィスツール

コンピュータと言えばパソコン、パソコンの使い方はすなわち、ワード、エクセル、パワーポイント等のオフィスツールの使い方と考える人が一般にはとても多いようです。しかし、これはとても狭い考え方です。リテラシーではこれらの使用法を特別に説明することはできません。Mac OS X 端末では Microsoft 社の Office が利用できるので、利用してください。Solaris 端末では OpenOffice が利用できます。

コンピュータを利用する活動	要求される能力や経験など	リテラシー (A1B) でカバーする範囲
(研究開発活動) の開発: プログラム開発 -ソフトウェアの開発 -論理回路設計 -基礎物理研究 -社会自然工学などのモダリング -による研究 -シミュレーションプログラム開発 -データ統計処理	プログラミング言語、アルゴリズム 論理回路 線型代数、微積分学、物理学 確率、統計学	
(研究開発活動) の実施する活動 -プロジェクト開発 -学会報告 -雑誌報告 -文書管理 -研究者同士の連絡 -研究結果の公開 -グラフや図形を書く -数学の問題を解決する	テキストエディタ利用 マルチメディアプレゼンテーション 報告書作成 ファイル管理 電子メール World Wide Web Plotter, Drawer 数式処理ソフト	I (Emacs) I (L ^A T _E X, 印刷) I (Unix File System) I (Email) I (Web Browser) II (HTML) I (Mathematica,Xmgr,Tk) II (Mathematica)
(ビジネス活動) -情報システム管理 -データベース導入とデータ管理 -データ統計処理を用いた的確な経営 -企業情報発信 -World Wide Web を軸とした電子商取引引き -社内情報システム-電子メール、電子掲示版の活用 -社内報告書の提出	データベースシステム 確率・統計学 Java, Perl 等を利用した Interactive なホームページ 電子メール、電子掲示版 ワードプロセッサ利用	II (Database) II (Perl) I (Email, News)
(市民生活) -情報収集-World Wide Web の利用 -情報発信-World Wide Web の利用 -通信-電子メール、電子掲示版の活用	World Wide Web Java, Perl 等を利用した Interactive なホームページ 電子メール、電子掲示版	I (Web Browser) II (Perl) I (Email, News)
(全てに共通する活動) -利用する計算機システムの理解と活用 -ネットワークや計算機を始めとする公共資源を正しく扱うこと -個人情報を公開しないこと -著作権などの法律を守ること	I (Unix, Shell) I (Network Etiquette) I (Network Etiquette) I (Copyright and Laws)	

1.2 リテラシーを構成する3つの技能

コンピュータリテラシーで獲得すべき技能を簡単に説明します。

テキストファイルの作成 文書の清書、印刷 コミュニケーション	Emacsなど TeX, HTML E-mail, NetNews, Web
--------------------------------------	--

1.2.1 テキストファイルの作成

テキストファイルの作成は情報の電子的生成です。社会を飛び交う情報、とりわけ学術的あるいは技術的な情報はテキスト(文字の列)の形で生成され、流通し、蓄積されるのが普通です。そして情報の流通というとき、それは電子的な形をとるのがもっとも効率的です。また、情報の蓄積は情報の再利用を目的としますから再び流通の場にのせられることを想定して行なわれます。したがって学術的情報あるいは技術情報は電子的に生成されなければ有効な情報とはなりません。電子的に生成された情報はテキストファイルという形で個人所有の記憶装置やインターネットで提供される記憶サービスなどに保存されることになります。

このテキストファイルを作成するのに必要な道具がテキストエディタと呼ばれるプログラムです。コンピュータリテラシーIでは代表的なエディタであるEmacsを取り上げます。

- 情報とは何でしょうか？

情報という場合には文学や新聞記事などの文章以外にも、画像や、音声、そしてさまざまなアプリケーションプログラムを含みます。C言語を始めとする多くのプログラミング言語を使用してソフトウェアを開発する場合にも最初は必ずテキストファイルとして作成されます。

1.2.2 文書の清書、印刷

電子的に生成された情報を紙の上に美しくレイアウトすることを文書の清書といいます。この作業はテキストエディタの役目とは区別して考えるのが普通です。自分の考えをテキストの形にするとときテキスト全体の論理構造が当然重要になります。テキストが印刷物の形(プレゼンテーション用の生成物)になるときどのようなレイアウトをしたらよいかも重要なことがあります。この2つが明らかに区別し、前者はテキストエディタが作業を助け、後者はテキストフォーマッタが支援します。

理工系の論文やレポートなどではTeX(D. Knuthによる)というプログラムを使って清書を行なうのが標準になっています。そのTeXを(一般事務文書まで含めて)さらに使いやすくしたLaTeX(L. Lamport)をコンピュータリテラシーIでは取り上げます。

- なぜワープロを利用しないのでしょうか？

ワープロは初心者が見た場合に「やさしそうだな」と感じさせるところに開発の力が注がれています。しかしながらその実、1冊の本を著作するなど大量の文章を作成する場合、論理構成を重視した論文を書く場合などのように専門的な作業には向いていません。また、それぞれのワープロソフトでは文書の保存形式が共通ではないので、他のワープロの文書に変換する場合にはテキスト形式に一旦移して、その後ワープロに固有の面倒なレイアウトの調整や文字フォントの選択をすることになり

ます。

TeXを始め、World Wide Webで利用されているHTMLなどはテキスト形式で文書中にマークを入れて整形するもので、一般的には、Standard Generalized Markup Language=SGMLと呼ばれるものの類形です。これらの文書はテキスト形式で作業ができるのでどのエディタでもどのワープロでも編集できます。TeXは論文雑誌に投稿する場合に指定される形式であったり、また、会津大学では卒業論文をTeXで書くことが規定されています。HTMLはよく知られているようにWebで使用されています。

以上の理由からワープロを利用しないのです。

1.2.3 コミュニケーション

文書の清書までできたら次は情報の発信です。コミュニケーションの手段として電子メールとニュースシステムの利用方法を学びます。会津大学はほとんどすべての利用者がネットワークを利用して学内外の利用者とコミュニケーションすることができます。さらにまた学内について言うならば実際に顔を突き合わせてコミュニケーションすることができます。従来の社会と比較してもコミュニケーションの密度が濃く、速度が速いことができます。これはひとつとしたら、近い将来に実現するネットワークが社会基盤として充実した社会(ネットワーク社会)に似ているかも知れません。その様なNetNews, E-mail, Webを通常の活動とする環境の中で

1. Networkを利用して得られる情報をどの程度利用するか？
2. Networkのコミュニケーションで通常の人間性を保てるか？

など基本的なものを始め、見当もつかない問題が現れてくる可能性もあります。

1.2.4 その他

自分の蓄積したデータファイルを系統立てて管理する方法を学びます。

1.3 プロフェッショナルとしての倫理

本学に入学された学生諸君はコンピュータサイエンスそしてコンピュータエンジニアリングの分野のプロフェッショナルコミュニケーションの入口に立っています。コンピュータリテラシーを技能の側面から定義すれば上に述べた通りですが、このコンピュータリテラシーIの講義と演習では、単に技能的なことがらだけでなく、プロフェッショナルコミュニケーターの一人としての職業倫理を理解し、身につけてもらうことを大きな目標としています。プロフェッショナルとなるために必須です。

1. コンピュータサイエンス
2. コンピュータエンジニアリング

3. プロフェッショナルコミュニケーション

4. 職業倫理

皆さんには今後これらのことがらについて、先生方や専門家の考え方を知り、自分の考えを作りあげていかなければなりません。

また、システムにはセキュリティというものがあつてだれもが本学のコンピュータシステムを自由に使える訳ではありません。ワークステーションの前に座ってシステムに入るごとに、これをログインと言いますが、ログインするときに正規のユーザーであることを確認する手書きがあります。そのときユーザー本人しか知らないパスワードが使われますので英字、数字の混じった8文字程度のものを考えてください。パスワードは他人に知られてはならないものです。パスワードさえ解れば悪意を持った人は学内外を問わず、その人になりますまして、悪いことをします。

1. ファイルを勝手に編集する。

2. 他のユーザに被害を与えるような罠を仕掛ける。

3. システム管理者のパスワードを解析してシステムを乗っ取る。

つまり個人的な問題だけでは收まりません。学内の利用者全体に大きな影響があります。会津大学は全世界からアクセス可能ですから、世界中にある悪意をもつた人たちに、パスワードを盗まれる可能性があります。そうしてシステムに不正侵入しようとすると人たちをクラッカーと呼びます。大学内のどの計算機のパスワードでも知られてしまうことは非常に危険です。

簡単な文字列のパスワードでは簡単に破られるプログラムをクラッカータちは開発しています。情報センターでは利用者のパスワードを定期的に調査しています。変更の指示が情報センターからあった場合には速やかに従って下さい。
さて、では皆さんはどのように対処しますか？

1. 偶然他人のパスワードを知ってしまった。

2. 偶然システム管理者のパスワードを偶然見つけた。

3. 個人のプライバシーに関するファイルを偶然見つけた。

4. 期末試験のファイルを偶然見つけた。

5. パスワードクラックプログラムを手に入れた。

6. セキュリティに問題のある仕掛けの方法を手に入れた。

1.4 第1回演習予告

第1回の演習では全員がコンピュータに初めて触れるという前提で基本的な操作法を勉強します。パソコンなどを使いなれている人にとっては少し退屈な進度になるかも知れませんが、余裕のある人は周りを見てうまく行かない友達を助けてください。きっといるはずですよ。他人に説明することは、自分自身がより深く物事を理解する助けにもなるでしょう。

1.5 第1回レポート宿題

1. コンピュータの使用経験

2. コンピュータを使って何をしてみたいか

3. どのようなコンピュータを作つてみたいか

4. なぜコンピュータを研究したいと思ったのか

以上についてA4サイズのレポート用紙2枚程度にまとめて次週の演習のときに提出してください。手書きでかまいません。レポート用紙はホッチキス等でとじてください。バラバラのままでは困ります。文章能力をみようということですから文章の組み立てをよく考えて取り掛かってください。

基礎的な文章力が不足していると、まともな文章を書くことは困難です。大学生の今からでも十分間に合うので、活字を沢山読んで文章に慣れ親しむ努力をしましょう。大学の教育では学生の皆さん的基本的な国語力を前提とします。

1.6 Users

コンピュータは様々な装置で構成されていますが、それだけでは動作しません。装置以外では、コンピュータで動作させるプログラム、プログラムや文書を記録したファイル、そして、プログラムを動作させるユーザーが必要です。ユーザーは管理者ユーザーと一般ユーザーの2つに大きく分けられます。

UNIXの場合、rootが管理者ユーザーであり、スーパーユーザとも呼ばれます。/etc/passwdファイルを眺めるとrootの他に様々なユーザーがあります。

rootは特権的な能力をえらっています。ファイルの所有者を変更したり、ユーザーの追加や削除など、管理業務に必要な操作ができます。アプリケーションプログラム等を/usr/の様なrootが所有するディレクトリ内にインストールするのはroot権限が無ければできません。そこで、我々が個人で所有するパソコンである各種LinuxやMac OS Xでソフトウェアのインストールの操作においてはsudoコマンドを利用してrootの特権を使用して操作することができます。sudoは登録された管理者のみが実効出来ます。大学の環境では一般的なユーザーは管理者ではないので、sudoの実行は許可していません。

WindowsPCの場合はrootではなく、Administratorが特権ユーザーです。
一般的ユーザーは学生や教員などのユーザーです。ホームディレクトリと呼ばれる領域が与えられます。その中にファイルを作ることができます。ホームディレクトリは/etc/passwdの第6フィールドに設定されます。大学のネットワーク(ANS)で管理されている情報はgetent passwdで表示できます。

1.7 ファイル

1.7.1 ディレクトリ(フォルダ)

UNIXでは、いくつかのファイルを一まとめにして、そのファイルのまとまりに名前をつけて扱います。このファイルのまとまりをディレクトリと呼びます。ちょうど、書類をまとめ

たファイルを、分類して箱に入れるようなものです。つまり、データの固まりであるファイルを、分類してディレクトリという箱に入れるということです。

UNIXではディレクトリもまた一つのファイルとして扱われます。ですから、ディレクトリの中にまたディレクトリを入れることができます。このディレクトリの構造を表すのに、図1.1のような木構造(ディレクトリツリー)の概念をよく用います。図1.1では、解り易い様にディレクトリ名には大文字を、ファイル名には小文字を使っていますが、ディレクトリ名、ファイル名に大文字小文字どちらを使っても構いません。

大元(おおもじ)のディレクトリ(=/)は木の根に似ていますので「ルート(ディレクトリ)」という特別な名前がついています。UNIXではこの仕組みを変更することはありません。そのディレクトリは一部の例外を除いて原則的にユーザによって勝手に命名することができます。

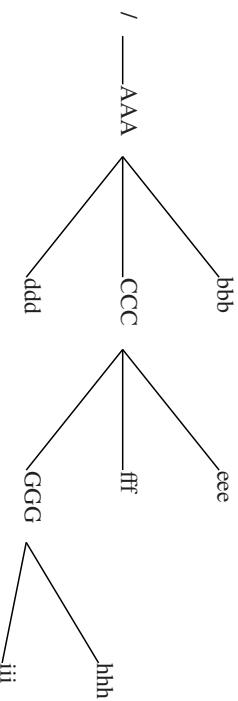


図1.1. ファイルの木構造(ディレクトリツリー)の概念図

1.7.1.1 カレントディレクトリ

皆さん方がloginした直後にカレントディレクトリの内容を表示する命令lsを詳細表示のオプション(スイッチ)lを付けて以下の様に

ls -alF

を実行すると、.bashrcやMailなどいくつかのファイルやディレクトリがリストされます。それぞれの行がファイルやディレクトリを表します。各行の空白で区切られた第1フィールドはアクセス制限の状態、第2フィールドはリンク数、第3フィールドは所有者ユーザ名、第4フィールドはグループ名、第5フィールドはファイルサイズ、第6フィールドは最終更新日、第7フィールドは最終更新年または時刻、最終フィールドはファイル名が表示されます。

ホームディレクトリの内容を表示

```
stdrsv1121[{s1XXXXXX}]: ls -alF |more
total 4678
drwxr-xr-x 25 s1XXXXXX student 4608 Apr 16 18:36 .
drwxr-xr-x 3 root root 512 Apr 16 18:04 ..
-rw-r--r-- 1 s1XXXXXX student 3596 Oct 14 2004 .bashrc
-rw-r--r-- 1 s1XXXXXX student 2357 May 3 2004 .emacs
...
drwx----- 4 s1XXXXXX student 3072 Apr 16 19:02 Mail/...
```

この様にこれらが置いてあるディレクトリを、皆さん方のホームディレクトリといいます。ホームディレクトリは、ユーザ名一つに対し一つずつ与えられます。ではホームディレクトリはルートディレクトリ(/)からはどの位置にあるのでしょうか?それはloginした直後に

pwd

という命令を実行することで確認できます。

現時点におけるユーザーの作業ディレクトリのことをカレントディレクトリと呼び、pwdコマンドでいつでも確かめることができます。

ls命令を引数なしで実行したときにはカレントディレクトリにあるファイルやディレクトリを表示することになっています。カレントディレクトリを変更するためにはカレントディレクトリを変更するためには

cd ディレクトリ名

を実行します。これを「ディレクトリを移動する」と表現することもあります。ディレクトリ名を省略するとホームディレクトリに移動します。

1.7.1.2 相対パスと絶対パス

ファイル名を指定するには、3通りの方法があります。

1. カレントディレクトリに存在するファイルは、そのファイル名だけで指定できます。
2. カレントディレクトリに存在するディレクトリの中に存在するファイルは、「ディレクトリ名/ファイル名」の形で表すことができます。
3. おおもじのディレクトリのことをルートディレクトリと呼び、「/」で表します。図1.1では、一番左にあるディレクトリです。すべてのファイルは、このルートディレクトリから順々にディレクトリをたどって指定できます。

ディレクトリ名つきのファイル名のことを、パス名(または単にパス)といいます。2のよう にカレントディレクトリから指定するパスのことを相対パス、3.のようにルートディレクトリから指定するパスのことを絶対パスといいます。例えば、図1.1でカレントディレクトリがCCCだったとします。hhhといふファイルを指定するには、相対パスではGGG/hhh、絶対パスでは/AAA/CCC/GGG/hhhとなります。今後、単にパス名といったときは、相対パス/絶対パスどちらでもよいこととします。

1.7.1.3 特別のディレクトリを表す記号

以下の3つの記号は、ディレクトリを表す記号として特別の意味を持ちます。

~/ ホームディレクトリを表します。**cd ~/** を試して移動の結果を確認しましょう。

. カレントディレクトリを表します。**cd .** を試して移動の結果を確認しましょう。

.. カレントディレクトリの属している、すぐ上のディレクトリを表します。親ディレクトリとも呼びます。**cd ..** を試して移動の結果を確認しましょう。

ホームディレクトリを指定する方法として、\$HOMEも利用可能です。これは環境変数 \$HOMEに予めホームディレクトリを設定してあるからです。実際に

```
echo $HOME
```

を実行して変数の値を表示するとその内容がわかります。Solaris 端末では学生のホームディレクトリは

```
/home/student/s125XXXX
```

です。Mac OS X 端末では\$HOME の内容は少し異なっています。注意が必要です。

```
/Network/Servers/profsv2/vol/vol1/home1/a-fujitu
```

また、

```
~user
```

はユーザー(user)のホームディレクトリを表す表記として使用できます。実際に

```
ls ~a-fujitu/
```

を実行するとユーザー a-fujitu のホームディレクトリのファイルやディレクトリのリストを参照することができます。

1.7.1.4 ディレクトリ関連コマンド

ディレクトリに関するコマンドを、表 1.1 に示します。

1.7.2 オーナーとパーセッション

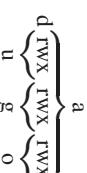
各々のファイルには、全て持ち主(オーナー)が決まっています。オーナーは自分のファイルに対して、書き込んでもよい、中身を読んでも(コピーしても)よいか、(実行可能なファイルなら)プログラムとして実行してもよいか、についての許可(パーミッション)を各ファイルに設定できます。

命令	機能	文法	説明
cd	カレントディレクトリの変更	cd path	path のディレクトリにカレントディレクトリを変更する
pwd	カレントディレクトリの表示	pwd	現在のカレントディレクトリの絶対パスを表示する
mkdir	ディレクトリの作成	mkdir path	path というディレクトリを新たに作成する
rmdir	ディレクトリの削除	rmdir path	path というディレクトリを削除する

表 1.1. ディレクトリに関するコマンド. 各コマンドの厳密な文法や、機能の詳細は、オンラインマニュアルや各種参考書を参照してください。

パーミッションの状態を表示するためには ls -l の様に -l というオプションスイッチをつけて実行します。

表示は以下のように左からユーザ(user)に対する許可、グループ(group)に対する許可、その他(other)に対する許可が表示されます。



```
chmod mode filename
```

mode, *filename* で指定されるファイルに, *mode* で指定される許可を設定します。

mode は、次の形式をしています。

who op permission [op permission]...

mode の例をいくつか書いてみます。

o-rw 他人が読み書きでないようにする。
a+rwx 全ての人に対し全ての許可を与える。
o-r-w オーナー以外に対して、読み出し許可を与えて、書き込み許可は与えない。

以下の表は *mode* に現れる文字や記号の説明です。

<i>who</i> に指定するもの	<i>op</i> に指定するもの	<i>permission</i> に指定するもの
u ユーザ自身.	+ 許可を与えること.	r 読み出し許可. (Readable)
g 同じグループのユーザ.	- 許可を与えないこと.	w 書き込み許可. (Writable)
o その他の人.		x 実行許可. (Executable)
augo 全て.		(意味)

1.8 参考資料

コンピュータリテラシーで学ぶ内容に関係する参考資料

- たのしいUNIX UNIXへの招待, 坂本文著, アスキー出版局
- 続たのしいUNIX シェルへの招待, 坂本文著, アスキー出版局
- 新 The UNIX Super Text(上), 山口和紀, 古瀬一隆監修, 技術評論社
- モーレツ UNIX 教室, UNIX MAGAZINE 1996/6 - 1997/7

- コンピュータに関する雑誌は商品の紹介が大部分であったりするが、中には新しい技術が紹介されていたりするので興味を持つて記事を読んで欲しい。
- コンピュータに関するニュース f1.comp., f1.os., f1.sys., f1.unix などに投稿される記事の中には便利なものがある。全てを読むことは不可能なので、定期的に記事を眺めることを勧める。
- ウェブで提供される情報を検索することも役に立つ場合がある。Googleなどの検索を利用すると良い。

1.8.2 ネットワークとセキュリティ

- ホームページにおける著作権問題, インターネット弁護士協議会(ILC)編著、毎日コミュニケーションズ
- Practical UNIX Security, S.Garfinkel and G.Spafford 共著、アスキー出版局
- Computer Security Basics, D.Russell and G.T.Gangemi Sr. 共著、アスキー出版局
- Building Internet Firewalls, D.B.Chapman and E.D.Zwickly 共著、オライリージャパン

1.8.3 Emacs

- 日本語GNU-EMACS(Mule)入門編(チュートリアル)
- 入門 Mule, 大木敦雄著 アスキーオ出版局
- GNU Emacs 完全ガイド上, シュノーバー, M. A. 著/小畠喜一訳 トッパン
- UNIXへの招待 Mule, UNIX MAGAZINE 1996/9 - 1997/3

1.8.4 コンピュータの基礎、一般的な知識

コンピュータリテラシーはコンピュータサイエンスの全てではない。コンピュータの基礎、一般的な知識についてはコンピュータリテラシーの担当する内容では無いが身につけることが望ましい。

- 情報工学概論, 三井田惇郎著, 森北出版
- 情報科学基礎編, 中易秀敏, 岡野博宣共著, 共立出版
- コンピュータ概論－情報システム入門, 石原秀男他、共著, 共立出版

1.9 Quiz

以下の問題の中から問題を選択し、回答を u-aizu.course.list にレポートしなさい。

- 評価は 10 点満点。
 - 先着一名のみ採点される。
 - 一人一題まで解答できる。ただし、別解であれば可とする。
 - 複数のクイズに解答し、他の学生の権利を奪った場合には、評価は 0 とする。
 - ニュースへ投稿したレポートで評価対象のものは削除してはならない。
 - 提出の〆切は最後の演習が行われる日の 20 時とする。
 - 完結したレポートにするために、回答の本文には必ず、問題を記載し、その後で回答を記述すること。
 - ニュースへの投稿のサブジエクトは指示に従うこと。
- 以上のルールに従って採点をする。ルールに従ったレポートのみが採点対象である。

クイズ 1

ニュースの Subject には必ず [Quiz1-X] (X はクイズの番号) を書き込むこと。

1. ls を実行すると、


```
aaa bbb
となるが、ファイルの中身を見ようとすると、
% cat aaa
cat: aaa: No such file or directory
となる。どのような理由が考えられるか?
```
2.


```
ls -l ~username
```

 と


```
ls -l ~username/
```

 の違いを解説せよ。
3. chmod をつかって、あるディレクトリにある全てのファイルのアクセスモード(ペーミッション)を変えるにはどうすれば良いか?

4. rmdir を実行してもディレクトリが消せないときがある。それはどんなときか? またどうすれば消せるようになるか?
5. finger コマンドを使って、隣のマシンに誰がログインしているか調べる方法を説明せよ。
6. ps コマンドを用いてワークステーションで動作しているすべてのプロセスのリストを得るにはどうしたら良いか?
7. 前述の問によって得られたリストからユーザが実行しているプロセスのみのリストを得る方法を示せ。
8. kill コマンドはユーザが実行しているプロセスを終了させることができる。通常の kill コマンドを用いても終了させられない場合にしばしば利用される強力なオプションを示せ。
9. アプリケーションの正常終了と強制終了の違いを述べよ。また、その具体例を示せ。
10. jobs コマンドを用いて得られるプロセスのリストと ps コマンドを用いて得られるプロセスのリストの違いは何か?
11. Microsoft Windows でプロセスを管理する方法を解説せよ。
12. 演習室 1 のワークステーションの /usr/bin というディレクトリに含まれているファイルの数を数えるには、どうすればよいか?
13. 演習室 1 から、「一行の命令」を用いて 演習室 4 のワークステーションの /usr/bin に含まれているファイルの数を数えるエレガントな方法を示せ。
14. getent passwd というコマンドでユーザのリストを得ることができる。これをを利用して、本学の学生の人数を数えるエレガントな方法を示せ。
15. 演習室の計算機の /etc/passwd ファイルの最終行の意味を解説せよ。
16. 本学の教員の人数を数えるエレガントな方法を示せ。
17. 会津大学にはメールサーバがいくつあるか?
18. 会津大学のニュースサーバの名前は何か?
19. getent hosts というコマンドで登録されている計算機をリストできる。これを用いて、CALL 教室の計算機を数えるエレガントな方法を示せ。
20. getent group をつかって、会津大学の UNIX グループの数を調べてみよう。
21. Emacs は X Window System に対応したソフトウェアであるが、コンソールや terminal emulator の中でも利用できる。どのようなオプションを使えば良いか?
22. Emacs を強制終了した場合に一時的に生成されるテンポラリファイルが削除されずに残る。このファイルの特徴を解説せよ。
23. ニュースに投稿されている記事をユーザ自身のファイルに保存するエレガントな方法を解説せよ。
24. Gnome-Terminal で使用する文字コードを変更する方法を示せ。
25. シャープ (#) で囲まれるファイル名を持つファイルはどの様な場合に生成されるか説明せよ。
26. チルダ (~) が末尾につくファイル名を持つファイルはどの様な場合に生成されるか説明せよ。
27. Emacs で、[IPA] モードに入ってしまった時の、復旧の仕方を解説せよ。
28. Emacs の c++-mode とはなにか?
29. Emacs の perl-mode とはなにか?
30. Emacs の latex-mode とはなにか?
31. Emacs でリモートファイルの編集を行う方法を解説せよ。
32. cat < filename と cat filename は、同じ働きをするが、なぜだろう。
33. cat > filename はどんな意味?
34. cat < f1 > f2 はどんな意味?
35. cat >> filename はどんな意味?
36. cat -n filename の動きは何?
37. grep コマンドの仲間に egrep, fgrep がある。それぞれの特徴を説明せよ。

38. env コマンドの利用方法を用いて解説せよ。
39. /home/student/s1XXXXXX と /a/stdfsuX/vol/volX/homeX/s1XXXXXX の違いと意味を解説せよ。
- クイズ2**
- ニュースの Subject には必ず [Quiz2-X] (X はクイズの番号) を書き込むこと。
40. xeyes の様に新しく窓を作るアプリケーションを実行する場合には & を付けると元の terminal emulator も利用可能である。xeyes はこの時 XXXXXX ジョブとして動作していると呼ばれる。XXXXXXX に当たるまる適切な言葉を示せ。
41. cp file1 file2 と同じ結果をもたらす操作を cat コマンドで実行するにはどうするか?
42. cat file | grep mojiretu
grep mojiretu < file
43. 標準入力や標準出力という言葉は何を表すのでしょうか?
44. ls * を実行して得られる結果を良く観察し、ls .-* の結果として得られるであろう出力の意味を説明せよ。
45. 命令
ls -l /home/student/s1234567
の結果として表示される文字の中に "->" がある。これは 2 つのディレクトリのつながる様子を表す。man ln を参考にしてリンクについての簡単な説明を以下
の点に注意して述べよ。
- 例えるとしたらどんなものに似ているか?
 - ファイルを繋ぐ操作の簡単なサンプルを一つ例示すること。
46. cat test > test のように、同名のファイルに直接書き込むとファイルの中身が空になってしまうのはなぜか?
47. 演習室1,2,3,4,5,6 の WS で
/usr/bin/ps -elf
を実行して得られるプロセスリストの中の PPID の意味は何か?

48. 演習室に設置されているスキヤナーの利用目的とスキヤナーをコントロールするソフトウェアの使用方法を説明せよ。
49. コマンド k2ps の利用目的を説明せよ。
50. 1 シートに4ページを配置する方法を示せ。
51. 命令 cal を利用して 2017 年の 5 月のかレンダーを表示する方法を書け。
52. 命令 cal を利用して 2017 年の 1 年間のかレンダーを表示する方法を書け。
53. 命令 cal を利用して 2017 年の 5,6,7 月のかレンダーをテキストファイル cal.txt に書き込む方法を書け。
54. UNIX 標準テキストエディタ vi の利用方法の中でカーソルの移動の方法を説明せよ。
55. UNIX 標準テキストエディタ vi の利用方法の中で行末に文字を挿入する方法を説明せよ。
56. UNIX 標準テキストエディタ vi の利用方法の中で新しい行を挿入する方法を説明せよ。
57. chmod 640 ファイル名と同じ操作を別の記法で書け。
58. Emacs の利用法の中で文字列検索を行なう方法を説明せよ。
59. Emacs の利用法の中で英単語のスペルチェックを行なう方法を説明せよ。
60. Emacs の利用法の中で文字列置換を行なう方法を説明せよ。
61. Emacs の利用法の中で矩形領域削除を行なう方法を説明せよ。
62. Emacs を端末として利用する方法を説明せよ。
63. ping コマンドとは何か? 利用する方法を説明せよ。
64. arp コマンドとは何か? 利用する方法を説明せよ。
65. traceroute コマンドとは何か? 利用する方法を説明せよ。
66. traceroute コマンドと同等の動作をする Windows にインストールされているプログラムは何か? 利用する方法を説明せよ。

67. netstat コマンドとは何か? 利用する方法を説明せよ。
68. netstat コマンドと同等の動作をする Windows にインストールされているプログラムは何か? 利用する方法を説明せよ。
69. rusers の動作と利用方法を説明せよ。
70. 'getent hosts | grep -e -rt' で出力されるデータを利用してルータ同士の接続の様子を図示せよ。
71. GNU Project の公式ホームページと会津大学FTPサーバー上のURLを示せ。
72. Cygwin とは何か? 調査せよ。
73. FreeBSD の公式ホームページと会津大学FTPサーバー上のURLを示せ。
74. Linux の各種ディストリビューションの公式ホームページと会津大学FTPサーバー上のURLを示せ。またその作業の方法を説明せよ。
75. time コマンドの利用法を説明せよ。
76. 各種 MUA (Mail User Agent) の特徴について述べよ。特にメッセージの管理方法についての違いを明らかにせよ。
77. 以下の操作と応答は何を意味するか?
- ```
ls -l ~/mozilla/*/*/lock
1rwxrwxrwx 1 a-fujii prof 20 Jan 13 10:52
/....../mozilla/akira8zniacg.slt/lock
-> 163.143.177.20:20303
```
78. 演習室の計算機へのアクセス記録を参照する方法を説明せよ。
79. Windows パソコンへのアクセス記録を参照する方法を解説せよ。
80. /var/log に存在するファイルについて解説せよ。
81. Windows で利用できる ssh クライアントをリストせよ。
82. Mac OS X で利用できる ssh クライアントをリストせよ。
83. xterm の背景色を会津グリーンにする方法を示せ。

## クイズ Search

検索サイトを利用して、以下のうち1項目を検索し、問い合わせよ。ファイル Search に検索に使用したサイト、キーワード、結果、問い合わせに対する回答を詳しくレポートせよ。ニュースの Subject には必ず [Quiz-S-X] (X はクイズの番号) を書き込むこと。

84. POSIX とは何か調査し報告せよ。
85. TCP/IP とは何か?
86. RFC とは何か?
87. DDNS について調査し報告せよ。
88. メールでテキスト文書以外のファイルを送るための仕様が書かれた文書は何か?
89. 無線 LAN の最新の規格について調査せよ。
90. WWW を最初に作り出したのは誰か?
91. n 次元球の体積は何か?
92. 球面調和関数の満たす微分方程式を検索せよ。
93. 過去 10 年間の為替変動グラフを探せ。
94. 過去 10 年間の空気中の二酸化炭素濃度の変動グラフを探せ。
95. 環境ホルモンのリストを探せ。
96. 口内炎に有効な食品を探せ。
97. レバニラいための作り方を検索せよ。
98. 美味しいご飯の炊き方を検索せよ。
99. 海洋汚染の影響による魚介類の食事制限を調査せよ。
100. 生活習慣病を予防するウォーキングの方法について検索せよ。
101. 肥満を解消するウォーキングの方法について検索せよ。
102. 定期的な防虫剤散布を停止した自治体を検索せよ。
103. GNU 版 paste コマンドのマニュアルを検索せよ。
104. GNU 版 join コマンドのマニュアルを検索せよ。
105. 発光ダイオードの仕組みを調べよ。
106. 光電効果を解説する良いホームページを検索せよ。
107. GPS の動作と仕組みを調査せよ。
108. 平均と分散を解説しているホームページを検索せよ。
109. "R による統計処理"を解説しているホームページを検索せよ。
110. 陽子や中性子の質量の起源を解説しているホームページを検索せよ。
111. ニュートリノの質量に関する最新の実験と測定値を調べよ。
112. LHC の実験で得られた最新の情報を調査せよ。
113. 会津大学の初代学長の業績をインターネットを利用して調査せよ。
114. 会津大学の第 2 代学長の業績をインターネットを利用して調査せよ。
115. 会津大学の第 3 代学長の業績をインターネットを利用して調査せよ。
116. 会津大学の第 4 代学長の業績をインターネットを利用して調査せよ。
117. 会津大学の第 5 代学長の業績をインターネットを利用して調査せよ。
118. ネットワーク上の複数のコンピュータでユーザ情報を共有するシステムは NIS の他に何があるか。調査せよ。
119. 各種 MTA (Mail Transfer Agent) の特徴について調査せよ。
120. 会津大学に関する (Web) 掲示版を検索し、リストせよ。また、検索の方法について解説せよ。
121. spec.org を検索し、演習室 WS の性能を調査し、報告せよ。
122. LATEX を HTML にコンバートするツールを調査し、利用方法を解説せよ。
123. Sunray を利用したシステムを利用している自治体を検索し、報告せよ。
124. OpenOffice を利用したシステムを利用している自治体を検索し、報告せよ。
125. 将来のコンピュータ演習室を提案し、導入、維持管理コストについて具体的に述べよ。
126. 生体情報を利用した認証システムを利用した出席管理について調査し、報告せよ。
127. Windows で利用できる X Window System を調査せよ。
128. Windows で UNIX のツール群を導入し使用することのできる便利な方法を調査し、報告せよ。
129. Mac OS X で UNIX のツール群を導入し使用することのできる便利な方法を調査し、報告せよ。
130. NetNews のあるニュースグループのすべての記事を一つのファイルにまとめるエレガントな方法を調査せよ。
131. Mathematica の価格とキャンパスライセンスの価格を調査せよ。
132. Microsoft Campus Agreement の内容と価格を調査せよ。
133. 各大学に導入されている図書館システムを調査しりストせよ。
134. 各大学に導入されている大学情報管理システムを調査しりストせよ。
135. よく利用されているウェブ (WEB) サーバソフトウェアと、現時点におけるそれぞれのセキュアなバージョンをリストせよ。
136. WEB サーバにおいて利用される CGI サービスを開発するプログラム環境をリストせよ。
137. 演習室の計算機で Linux 用のプログラムを動作させる方法を調査せよ。
138. 演習室の計算機で Windows 用のプログラムを動作させる方法を調査せよ。
139. 演習室の計算機で DOS 用のプログラムを動作させる方法を調査せよ。
140. 過去 20 年間の標準的なパソコンの CPU の性能の変化を調査せよ。また、価格対性能のグラフを作成せよ。
141. 過去 20 年間の標準的なパソコンの HDD の容量の変化を調査せよ。また、価格対容量のグラフを作成せよ。

142. 過去 20 年間の標準的なパソコンのメモリの容量の変化を調査せよ。また、価格対容量のグラフを作成せよ。
143. 各演習室の WS の価格を調査せよ。
144. ソフトウェアの使用ライセンスの種類を調査せよ。また、それぞれの特徴と差異を解りやすく解説せよ。
145. GPL とはどの様なライセンスか? これにより何が保証されるのかを解説せよ。
146. オープンコンピューティングを調査し解説せよ。
147. オープンソースソフトウェアを調査し解説せよ。
148. FSF(フリーソフトウェアファウンデーション)によるフリーソフトの定義を調査し解説せよ。
149. su コマンドの利用方法と Windows で同等の操作をする方法を調査し、解説せよ。
150. Windows でバックグラウンドジョブを実行する方法を調査し、解説せよ。
151. ユビキタスコンピューティングについて調査し解説せよ。
152. ログインスクリーンの画像をキャプチャする方法を調査し解説せよ。
153. ネットワーク障害が発生しているネットワークを調査する方法を解説せよ。
154. 一様乱数からガウス分布に従う乱数を生成する方法を調査せよ。

158. 上の問題で作成された file0 に対して以下の操作をすは file1 に何を出力するか?
- ```
gawk 'BEGIN{ srand(); ifprint rand(),$1}; file0|\
sort +1 -n| awk '{print $2}' >file1
```
159. 上の問題で作成された file1 と file2 を組み合わせてリストを作成する方法を提案せよ。
160. /home/course/lit1/pub/literacy/ にあるスコアファイル physics.csv を観察し、追試の結果を考慮した成績リストを求める awk のプログラムを提案せよ。
161. /home/course/lit1/pub/literacy/ にあるスコアファイル english.csv と english-makeup.csv を比較して追試の結果 50 点を越えた者の成績を 50 にする処理を awk や paste 等のツールを用いて書け。
162. Awk を用いてデータの行と列を入れ替えるプログラムを書け。

163. テキストファイルに行番号をつけて表示する awk スクリプトを書け。
- クイズ3
ニュースの Subject には必ず [Quiz-A-X] (X はクイズの番号) を書き込むこと。
164. Awk を用いて元データからヒストグラムを生成するプログラムを書け。
165. Awk を用いて 10000 個の乱数を発生させよ。またその乱数を用いて円周率の値を数値計算する方法を示せ。
166. 以下の様に書かれたファイル ph.awk がある。
- ```
ph.awk
BEGIN{
 dt=0.01;
 t=0.0;
 omega=10.0;
 baneteisuu=0.1;
 teikou=0.3;
 gairyoku=0.21;
 constforce=0.001;
 x=0.0;
 p=0.0;
}
{
 t+=dt;
 x+=p*dt;
 force=constforce*baneteisuu*x;
 p+=force*dt;
 print t,x,p;
}
```
- これに対して、
167. 上の問い合わせ同様にして、
- ```
seq 1 3000 | awk -f ph.awk |\
awk '{print $1,$2}' | xmgrace -
```
- を実行すると何が得られるか?
168. スピログラフを出力する awk プログラムを作成せよ。
- クイズ3
ニュースの Subject には必ず [Quiz3-X] (X はクイズの番号) を書き込むこと。
169. 会津大学の卒業論文のためのクラスファイルの所在を示せ。
170. LATEX を利用して縦書きの文書を作る方法を説明せよ。
171. jarticle の基本クラスファイルが置かれているディレクトリを示せ。

172. `platex` の `README.txt` を読み、pTeX と LaTeX の最新情報を公開しているホームページを報告せよ。

173. `man dvips` を読み、-pp オプションの利用方法を説明せよ。

174. `tabular` 環境の中で `multicolumn` を利用する方法をサンプルを例示して説明せよ。

1	2	3	4
3		7	
	10		

175. `tabular` 環境では各要素を右寄せ、左寄せ、センタリングをするためにそれぞれ、`r`, `l`, `c` を使って設定する。`p{3cm}` も使用できるが、どの様に利用するのが便利なのだろうか？便利な使用法の具体例を示せ。

176. 以下のメッセージが出てきてコンパイルに失敗する場合の原因を説明せよ。

```
! LaTeX Error: File 'jreport.cls' not found.  
Type X to quit or <RETURN> to proceed,  
or enter new name. (Default extension: cls)  
Enter file name:
```

177. 以下のメッセージが出てきてコンパイルに失敗する場合の原因を説明せよ。

```
! LaTeX Error: Missing \begin{document}.
```

See the LaTeX manual or LaTeX Companion
for explanation.
Type H <return> for immediate help.
..

1.1 感じ
?

178. $ABCD \dots XYZ$ の印字の方法を説明せよ。

179.

$$\oint_C f(z) dz$$

の印字の方法を説明せよ。

180. $\int_C f(z) dz$ と $\oint_C f(z) dz$ の違いを説明せよ。

181. $\sqrt[3]{a}$ の印字の方法を説明せよ。

182. `\section{}` と `\section*{}` の違いを述べよ。

183. ページ番号を印字しないための命令を書け。

184. 目次を出力する方法を説明せよ。

185. `enumerate` 環境を利用して

- I. アイテム 1
- II. アイテム 2

の印字の方法を説明せよ。

186.
$$\begin{aligned} a_{1n}x_n &= b_1 \\ a_{2n}x_n &= b_2 \\ a_{3n}x_n &= b_3 \end{aligned} \quad \cdots (1)$$

の印字の方法を説明せよ。

187.
$$\underbrace{a + b + \dots + y + z}_{24}$$

の印字の方法を説明せよ。

188.
$$\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{x^2 + 1}}}}}$$

の印字の方法を説明せよ。

189.
$$\begin{pmatrix} x & y \\ -y & x \end{pmatrix}^2 = (x^2 - y^2) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + 2xy \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

の印字の方法を説明せよ。

190. $\overline{\text{アンダーライン}}$ を印字する方法を説明せよ。

191. `\overline{A}` の印字領域の幅を指定する方法を説明せよ。

192. 4cm の縦方向の空白を挿入する方法を説明せよ。

193. \swarrow , \searrow , \nwarrow , \nearrow , \leftarrow , \rightarrow , \downarrow , \uparrow の印字の方法を説明せよ。

194. 関数 $f(x) = (\log x)^2$ の増減表を LATEX で表示せよ。

195.
$$\frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{x-1}}}}$$

の印字の方法を説明せよ。

196. $1\text{Volt} = 1\text{A} \times 1\Omega$ の印字の方法を説明せよ。

197. $10 \geq x^2 + 1$ の印字の方法を説明せよ。

198.
$$\int_S (\nabla \times \vec{B}) \cdot d\vec{s} = \int_{C=\partial S} \vec{B} \cdot d\vec{r}$$

の印字の方法を説明せよ。

199.
$$\int_V (\nabla \cdot \vec{E}) dv = \int_{S=\partial V} \vec{E} \cdot d\vec{s}$$

の印字の方法を説明せよ。

200. $A \rightarrow B \equiv \bar{B} \rightarrow \bar{A}$ の印字の方法を説明せよ。

201. $A = \{x | \sqrt[3]{x} \leq 3, x \in R\}$ の印字の方法を説明せよ。

202.

$$\sum_{k=1}^{\infty} k = -\frac{1}{12}$$

の印字の方法を説明せよ。

203.
$$\begin{pmatrix} x & y \\ -y & x \end{pmatrix}^2 = (x^2 - y^2) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + 2xy \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

の印字の方法を説明せよ。

204. $\overline{\text{アンダーライン}}$ を印字する方法を説明せよ。

205.

1ページの印字領域の高さを指定する方法を説明せよ。
1ページの印字領域の幅を指定する方法を説明せよ。
 $dx \wedge dy = (dr \cos \theta - r \sin \theta d\theta) \wedge (dr \sin \theta + r \cos \theta d\theta)$

206.
$$\left(\left(\left(\left((x+1) \right) \right) \right) \right)$$

の印字の方法を説明せよ。

207. 四角に囲まれた部分の印字の方法を説明せよ。

208. `@_avv=$sum/_$num`

の印字の方法を説明せよ。

209. コマンド `latex` と `plateax` の違いを説明せよ。

210. `TeX` を開発した人の名前と職業を説明せよ。

211. `TeX` を開発した人の名前と職業を説明せよ。

212. `TeX(LaTeX も含む)` の話題のニュースグループを示せ。

213. AMS-`TeX` の利用方法を解説せよ。

214. list 環境の利用法を解説せよ。

215. 以下の命令による出力を示し、解説せよ。

```
\newcounter{abc}
\newcommand{\abc}{%
\addtocounter{abc}{1}\theabc. %
}%
\abc\abc\abc\abc
```

216. 以下の印字方法を示せ。

```
Title
Hello.
```

217. 以下の印字方法を示せ。

```
Title
Hello.
```

218. 以下の印字方法を示せ。

```
Hello.
```

219. Windows で `TeX` を利用する方法を解説せよ。

221. Windows で `TeX` の DVI ファイルを表示する方法を解説せよ。

クイズ4

ニュースの Subject には必ず [Quiz4-X] (X はクイズの番号) を書き込むこと。

222. コマンドラインで

`~/literacy/see 3 tex`

を実行するとカレントディレクトリから数えて3つ下までのディレクトリをサーチして `tex` という文字列をファイル名に持つものだけを表示してくれるコマンド `see` をシェルスクリプトで作成せよ。

223. ハンドアウト中の `gamemenu` というシェルスクリプトを `perl` で書きなおせ。

224. データの平均値を計算するプログラムを `perl` で書いてみましょう。

225. 以下の命令を実行して得られる出力を元に課題毎に配点を設定して合計得点を得られるようなスクリプトを作成してみましょう。

```
grep sxxxxxx \
~a-fujitu/education/literacy/2017/summer/* .sc
```

226. コマンドライン上で `getent passwd` の標準出力から教職員だけを取り出し、そのリストを元にそれぞれの教員のホームページをリストする HTML 文を出力する自動生成のプログラムを作成せよ。ただし、ここでいう教職員とは `home directory` が `/home/professor/*` のユーザーとする。余力があれば、データを `sort` して(辞書順列で並べること)出力すること。その出力をファイルに保存して、ウェブプラウザで見てみよう。

227. カレントディレクトリのテキストファイルのみリストするスクリプトを書け。

228. `tcsh` は `csh` を基本にして、どのような機能が拡張されているか説明せよ。

229. Mac OS X で `TeX` を利用する方法を解説せよ。

230. シェル変数の値を設定するには `set` コマンドを使うが、間違って設定した変数を取り消す時にはどんなコマンドを使うか?

231. `csh` では環境変数の値を設定するには `setenv` コマンドを使うが、変数を取り消す時にどんなコマンドを使うか?

232. `csh` でプロンプトにカレントディレクトリを表示させる設定を示せ。

233. `tsh` でプロンプトにカレントディレクトリを表示させる設定を示せ。

234. `bash` でプロンプトにカレントディレクトリを表示させる設定を示せ。

235. `zsh` でプロンプトにカレントディレクトリを表示させる設定を示せ。

236. PID が 0 のプロセスの働きを解説せよ。

237. PID が 1 のプロセスの働きを解説せよ。

238. PID が 2 のプロセスの働きを解説せよ。

239. テキストファイル `a.txt` の 1 行目から 100 行目に順に `a1.a2,...a100` が書かれている。10 行ずつを一行にまとめる最も効率の良い方法を提案せよ。

240. N 行 M 列のデータがテキストファイルに書かれている。行と列を置換したものを作り出す最も効率の良い方法を提案せよ。

241. UNIX の `&` (バックグラウンド処理) に相当する操作を Windows で行う方法を解説せよ。

242. `/usr/bin` や `/usr/sbin` などにあるコマンドの中で、`setuid` 属性が `on` になっているものをリストし、なぜ `setuid` 属性を `on` にする必要があるのかを幾つかのコマンドの動作を例に説明せよ。

243. ステイックビットが `on` になっているディレクトリやファイルをリストせよ。また、通常のファイルやディレクトリとの扱いの違いを説明せよ。

クイズ その他のツール

ニュースの Subject には必ず [Quiz-T-X] (X はクイズの番号) を書き込むこと。

244. OpenOffice の表計算でデータの行と列を入れ替える操作を解説せよ。

245. OpenOffice の表計算で元データからヒストグラムを生成する方法を解説せよ。

246. Sylpheed 以外の演習室で利用可能な MUA をリストせよ。

247. Mathematica のノートブックのフォントサイズを変更する方法を解説せよ。

248. Mathematica 等を利用して行列

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 2 \end{pmatrix}$$

の逆行列 A^{-1} を求めよ。

249. Mathematica 等を利用して前問の行列 A の固有値と固有ベクトルを求めよ。

250. Mathematica 等を利用して周期 2、振幅 1 の関数 $f(x) = (x \bmod 2) - 1$ のフリー工成分を求める。

251. Mathematica 等を利用して球面調和関数の 3 次元プロットを表示せよ。

252. あるファイルに文字 a が含まれている個数を調査するためのエレガントな方法について解説せよ。

253. あるファイルの n 行目だけを表示するエレガントな方法について解説せよ。

254. angeftp の利用方法を解説せよ。

255. tramp の利用方法を解説せよ。

256. 2 枚の画像を連結する方法を解説せよ。

257. 1 枚の画像を縦 n、横 m に分割し、n × m 枚の画像に分割する方法を解説せよ。

258. scp の利用方法を解説せよ。

クイズ レポート

ニュースの Subject には必ず [Quiz-R-X] (X はクイズの番号) を書き込むこと。

259. SFU (Microsoft Windows Services for UNIX) について調査し、その存在の意味について考察せよ。

260. SUA (Subsystem for UNIX applications) について調査し、その存在の意味について考察せよ。

261. Windows PowerShell について調査し、その存在の意味について考察せよ。

262. Windows 以外のシステムで NTFS を利用するための方法を調査し、報告せよ。

263. Postscript で記述されたファイルを Windows で表示するための方法を調査し、報告せよ。

264. 各種ファイルシステムの特徴を調査し、報告せよ。

265. getent hosts の出力からマルチホーム、すなわちネットワークインターフェースを複数持っている計算機のリストを作成し、それらの相互接続図を完成せよ。

クイズ 会津地域情報

ニュースの Subject には必ず [Quiz-Aizu-X] (X はクイズの番号) を書き込むこと。

266. 会津地域の"病院"のホームページのリストを種類別に作成せよ。

267. 会津地域の"祭り"やイベントのホームページのリストを作成せよ。

268. 会津地域の"図書館"のホームページのリストを作成せよ。

269. 会津地域の"病院"の Map を作成せよ。

270. 会津地域の"スーパー"の Map を作成せよ。

271. 会津地域の"コンビニ"の Map を作成せよ。

272. 会津地域の"銀行、郵便局、ATM"の Map を作成せよ。

274. 会津地域の"公園"の Map を作成せよ。

275. 会津地域の"パン屋"の Map を作成せよ。

276. 会津地域の"ラーメン屋"の Map を作成せよ。

277. 会津地域の"カツ丼屋"の Map を作成せよ。

278. 会津地域のゴミ収集の情報のホームページの URL を調べよ。

1.10 競争的課題

この課題は競争的課題です。通常の課題などでは満足できない人は是非チャレンジしましょう。

- 著作権の無い古典作品の絵本
- 創作絵本
- リテラシーの挿絵
- ニュースの説明

1.10.1 作品のエントリ

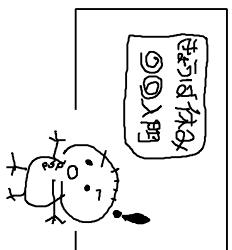
以下に指定したファイル名で作品と作品説明のファイルを\$HOME/literacy/compe/に21回目の演習の行われる日(20:00まで)に作成してエントリーします。説明ファイルはUTF-8 エンコードのテキスト形式で書式は

作品説明ファイル

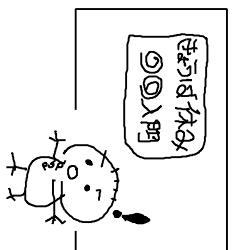
```
#title タイトル
#author 作者
#description 説明文
```

内	作品ファイル名	説明
チケットのポスター	poster.pdf	作品ファイル 説明ファイル 作品ファイル
絵本	ehon.pdf	説明ファイル 作品ファイル 説明ファイル
リテラシーの挿絵	sasie.eps	作品ファイル 説明ファイル 説明ファイル
企画書	kikaku.pdf	作品ファイル 説明ファイル 作品ファイル
アニメーション	animation.gif	説明ファイル 作品ファイル
	animation.txt	

です。作品は締め切りの翌日から指定されたWebサーバで学内公開されます。



- 企画書
- 商品企画



1.10.3 投票

投票により各テーマの上位3作品にそれぞれ20ポイントを、上位3批評にそれぞれ10ポイントを与えます。以下に指定したファイルを\$HOME/literacy/compe/に最終回の演習の行われる日(20:00まで)に作成して投票します。UTF-8 エンコードのテキスト形式で書式は

投票ファイル

```
#work 5
#rev 3
```

です。最も気に入った作品の番号と役に立った批評の番号をそれぞれのファイルに記入してください。

1.10.1.1 各テーマの説明

表現したいテーマ等は基本的に自由です。以下ではテーマの例を列挙します。

- ネチケットのポスター

- なりすましは許しません
- コンピュータ破壊禁止

- 絵本

1.10.2 批評のエントリ

以下に指定したファイル名で批評を\$HOME/literacy/compe/に23回目の演習の行われる日の(20:00まで)に作成してエントリーします。UTF-8 エンコードのテキスト形式で書式は

内	投票ファイル名	
チケットのポスター	poster.bid	
絵本	ehon.bid	
リテラシーの挿絵	sasie.bid	
企画書	kikaku.bid	
アニメーション	animation.bid	

1.10.4 参加賞

すべての投票に参加した場合に4ポイントを与えます。

批評ファイル

```
#work 1
作品番号 1 の批評
#work 4
作品番号 4 の批評
....
```

とします。作品批評は翌日から指定されたWebサーバで学内公開されます。

内	批評ファイル名	
チケットのポスター	poster.rev	
絵本	ehon.rev	
リテラシーの挿絵	sasie.rev	
企画書	kikaku.rev	
アニメーション	animation.rev	

第2章 UNIX, Mail, News, Web

2.1 はじめに

この資料中のコマンドの文法の記述で、イタリック体(*italic*)で記述されている部分は、そのまま打ち込むのではなく、そこに該当する適当な名前を入力するということです。例えば、*filename*とあつた場合は、*filename*と打ち込むのではなく、そこに適当なファイル名を打ち込むということです。また四角括弧 [...] で囲まれた部分は、省略可能であることを意味します。

2.2 UNIX

UNIX の成り立ち UNIXは、1969年にAT&Tベル研究所で開発されたOS(オペレーティングシステム)です。もともとベル研ではMULTICSというミニコン用のOSが開発途中でした。MULTICSが非常に大きなシステムで、当時のミニコンの処理能力に對して重すぎました。そこで、もっとコンパクトで軽いシステムを作ろうと開発されたのがUNIXでした。UNIXには、その後何度も改良が加えられました。とくに大きい特徴は、カーネルと呼ばれるUNIXの中心となる部分を除いて、すべてC言語で書かれていることです。これによつて、様々な機種への移植が容易になり、スーパーコンピュータから、パーソナルコンピュータまで、広く使われるようになりました。

UNIXについての解説書や雑誌は大変多く出版されていて現在では大変身近なものになりました。その中から参考図書として

- 『UNIX原典 AT&T ベル研究所編』
 - 『UNIXプログラミング環境 B.W.Kernighan and R.Pike』
- を紹介します。興味のある人は是非読んでみてください。
- UNIXの特徴** UNIXの特徴を述べておきましょう。UNIXはもともとインターネット開発の基本OSとしても利用されてきたことから、UNIX上で機能するネットワークのサービスはインターネットにおいて今日まで事實上の国際標準(デファクトスタンダード)となっています。ネットワークの機能が豊富で安定しており、他のOSに比較して安全性が高いといつることができます。ネットワークの機能が豊富で安定しており、他のOSに比較して安全性が高いといつことができます。したがって、UNIXは
- ネットワークを利用して利用者やハードディスクなどの情報資源を集中管理すること。
 - ネットワークを利用してその他のサービスを提供すること。
(DNS, MAIL, NEWS, WEB, NTP Serverなど)

などの用途に非常に適しています。UNIX以外のOSでは(Internet接続を前提とした)ネットワークの機能の開発の歴史が浅く、個人利用に限れば発生する問題に対し個人で責任を持ちうることができるので利用は可能ですが、Internetに接続することやネットワーク上でサービスを提供すること、(教室のように)多くの利用者が共用することには問題があります。

GNUを始めとするいろいろな団体やユーザがUNIX上のアプリケーションやサービスを開発しています。しかもそれらのうちの多くが無償で提供されています。(EmacsやTeXなどが良い例です。)

2.2.1 なぜ演習室はWindowsパソコンではなく、UNIXワークステーションか?

実際に現在のエントリーレベルのUNIXワークステーションはパソコンで利用される部品とほぼ同じ性能を持つ部品で構成されています。これは、近年になりパソコンの部品の性能が著しく進歩した為です。

もしも構成が同じあれば本体の価格差はありません。メモリなどの信頼性を左右する部品についてはパソコンとワークステーションでは異なっています。では"なぜWindowsではなくUNIXなのか?"に対する回答を以下に列挙します。

- 全ての教室の全ての端末で共通の基準を満たす環境を提供すること。
- NFSに同等であつてかつ、性能の高いサービスが必要です。

- 伝統的な情報処理教育が追加の費用を最小にして無理無く行えること。
- 多くの優良なフリーソフトウェアが使用できること。

- GCC, Perl, Python, Ruby など。
- Firefox, Thunderbird など。
- TeX, Emacs など。
- Tgif, Xfig, Gimp, Xmgrace など。
- Aptech, PostgreSQL, MySQL, PHP など。
- 大規模ネットワークシステムとして構築した場合にも管理コストが急激に上昇しないこと。
- コンピュータ業界、IT業界の基礎技術を追加の費用を最小にして教育できること。
- サーバ系はUNIXの知識が必須です。
- 複数のアプリケーションを導入利用しても不安定にならないこと。
- パソコンは複数のアプリケーションが競合した場合は異なるパソコンにアプリケーションを分けて導入し利用します。

- ・長期間連続運用が可能なこと。

- セキュリティのための更新が頻繁でないこと。

- ・Windowsは確かに一般ユーザのための使いやすさを追求しているが、教育目的には不足している。

- Windows付属のシェルが貧弱で管理に利用できない。
Microsoft自身がSFUやPowerShellによって補っている。
- UNIXで培ってきた多くのツール群をパワーユーザはインストールする。

以上の回答は現在までの実績に基づいて述べられたものです。もしも、Windowsにより素晴らしい教育環境、研究環境が実現できるのであれば、会津大学の演習室にも自然とWindowsワークステーションが導入されるかも知れません。世の中というものは教育や研究にとって理想的な方向に変化するとは限りません。例えば、大学のカリキュラムの一部ではWindows用の特殊な装置を利用する必要があり、Windowsの端末を利用する必要が出てきました。その為にハードウェア実験室1,2にはWindowsの端末が追加されています。

教育を受ける側の事情としては、以下のことが予想されます。

UNIXベースで教育を受けた(優秀な)学生はより広い視野を持って、将来も他のコンピュータ技術を取り組むことができる。一方、そうでなければ、UNIX系の技術習得の際に、より大きなストレスを感じることで、これを避けるようになる。

また、最近の技術者と呼ばれる人々の中には、残念なことです、以下の様な情報処理技術の不足した者が出現しています。

- ・サーバの重要性を理解していない。

- ・サーバのログを解析して、異常なイベントを検出することができない。
 - sortしてパターンで分類すると良い。
 - grepで特殊な文字列を含むデータを検出すると良い。
 - grepで正常なイベントを削除したデータを生成すると良い。
- ・サーバのログデータの集計にExcelを利用しデータサイズが大きすぎて時間以内に処理できない。
- awkを用いて必要なフィールドのみ集計すれば良い。

UNIXベースで教育を受けていない”技術者”が増えているので業界の常識やモラルが以前に比べてとても低レベルになってしまっていると感じます。

コンピュータはそれを使いこなす道具がなければ役に立ちません。一般的利用者はコンピュータを使って特定のアプリケーションを利用するだけですが、コンピュータの専門家であれば限られたアプリケーションだけではなく、あらゆる道具を知つていなければなりません。また、本物の職人であれば、適切な道具が無ければ作ったり、チューニングして使います。皆さんには、まだコンピュータの専門家でないかもしれません。したがって、会津大学の教育のための共通基盤は、

- ・伝統的なツール群を標準で備えている

- 小さなコンピュータからスーパーコンピュータまで利用されている

- 以上の特徴を持つUNIXを利用できるように設定されています。また、UNIXは
 - ・サーバOSなので、複数のユーザプロセスが無理無く動作する
 - ・NFSサーバを利用することで「透過的」利用を提供できるなどの優位性を持っています。

もちろん、UNIXを使いこなすことが最終的な目標ではありません。皆さんが、UNIXを通してコンピュータの基礎や情報処理の伝統的手法、インターネットの利用の作法を学ぶことが目標です。これらの基礎の上に立ってコンピュータサイエンティストとして活躍できる様になって貰いたいと希望します。

さあ、学生諸君。後になって一人で勉強するのは大変かも知れません。まずは、Computer Literacyで易しいところからUNIXを始めてみましょう。

2.2.2 Mac OS X 端末

演習室1,2には独立したデスクトップがあり、演習室3,4,CALL教室にはSunrayサーバを利用するSunray端末があります。また、中講義室、大講義室やハードウェア実験室1,2,3,4にはWindowsPCがあり、それら経由でSolarisサーバ(solsv1[-4])にアクセスできます。
<http://web-int.u-aizu.ac.jp/Labs/istc1pc/faq/lml/lml.html>

2.2.3 Solaris 環境

演習室1,2には独立したデスクトップがあり、演習室3,4,CALL教室にはSunrayサーバを利用するSunray端末があります。また、中講義室、大講義室やハードウェア実験室1,2,3,4にはWindowsPCがあり、それら経由でSolarisサーバ(solsv1[-4])にアクセスします。

2.2.4 aplisv1 - aplisv2

aplisvサーバは共同利用サーバです。学生モリモートアクセスできます。

2.2.5 hdw[12]dc1-hdw[12]dc49,hdw[34]dc1-hdw[34]dc25

ハードウェア実験室1,2,3,4にはWindowsPCデスクトップがあります。回路設計用演習サーバcadsv1[-6]に、これらのWindowsPCからアクセスします。

2.2.6 provncsv1, stdvncsv1

VNCサーバ利用のための計算機です。

2.2.7 UNIXのコマンド(命令)

UNIXの主なコマンドについて説明します。ファイルやディレクトリに関するコマンドは、この後に記載します。ユーザーの login や, login 状況等を管理するコマンドには、表 2.1 のものがあります。

命令	機能	文法	説明
getent	管理データの取得	getent database	database の管理データを表示する。 <i>/etc/nsswitch.conf</i> に database のリストを表示。
whoami	ログイン名の取得	whoami	ユーザのログイン名を得る
who	ログインユーザー名の取得	who	同一の計算機にログインしているユーザ名を得る
finger	ユーザ情報を得る	finger -m [name]	[name]で指定されるユーザの情報を得る。 <i>name</i> には、姓、名、ログイン名のいずれかを指定する。 <i>name</i> の指定がない場合は、ログインしているユーザの情報を得る。

表 2.1. コマンド例 (1). 各コマンドの厳密な文法や、機能の詳細は、オンラインマニュアルや Info を参照して下さい。

2.2.8 オンラインマニュアル

UNIXを使っていると、コマンドの意味やオプションの意味、あるいはこのコマンドにはどういうオプションがあるのかといったことを調べたくなる時があります。こんな時はマニュアルを見るわけですが、UNIXには紙に印刷したマニュアル以外に、オンラインマニュアルというものが用意されています。オンラインマニュアルというのは、コマンドの意味や使い方、オプションの意味などを WS の画面で見ることができる仕組みのことです。オンラインマニュアルを見るには、man コマンドを使います。man コマンドの使い方は、以下の通りです。

man command	commandに対するオンラインマニュアルを表示する。
man -k keyword	keywordに関連するコマンドとその機能の一覧を表示する。

例えば、ls コマンドのマニュアルを見るには、man ls を実行します。

Webで検索するといろいろなコマンドのマニュアルを見ることがあります。ただし、同じコマンド名でもバージョンが異なったり、開発者の異なるものでオプションや動作が異なるものや、全く別物であることもあるので Web の情報は注意して利用すべきです。

2.2.9 UNIXのファイルシステムの特徴

2.2.9.1 全てがファイル

ディレクトリもファイルであるということは、既に述べました。UNIXでは、入出力を伴うものは全てファイルとみなされます。キーボードも、コンソールや各々のウインドウ、ハードディスク、プリンタに至るまで、全てファイルです。従って、これらは全ていざれかのディレクトリの下に入っています。実際に

ls /dev

で確認できます。

ただし、これらの入出力の機器は、それぞれ個別の制御命令を持つていて、それを一々ユーザが指定するのは困難です。そこで、実際には、これらの制御情報を持つている、デバイスドライバというプログラムが、入出力を管理します。従って、ファイルシステムにあるのは、デバイスドライバということになります。

2.2.9.2 特別のファイル—標準入力(stdin)・標準出力(stdout)

UNIXには、標準入力、標準出力という概念が存在します(標準エラー出力というのも存在しますが、これについては後日配布される講義資料や参考書等を見て下さい)。標準入力と標準出力を一緒にして、標準入出力という呼び方をすることもあります。UNIXには、標準入力からデータを読み込んだり、標準出力に書き出すという動作をするコマンドが多くあります。この標準入出力は、通常何も指定しなければ、標準入力はキーボードに、標準出力はディスク(フォーカスされているウインドウ)に割り当てられます。そして、この割当をユーザーの指定でファイル等に変更できます。

UNIXでは、すべての入出力がファイルとして扱われると言えました。つまり、標準入出力に取り替えるだけ、ということです。すべての入出力をファイルとして扱うことは、このような処理の抽象化、概念の簡略化をもたらします。

UNIXの割当の変更は、単に割り当てられたファイルをキーボードやディスクから他のファイルに切り替えるだけ、ということです。すべての入出力をファイルとして扱うことは、このようないくつかの方法があります。

標準入出力の割当の変更を、リダイレクトと呼びます。例えば、ls というコマンドは、ファイルの一覧を標準出力に出力します。これを、filenoichiran というファイルにリダイレクトしたければ、次のように入力します。

ls > filenoichiran

UNIXの標準入出力のリダイレクト先には、パイプという特別のものがあります。これは、一つのコマンドの標準出力を、別のコマンドの標準入出力につなげる働きをします。command1 というコマンドの標準出力をパイプにリダイレクトして、command2 というコマンドの標準入力につなげるには、次のように入力します。

command1 | command2

例を示しましょう。

```
ls /usr/bin | grep aw
```

これはディレクトリ /usr/bin にあるファイルのリストの中で aw という文字列を含む行を表示します。パイプを使ってたくさんのコマンドを次のようにつぐこともできます。

command1 | command2 | command3 | ... | commandn > file

パイプでの使用を前提に作られたのが、フィルタと呼ばれるコマンド群です。フィルタは、標準入力からデータを読み込んで、適当な処理を施した後、標準出力に書き出すという動作をします。上記の例では、*command2*, *command3*, ... が、フィルタの役目をしています。この機に幾つかの単機能のツールを組み合わせて複雑な情報処理を行うことができます。

フィルタとファイルを扱うコマンドの内いくつかを、表2.2に掲げておきます。

命令	機能	文法	説明
wc	ワードカウント	wc	入力の行数、単語数、文字数を数える。
sort	並べ換える	sort	入力を順に並べ替える。
grep	パターン検索	grep pattern	pattern を含む行を抜き出す
head	先頭抜きだし	head [num]	入力の先頭から num 行抜き出す。num が省略されたときは、10 行抜き出す。
tail	末尾抜きだし	tail [num]	入力の末尾から num 行抜き出す。num が省略されたときは、10 行抜き出す。
more	ページヤ	more	本文参照
less	ページャ	less	本文参照
cat	ファイル内容の表示	cat file	file の内容を表示する。
ls	ファイル一覧	ls path	path で指定されるディレクトリの下のファイルの一覧を表示。path が省略されたときは、カレントディレクトリが指定されたとみなされる。
rm	ファイルの削除	rm file	file を削除する。
cp	ファイルの複写	cp file1 file2	file1 の複写を file2 という名前で作る。file2 がディレクトリの場合は、そのディレクトリの下に file1 と同じファイル名で複写される。
mv	ファイルの移動	mv file path	file を path へ移動する。path がファイル名であった場合は、ファイル名の付け換えが行なわれる。

表2.2. フィルタ. 各コマンドの厳密な文法や、機能の詳細は、オンラインマニュアルを参照して下さい。

2.2.9.3.1 ページャについて *man* コマンドで表示されるオンラインマニュアルの文章は、とても長いものが多く、大抵の場合とても1画面では表示し切れません。そこで、*man* コマンドは *more* というプログラムを呼び出して、文章を表示させます。*more* のようなコマンドを一般にページャと呼びます。ページャは、1画面に入り切らない文章（ファイル）を、少しづつ表示して文の最初の方が画面の上に出でてしまうのを防ぎ、ユーザーが文を最初から最後まできちんと読めるようにするプログラムのことです。*less* は *more* の機能にページを逆向きにめくる機能を追加した同様のプログラムです。

2.3 インターネット：メール、ニュース、ウェブ

会津大学のワークステーションはネットワーク（インターネット）を通じて世界中の多くのコンピュータとつながっています。そして、いながらにして世界各地の遠く離れた人と情報を交換が可能です。これを実現する強力な手段がメールとネットニュース、そしてウェブです。メールとネットニュースは社会や周囲に与える影響が大きく、また、これらの電子的コミュニケーション手段は魅力があり楽しいものなので本業を忘れて熱中しがちなので、効率的に用いるよう、注意が必要です。

ネットニュースやウェブ掲示板の討論における表現は一部非常に過激であり、初心者が参加すると精神的に傷つくこともあります。そして、いながらにして世界各地の遠く離れた人と情報ならぬ事態も生じます。激しい討論にはできるだけ参加しないことが無難です。また参加する場合もそれなりの心構えをもって臨むようにしましょう。

メールやネットニュースが正常に動いているのは、それを支える多くの人たちの尽力であることを認識しましょう。将来コンピュータの専門家になるものとして、これらのサービスを享受するだけでなく、提供する側のことを思いやり、自分のできる範囲で手伝う、あるいは迷惑をかけないよう注意する態度をが望まれます。

2.3.1 メールとネットニュースに共通のマナー

1. アドレスは間違えないように慎重に入力する。
2. サブジェクトには「いわゆる全角文字」(16bit 文字, mime エンコードされていない日本語)や「半角カタカナ」を使わない。通常の ASCII 文字である「半角の英数字と記号」以外のものをサブジェクトに含める場合は、適切なエンコードをすること。
3. メールの内容が極端に長いときは、メールを複数に分割する。

2.3.2 電子メール

まずメールの宛て先を指定しなければなりません。この宛て先を文字どおりアドレスといいます。学内のユーザにメールを送るには、そのユーザの login 名のみをアドレスに指定すれば十分です。学外のユーザにメールを送るには、アドレスは一般に

ログイン名@ドメイン名

の形式となります。ドメイン名とは、簡単に言えば、そのユーザが属する計算機環境の名前

です。例えば、学外から foo さんに入力する時は `foo@u-aizu.ac.jp` とアドレスを指定します。この `u-aizu.ac.jp` がドメイン名は後ろから `.jp` は日本を、`ac` は教育研究機関を、`u-aizu` は会津大学をそれぞれ表す階層構造になっています。

2.3.2.2 メールを送る際のマナー

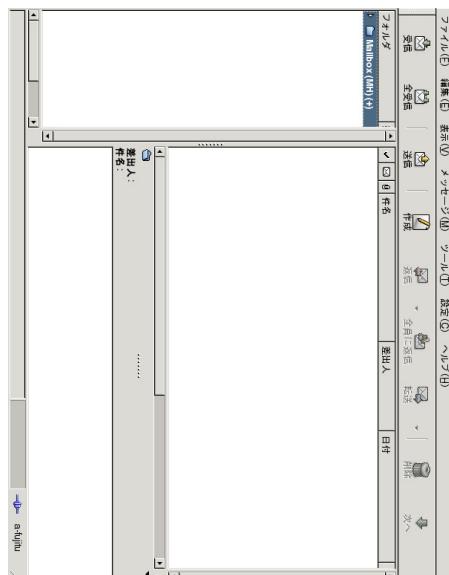
電子メールも通常の手紙と同じように、相手に対するマナーが必要です。特に電子メールならではのマナーがあります。

1. アドレスは間違えないように慎重に入力する。
 2. 内容は、1行が ascii 文字 80 字、日本語文字 40 字を超えないようにする。ascii 文字に換算して 60~70 文字程度を目安にすると良い。これを越えるようなら、リターンキーを押して適当に改行する。
 3. サブジェクトに(そのままでは)日本語を使わない。
 4. 相手が興味のないであろうメールを多人数に送らない。例えば政治的、宗教的話題は興味のない人にとっては迷惑である。
 5. 個人及び団体の中傷を不特定多数の人々にメールで送りつけない。
 6. 巨大なサイズの添付ファイルを送りつけない。
- もちろん、これに加えて通常の手紙の時のマナーを守るのは当然のことです。

2.3.3 メールの転送

大学のメールアドレス宛のメールを他のメールアドレスへ転送して携帯電話や Gmail 等で読むことができます。転送の設定の具体的な方法は 185 ページの 30.1.4 節「メールを転送する」や情報センターの説明を参考にしましょう。

<http://web-int.u-aizu.ac.jp/labs/istc/ipc/topic/mail-forward/mail-forward.html>



Sylpheed の設定は `$HOME/.sylpheed-2.0/` 以下の各種ファイルによって保持されます。Sylpheed の設定は付録の A(187 ページ) を参照してください。

• Thunderbird... Thunderbird は GUI のあるメールクライアントです。メールのメッセージを保存するファイルの形式は Sylpheed とは異なります。

- Mew on Emacs

2.3.5 ネットニュースなどの公開メッセージツール

世界中から情報が提供され、また世界中に情報を発信できるツールはインターネットならではの情報交換ツールです。その内容も、学問の先端分野に近いことから政治や行政、生活情報、ジョークに至るまで多岐に渡ります。これらを上手に使いこなせば、得られる情報の世界がぐっと広がります。

- 携帯電話で表示できないメールも送りつけられる可能性があります。
- 携帯電話で表示される可能性があります。

2.3.4 メールクライアント

演習室の端末で電子メールを取り扱うことのできるツールは以下の通りです。各メールクライアントで添付ファイルを利用できます。

- sylpheed... GUI のあるメールクライアントです。迷惑メール(スパム=SPAM) 対処の機能があります。メールのフォルダは `$HOME/Mail/inbox` です。メッセージの形式は 1 メッセージ 1 ファイルです。



2.3.1 メールを携帯電話に転送する際の注意

携帯電話にメールを転送する際の問題が起こります。

- 携帯電話の契約によっては料金がかかります。

- 巨大な画像ファイルを送りつけられることができます。
- メーリングリストの設定のミスなどでメールシステムのトラブルが発生し、大量のエラーメールを受け取る場合があります。

- 大量のエラーメールが携帯電話会社と会津大学間でやり取りされると会津大学と携帯電話会社の間の通信が遮断されることがあります。

ネットニュースはインターネットの初期から利用されてきた伝統的なツールですが、近年ではありません利用されていません。これは、Web掲示板、facebook、twitterなどのWebベースのツールなどに利用者が移って行ったからです。

2.3.5.1 学内ニュースのマナー

メールとは異なり、読み手が非常に多人数であることを認識する必要があります。

1. 個人及び団体への中傷をしない。
2. テーマに合ったことを書く。

3. 常に公正さに立脚し、個人的な感情からの非難攻撃をしない。
4. 後から取り消すことのないよう表現や事実関係を十分確認してから投稿する。できれば又聞きの情報を投稿することは避けた方がよい。

5. 著作権を侵害しない。雑誌などに掲載された他人の文章や写真などを借用しない。電子のニュースの世界であっても他人の文章をそのまま発表することは窃盗と同じである。
6. 法律に抵触するような文章や写真を投稿しないこと。

ニュースの記事は各ニュースグループに設定された一定の期間が過ぎれば、自動的にサーバから削除されます。必要な記事はファイルに保存しましょう。

2.3.5.2 ニュースクライアント

演習室の端末でネットニュースを取り扱うことのできるツールは以下の通りです。

- sylpheed … ニュース用のアカウントを設定すると利用できます。新規ユーザーに配布される標準環境では設定済みです。
- gnus (Netnews on Emacs)
- Thunderbird … Thunderbird でもネットニュースを読んだり、投稿することができます。

2.3.6 Web

検索 Web を利用した情報検索はとても便利です。その一方で、検索結果をそのまま授業のレポートに利用するような学生も増えてきています。情報を簡単に取得できることはとても良いことです。しかしながら、自分の頭でしっかり考えて自身のために理解することを省略してしまっては全くの無駄というものです。また、Web で検索して得られる情報が正しいものかどうかはやはり自分で判断する必要があります。

掲示板 ネットニュースに良く似たウェブ (Web) 掲示板というものがあります。現在では Web 掲示板や Blog, Twitter の利用者が増加し、ネットニュースの利用が激減しました。ネットニュースとは異なり、多くの Web 掲示板では匿名による記事の投稿ができます。したがって、無責任な気楽な記事が多い傾向にあります。表向きは匿名でもアクセスログ等から調査すれば「どの端末から投稿したか」や「誰が投稿したか」を特定できます。

2.3.6.1 ウェブ (Web) ブラウザ

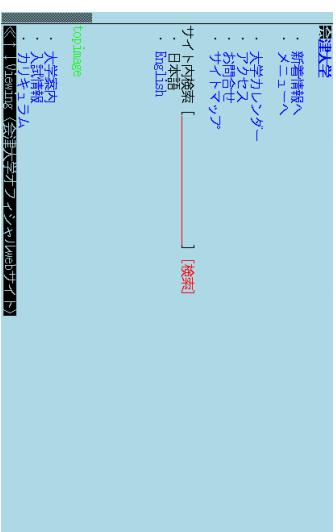
Web を利用することができるツールは以下に説明する通りです。

- Internet Explorer Microsoft 社の Windows 用のブラウザ。演習室の Solaris 端末では利用できません。

- Safari Apple 社のブラウザ。演習室の Solaris 端末では利用できません。

- firebox Firefox は Mozilla Project の Web browser です。

- w3m 端末内で動作するブラウザ。



- wget URL を指定してファイルを GET するツール。

第3章 ネットワーク, WiFi

演習室のワークステーションは、全てネットワークでつながっています。この節では、ネットワークに関する機能を幾つか説明します。

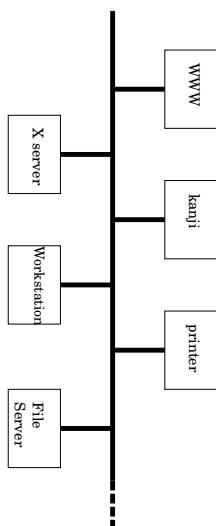


図 3.1. サーバとクライアント

コンピュータ等をネットワークに接続して利用するといろいろな利点があります。理解が簡単な例としてはネットワークプリンタの利用があります。ネットワークに接続していくない、いわゆる"スタンドアローン"と呼ばれる利用形態では印刷をするためには必ずプリンタをコンピュータに直接接続する必要があります。ネットワークに接続した場合にはプリンタを共有して印刷することが可能になります。印刷に限らず、様々なサービスを共有することで便利に利用できます。

ニュースサーバ 会津大学に送られてくるニュースを蓄えるシステムです。会津大学のニュースサーバは nhl.u-aizu.ac.jp です。
メールサーバ ユーザに送られてくるメールを一旦蓄えるシステムです。

```
% openldapsearch -x -LLL '(sendmailMTAaliasValue=s125xxxx@stdml4.u-aizu.ac.jp'
...  
sendmailMTAaliasValue: s125xxxx@stdml4.u-aizu.ac.jp  
...
```

となつていれば、s125XXXXさんのメールサーバは stdml4.u-aizu.ac.jp という名前の計算機です。メールアドレス s125XXXX@u-aizu.ac.jp 空に遅れられてきたメールは一旦 stdml4.u-aizu.ac.jp にあるメールスプールに蓄えられます。

```
rsh stdml4.u-aizu.ac.jp 'ls -l /var/mail/s125XXXX'  
-rw-rw---- 1 s125XXXX mail 12896879 Feb 28 05:46 /var/mail/s125XXXX
```

ファイルサーバ ユーザのホームディレクトリやコマンド等を物理的に保持するコンピュータです。ホームディレクトリは Network File System (= NFS)¹ 等のネットワークを介してディレクトリを共有するソフトウェアを利用することによって、どの端末からでも利用することができます。/bin/df -k コマンドを利用して以下の様に表示された場合は、

利用している端末に対してファイルサーバ stdfsv1 から /home/course という名前のディレクトリが提供されているという意味です。

また利用者のホームディレクトリもファイルサーバにより提供されています。このようふく一ムディレクトリやコマンドなどを集中管理することによって管理コストの削減と利用の透過性を実現しています。

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
rpool/ROOT/s11.1ga-8	409907016	6187500	403719516	2%	/
swap	48572468	136380	48436088	1%	/tmp
rpool/local	428923529	25204013	403719516	6%	/usr/local
profsv2:/vol/vol1/home1/a-fujitu	10485760	5264812	5220948	51%	/home/professor/a-fujitu
stdfsv1:/vol/vol12/course	157286400	53369428	103916972	34%	/home/course
...					

ウェブ (Web) サーバ 登録されたファイルやホームページを利用者に提供します。

- 会津大学の公式ウェブサーバ www.u-aizu.ac.jp (211.16.217.149)
- 学内向けウェブサーバ web-int.u-aizu.ac.jp (163.143.3.86)
 - 学外向けにウェブの標準ポートで公開しているサーバは <http://web-int.u-aizu.ac.jp/labs/istc/isc/Security/www-list.html> で確認できます。

漢字サーバ ユーザからの漢字変換の要求に応えて漢字を表示するサービスです。会津大学では負荷分散のために全ての演習室で動作させて、ユーザ毎にサーバを割り当てるよう設定しています。

NIS サーバ ユーザや接続ホストのデータを一括して管理します。² 2012 夏で利用を終了し、LDAP サーバ ユーザや接続ホストのデータを一括して管理します。³ 2012 秋から利用開始しました。

¹See 'man nfsd': NFS daemon

²See 'man nis': NIS daemon

³See 'man -s 5 ldap': ldap - LDAP as a naming repository

DNS サーバドメイン名を IP アドレス⁴に変換します⁵。会津大学の DNS サーバは eureka.u-aizu.ac.jp (= 163.143.1.100) です。

```
proxsv26{a-fujitu}61: nslookup u-aizu.ac.jp
Server: 163.143.1.100
Address: 163.143.1.100#53

Name: u-aizu.ac.jp
Address: 163.143.1.100
```

SSH サーバ • 学外からの ssh アクセスのためのサーバです。

- sshgate.u-aizu.ac.jp ここを通じて stdrv1 等の学内の計算機にもアクセスします。(Sunray 端末は表示装置なのでリモートアクセスできません。) stdrv1 では solaris11 がご利用でき、演習室 1,2,3,4 と同じツールを利用できます。
- 学内のはぼりの計算機で ssh サーバーが動作しています。solsv[1-4], stdj56[dc, stdsv1[1-8], lab[12]dc, から名前が始まる計算機です。stdj56[dc では Mac OS X が利用でき、ターミナル内で動作するアプリケーションに関しては演習室 5,6 と同じツールを利用できます。
 - 学外からの ssh アクセスを標準ポートで受け付ける研究室のサーバは <http://web-int.u-aizu.ac.jp/labs/istc/ipc/Security/form21.html> で確認できます。

Sunray サーバ 演習室 3,4 (std34sv[1-5]), CALL 教室 (stdsv[1-8]), 卒研用端末のためのサーバです。Sunray 端末で表示する X サーバが動作します。

3.1.1 サーバとクライアント

ネットワークを理解するために、サーバとクライアントという概念を説明しておきます。サーバというのは、ある特定の機能を提供する(=サービスを提供する)立場にあるもののことです。例えば、この後のネットワークプリンタでは、このプリンタ(プリンタを制御している計算機を示すこともあります。)はプリントサーバと呼ばれます。クライアントとは、サーバが提供する機能を使用する立場にあるもののことです。ネットワークプリンタでは、プリンタに印刷する内容を送り出すワークステーションが、クライアントとなります。

3.1.2 ネットワークプリンタの使い方

各演習室には、それぞれ 1,2 台のネットワークプリンタが設置してあります。既に LATEX の演習で試した人もいるかも知れませんが、これらのプリンタを使って印刷することが出来ます。ネットワークプリンタは直にイーサネットでネットワークに接続されています。ワークステーションから印刷の以来を受けて印刷のサービスを提供するのでプリントサーバとも呼ばれます。この利用の仕方は Windows パーソナルコンピュータや Macintosh に直接プリント

タを USB 等で接続する方法とは異なります。最近は家庭内でも LAN を利用することが多くなり、ネットワーク対応のプリンタがホームユース向けに販売されています。

3.1.2.1 印刷コマンド

印刷に関連するコマンドは、次のようなものがあります。

コマンド	文法	機能
lpr	lpr -P <i>printename</i> [<i>option</i>] [<i>filename</i>]	<i>filename</i> で指定された内容を印刷する
lpq	lpq -P <i>printename</i>	<i>printename</i> で指定されるプリンタのキーを調べる

この *printename* には、ネットワーク上で定義されたプリンタ名を書きます。演習室 1 なら、stdlpr1, stdlpr2, 演習室 2 なら, std2pr1, std2pr2 が各プリンタ名です。

印刷が指示されると、印刷内容は、まずプリントサーバのプリントキュー(印刷の順番待ちのことです)に登録されます。プリンタが空いていれば、ただちに印刷が開始されます。ただし、ネットワークプリンタは、大勢で共有して使うものなので、いつも空いているとは限りません。その場合は、印刷する情報は、印刷の要求のあった順にキューに溜められます。これを、キューに印刷 **job** を登録するという言い方をすることもあります。キューに溜められた job は、順に処理され、印刷されます。
指定したプリンタのプリントキュー(印刷の順番待ち)の状態を調べるのが、**lpq** というコマンドです。下は、lpq を実行してみたところです。

```
s0@000000@stdrv1121% lpq -Pstdlpr1
Processing Job s1010000 from Host stdrv1121.
Rank   Owner    Job   Files          Total Size
active s1010000   101  kada1.ps      944 bytes
1st    s1010001   102  kada12.ps     1036 bytes
2nd    kikuchi    103  text3.ps      1164 bytes
3rd    kikuchi    104  sample.ps     1032 bytes
```

出力の意味は、左から順に説明すると、Rank の欄は、印刷の順番を表します。active となっているのは、現在印刷中という意味です。Owner は、その印刷の指示を出したユーザのユーザ名で、次の job という位置に書いてある番号は、印刷の各 job に対して、システム側から一意に付けられた登録番号です。この job 番号は次の lprm を実行するときに必要になります。Files は、文字通り印刷中のファイルのファイル名、Total Size は、印刷内容の大きさを表します。当たり前ですが、同じ種類のファイルの場合、Total Size の大きい方が印刷に時間がかかります。lpq の指定を間違えたり、プリンタが混み合っていたりして、印刷 job を取り消したいことがあります。そのときには、**lprm** というコマンドです。例えば、上の例では、s1010001 さんが、kada12.ps の印刷を止めたいと思った場合、

⁴See 'man ip': Internet Protocol

⁵See 'man named': internet domain name server

本当に、サーバとクライアントというのは、ネットワーク上に限った話しではありません。

ビジネスの世界ではクライアントというのは、顧客という意味です。

```
lprm -Pstdlpr1 102
```

と入力します。lpq, lprm の 2 つの命令を使いこなせない人は印刷に関係するトラブルを

理できないという理由で『印刷をする資格がない』という大変厳しい言われ方をしては仕方ありません。自分自身で印刷することは学生にとって必要なことでしようから、(lprを含めてもたったの3つしかない) lpr, lpd, lprm の使用方法については曖昧な理解のまま放置せず、しっかりと身についてください。

`lpr` のオプションについて `lpr` 命令に使用するオプションはプリンタの機種によって異なることがあります。

対象機種	オプション	機能
LP-S4200PS, B841	-o Duplex=DuplexNoTumble -o Duplex=DuplexTumble	両面印刷(オートレバー) 両面印刷2(ランドスケープ)

演習室1,2 CALL1,2 教室ハードウェア実験室1,2,3,4 にあるプリンタはOKI B841です。これらのプリンタはポストスクリプト(=PS)専用プリンタなので、PS言語で書かれた画像データ以外のファイル(テキスト形式、DVI形式、PDF形式、ワープロの文書ファイル、PS以外の画像ファイル)等はすべて一旦、PS形式のファイルに変換(=コンバート)した後に印刷可能になります。PS形式のファイルに変換するためには元のデータの形式に合った適切なソフトウェアを利用します。以下にPS形式に変換する例を示します。

上で説明した Solaris11 や MacOS の `/usr/bin/lpr` は CUPS と呼ばれるプログラムです。他に `/usr/local/LPRng/bin/lpr` があり、以下の様にオプションが異ります。

対象機種	オプション	機能
LP-S4200PS, B841	-cduplex -Cduplex,tumble	両面印刷(ボートレバ) 両面印刷(ランドスケープ)

テキストファイルの印刷
テキストファイルを postscript に変換するコマンドとして、a2ps や k2ps があります。これは、日本語にも対応しています。a2ps は UTF-8 対応で k2ps は EUC 対応です。両ツールも古いので近い将来に使用不可になるか、新しいツールに置き換わるかもしれません。

- 元キリストファイル sample.txt を元にキリストクリアト形式のファイル sample.ps を生成する。
nkf -e sample.txt | k2ps > sample.ps
または a2ps sample.txt > sample.ps
 - 使い方を表示する。
k2ps -help

PDF ファイルの印刷

- `pd2ps9` を利用し PS 形式のファイルを生成します。以下は変換の例です。

1

- sample.pdf を元に sample.ps を生成する
pdf2ps sample.pdf

- pdfstudio を利用して印刷します。
- xpdf や evince を利用して印刷します。
- Acrobat Reader を利用して印刷します。

その他の画像ファイルの場合
convert¹⁰ を利用します。

- sample.tiff を元に sample.ps を生成する
convert sample.tiff sample.ps

convert sample.tif sample.ps
sample.ps が sample.eps へと変換されます。

- sample.jpg を元に sample.eps を生成する。
bmeps sample.jpg sample.eps
 - sample.jpg を元に sample.pdf を生成する。
bmeps sample.jpg sample.pdf

⁸See ‘man dvips’: convert a TeX DVI file to PostScript
⁹See ‘man pdflps’: pdflps - Ghostscript PDF to PostScript to
See ‘man convert’: converts an input file using one image for
ii結果、ソフトウェアはナマモノです。

⁸See 'man dvips': convert a TeX DVI file to PostScript

⁸See 'man dvips': convert a TeX file.

See 'man pdf2ps': pdf2ps - Ghostscript PDF to PostScript translator
See 'man convert': converts an input file using one image format to another.
結局、ソフトウェアはナマモノです。

「結局、ソフトウェアはナマモノです

3.1.2.2 印刷に便利なコマンド

テキストファイルをそのまま印刷すると、紙の左上の端からびっしり印刷されて、見づらくなってしまうときにも不便です。また、何のファイルを印刷したかは自分で管理しなければなりませんし、バラバラになつたら収集がつきません。そこで、次に挙げるコマンドを使うと、事態が改善されます。

3.1.2.2.1 pr このようなプログラムを、フォーマッタといいます。**pr**は、印刷出力の各ページに、日付やファイル名、ページ番号等をつけてくれるプログラムです。また、**pr**は豊富なオプションを持っており、多彩な印刷フォーマットを指定できます。各オプションについては、オンラインマニュアルを参照して下さい。**pr**は、ファイル名を直接指定してもいいですし、またファイルとしても働きます。これを利用すると、印刷出力がずいぶん改善されます。

3.1.3 他のワークステーションを使う

ネットワーク環境では、離れたコンピュータを、あたかも自分の目の前にあるかのごとくに使うことができます。

3.1.3.1 リモートログイン

一つのコンピュータから、ネットワークを介して別のコンピュータにログインすることを、リモートログインといいます。リモートログインをするコマンドは、**rlogin**です。

```
rlogin remotehost [-lusername]
```

remotehost は、リモートログインしたい先のコンピュータ名です。ユーザ名は、デフォルトでは現在のユーザ名が用いられますが、-l オプションで変えることができます。通常は、**rlogin** を実行すると、パスワードの入力が求められます。正しく入力するとリモートログインが完了します。パスワードを間違えると、ユーザ名からの再入力が求められます。入力を 5 回間違える、または 60 秒以内に正しくログインできないと、リモートログイン失敗となり、接続が解除されます。また、気が変わつてリモートログインをやめたいときは、60 秒間待つか、ユーザ名入力のときに、C-d を入力します。

3.1.3.2 rlogin した先でやってはいけないこと

リモートログイン先で他のユーザが使ってない場合は問題がありませんが、大抵の場合は誰か他の人が使っています。他の利用者に迷惑を掛けないようにしなければなりません。特に、次のことはしてはいけません。

1. **xinit** リモートログイン先で、X ウィンドウシステムを破壊し、使えないするおそれがあります。

2. **lock** コンソールをロックするプログラムです。本来は、ちょっと席を外すときに一時的に他人が使えないようにする目的のものですか、リモートログイン先で実行するとそのコンピュータが使えなくなってしまいます。

3. その他 X ウィンドウに関するコマンドあくまでも、リモートログイン先のコンピュータを使っている訳ですから、X ウィンドウに関するコマンドは、リモートログイン先で働きます。自分の端末にウィンドウを開くには、それなりの設定が必要です。

その他、不要なファイル操作(ファイルシステムの効率が落ちます)や不要な重い処理(他の処理が遅くなり、場合によってはシステムがストップします)などを行なってはいけません。

3.1.4 信頼したホスト間のリモート利用

正しく管理され安全であると確信したホスト間ではホームディレクトリに .rhosts という名前のファイルを置き、その中にホスト名とユーザ名を記述することによって remote access の認証を省略することができます。この設定により、

```
rsh.stdrsvltop
```

の様に指定した計算機に対して命令を実行させることができます。

3.1.5 X client プログラム

Solaris 端末の X Window System を利用している場合は(Tectia 社の ssh なので)

```
ssh.a-fujitu@aplia1a121.xeyes
```

Mac OS X 端末を利用している場合は(OpenSSH なので)

```
ssh_a-fujitu@aplia1a121_xeyes
```

の様にして、指定した計算機の X client プログラムを動作させることができます。

3.1.6 VNC

VNC(=Virtual Network Computing)を利用するとデスクトップをリモートで使用することができます。付録の F.1.2 を参照する。

3.1.7 学外から学内へのログイン

会津大学では学外からの rsh, rlogin, telnet, ftpなどを用いたアクセスは制限されています。これらの接続では認証のパスワードや通信の内容が暗号化されないで元のテキストのままネットワークを流れるので、盗聴などによる情報の不正取得に対する手段がありません。そこで学外からのアクセスは暗号化された通信を行う ssh や sftp を利用します。Mac OS X には ssh クライアント(Open SSH)が標準で導入されています。ターミナル内で

```
ssh s125XXX@sshgate.u-aizu.ac.jp
```

- ains-air
EAP-TLS 方式のサービスです。ユーザ証明書を PC、smart phone 等にインストールし利用します。ユーザ証明書を利用してユーザ認証が行われるので利用時に AINS のユーザ ID とパスワードの入力を求められません。

現在のところ Windows には ssh クライアントは標準では導入されていません。適切な ssh クライアントをインストールして利用できます。以下のドキュメントに例があるので参考できます。

<http://web-int.u-aizu.ac.jp/labs/istc/ ipc/topic/sshgate/sshgate.html>
sshgate は使用について以下の制限があります。

- プリンタへの印刷ができない。
- cron を利用できない。
- 一定時間以上接続を放置すると自動切断する。

3.1.7.1 SSH トンネル

ssh を利用したトンネルを利用した通信を行うと sshgate 経由で指定したポート番号の通信ができます。場合は以下の通りになります。まず端末を 2 つ用意し、片方で以下を実行しログインします。

```
ssh -L 9999:stdrsrv1.u-aizu.ac.jp:22 s125XXX@sshgate.u-aizu.ac.jp
```

これによりローカルホストの 9999 番ポートから stdrsrv1.u-aizu.ac.jp の 22 番ポート (ssh サーバ) へのトンネルが sshgate.u-aizu.ac.jp により提供されます。次に、2 つ目の端末で

```
ssh -p 9999-X_s125XXX@localhost
```

を実行するとトンネル経由で stdrsrv1.u-aizu.ac.jp が提供する ssh サーバにアクセスできます。

学内の端末に導入されている ssh のクライアントプログラムは Tectia 社のもので openssh とは若干オプションが異なります。特に (-X) は X11 フォワーディングを停止します。Tectia 社の ssh のクライアントプログラムの場合は、

```
ssh -h
```

としてオプションを確認できます。

3.2 無線 LAN サービス

会津大学では構内の殆どで無線 LAN を利用して、個人のパソコンやタブレットなどの WiFi 端末をネットワークに接続することができます。
現在、会津大学の情報センターが提供している無線 LAN のサービスの SSID は ains-air と ains-ttls です。

- ains-air
EAP-TLS 方式のサービスです。こちらはユーザ証明書をインストールする必要がありません。その代わりに ains-ttls を利用する時には AINS のユーザ ID とパスワードの入力を求められます。

学生は 2 枚までユーザ証明書を利用できます。

利用する場合には方法センターの利用ガイド <http://web-int.u-aizu.ac.jp/labs/istc/ ipc/topic/wlan/eap-ttls/index.html> を参考にしてください。

3.2.1 認証ネットワーク

http://web-int.u-aizu.ac.jp/labs/istc/ ipc/topic/Authenticated_Network/auth_net.html
中講義室や、小講義室、会議室などで有線接続をした場合に、認証ネットワーク、と呼ばれるサービスを利用してネットワークの利用を許可します。有線のケーブルを差し込んでウェブブラウザを立ち上げると認証のためのページが現れます。ここで AINS のユーザの ID とパスワードを入力して認証します。

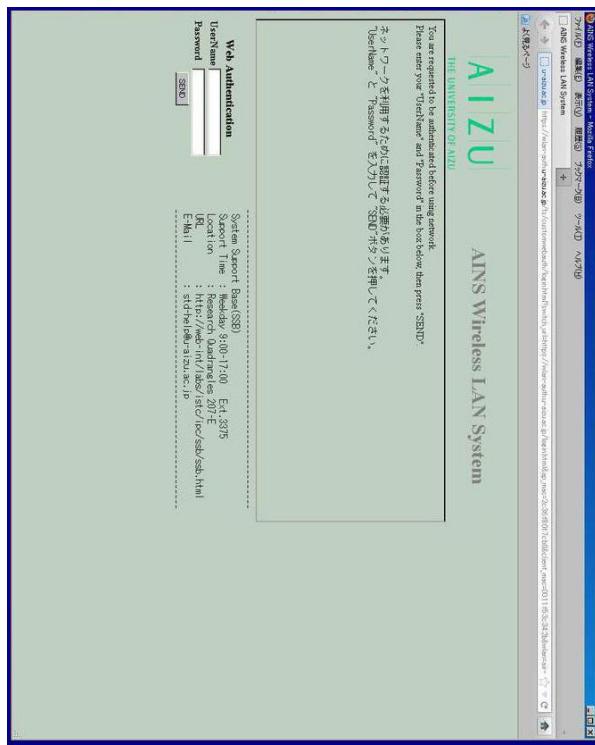


図3.2 認証ネットワーク

第4章 文字、文字入力、正規表現

場合は、8bits で表される英数字とカタカナにつけられたものをしますが、最近では JIS コードといつた場合は、この JIS 漢字・非漢字コードをさせことが多いようです。ところで、この JIS コードは、そのままでは 1 文字分の JIS コード(16bits)なのか、2 文字分の ascii コード(8bits)なのか区別がつきません。そこで、実際にはこの区別のためにくつこの異なる方法がとられています。それらの方法の中で現在良く利用されているものは以下の 5 種類です。

- 7-bit JIS (または単に JIS) — メールで利用される文字コードです。

- shifted-JIS (SJIS) — Windows で利用してきた文字コードです。
- EUC — 拡張 UNIX コード (Extended UNIX Code) です。
- UTF-8 — UTF-8 コード (8-bit UCS Transformation Format) です。
UCS=Universal multi-octet coded Character Set の略。
- UTF-16 — UTF-16 コード (16-bit UCS Transformation Format) です。

本学のワークステーションで標準として用いられているのは、UTF-8 コードです。Windows2000 以降の Windows では SJIS に加えて、Unicode (UTF-16) が使用されていて、Mac OS X では Unicode (UTF-8-Mac) が使用されています。今後は Unicode が主流になると予想されます。

Emacs を起動したときに表示される scratch バッファは UTF-8 コードになっています。シェルプログラムや C のプログラム等は UTF-8 コードで書かなければなりません。Emacs で新規ファイルを開いた場合は UTF-8 コードに設定されます。

で表示される様に 2 進数で 8 桁 (=binary 8 bits) で表される文字を ascii 文字と呼びます。'A' は 2 進数では 01000001, 10 進数では 65, 16 進数では 41 と表記されます。2 進数の 8 桁は 2 の 8 乗 (=256) 通りの 0 か 1 の並びのことなので、256 種類の文字を表すことができます。

4.1.2 漢字コードの変換ツール
複数の漢字コードが世の中で利用されていることで混乱することもあるので注意が必要です。

- emacs
Emacs は 7-bit JIS, shifted-JIS, EUC, UTF-8 あるいは UTF-16 の 5 種類に対応しています。emacs では文字コードを変更してファイルを保存できます。C-x RET f をタイプするとミニバッファで Coding system for saving file (default utf-8): と、文字コードの指定を求されます。
- nkf
漢字コードを変換する nkf (Network Kanji code conversion Filter) という便利なツールがあるのです。必要な場合には利用します。
- nkf の利用例を以下に紹介します。
- JIS から UTF-8 への変換

```
nkf -w sample.jis > sample.utf8
```

英数字を表現するには 8bits(=1Byte) で十分ですが、漢字などは数千字以上あるので日常的によく使われるものに限っても、8bits では足りません。そこで、漢字や仮名を表すのに、16bits(2 Bytes) の 2 進数が使われます。この数字は JIS(日本工業規格) で決まっていて、これを JIS 漢字・非漢字コード、あるいは単に JIS コードと呼ばれます。厳密には、単に JIS コードといった

- nkf -e sample.jis > sample.euc
- EUC から shifted-JIS への変換

- nkf -s sample.euc > sample.sjis

- Windows,DOS などの改行コードを UNIX の改行コードに変換

- nkf -d sample.dos > sample.unix

- UNIX の改行コードを Windows,DOS などの改行コードに変換

- nkf -c sample.unix > sample.dos

- shifted-JIS でかかれた Windows のファイルを UNIX のファイルに変換

- nkf -d -e sample.dos-sjis > sample.euc-unix

Windows,dos などの改行コードは (0d cr)(0a nl) です。UNIX の改行コードは (0a nl) のみです。以下の 2 つの命令の出力を比較すると良いでしょう。

- echo "test" | od -x
- echo "test" | nkf -c | od -x
- echo "test" | od -a
- echo "test" | nkf -c | od -a

4.1.2.1 かな漢字変換

日本語文字の入力方法では、漢字や仮名のような日本語を、どのように入力すればよいですか。最も単純な方法は、漢字コード表を片手に、JIS コードや EUC コードを直接入力することです。しかしこれでは能率が悪すぎます。そこでいくつかの方法が考えられましたが、今のところもっともポピュラーなのは、かな漢字変換と呼ばれる方法です。

かな漢字変換とは、まずローマ字表記の仮名をひらがなに変換したり、あるいは直接仮名を入力でき るキー ボードを使つたりして仮名を読みにつつ漢字をコンピュータ内の辞書から調べ、その漢字を表示するものです。

まずユーザはアプリケーションプログラムを通じてキーボードを利用してアルファベットや平板名を入力します。アプリケーションプログラムは入力された仮名を、かな漢字変換クライアントプログラムに渡します。かな漢字変換クライアントプログラムはかな漢字変換サーバにアクセスして漢字の候補を取り、アプリケーションプログラムへ返します。アプリケーションプログラムによって表示された漢字をユーザが確認をして満足すれば、かな漢字変換クライアントプログラムへ確定を送るためのキーを入力をして確定されます。不満足な場合は次の候補を要求するためのキーを入力をしてかな漢字変換クライアントプログラムは次の漢字の候補をかな漢字変換サーバから受け取ります。(図 4.1)。

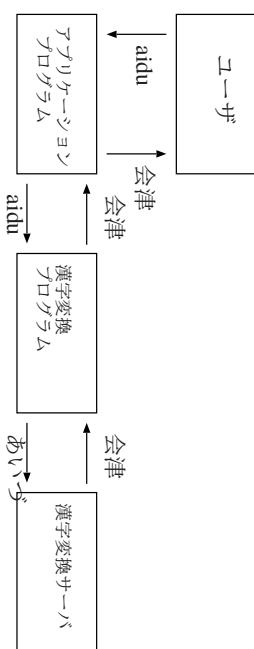


図 4.1. 漢字変換の動作

この方式をバックエンド方式と呼び、利用するアプリケーション自体がかな漢字変換クライアントプログラムに対応している必要があります。アプリケーションプログラムがかな漢字変換クライアントプログラムを利用する方法には X Window System で伝統的な XIM(X Input Method) とプラットフォームに依存しない IIMF(Internet/Intranet Input Method Framework)、その他に様々なプロジェクトがあります。

かな漢字変換サーバ(Language Engine=LE とも呼ばれます。) はいろいろな種類があります。

- wnn — オムロンソフト社の製品版 Wnn8 や FreeWnn 等があります。Android では iWin IME が動作しています。(うんぬ、わたしのなまはなかのです、の略)
- Canna 「かんな」は、UNIX 系 OS で動く、クライアント・サーバー型の日本語入力システムです。
 - SKK (=Simple Kana to Kanji conversion program)
Emacs Lisp で実装された。 Simple Kana to Kanji conversion program
 - ATOK — Justsystem の製品。 Solaris 11 で日本語入力のために利用しています。 Windows, Mac, Linux 用の製品があります。
 - 日本語支援ツール(かな漢字変換クライアント)もいろいろな種類があります。
 - kiinput2 — OpenOffice, firefox や tigf など多くのアプリケーションで利用可能。 XIM を利用する代表的なプログラム。 Canna や Wnn と組み合わせて使用できる。
 - iim Internet/Intranet Input Method
 - ms-im — Windows に付属。 Input Method Editor

- ・ことえり — Mac OS X に付属。

Solaris 端末では X Window System の日本語に対応したいいくつかのアプリケーションに対し `kiinput2` を利用して日本語入力を行います。あらかじめ terminal 等の端末内で `kiinput2 &` を実行して動作させます。

- ・Google 日本語入力 Google が提供する日本語入力支援ツールです。

4.1.3 Emacs での日本語入力

4.1.3.1 Solaris 端末の Emacs

Solaris 端末の Emacs で、`C-\` と入力すると、`quail+kkc` と呼ばれる日本語入力機能を利用できます。ステータスバー(画面下の黒い一行)の左側のほうに、[A a u] と表示されます。この状態でローマ字表記の日本語を入力すると、直ちにひらがなに変換され、下線が現れます。この状態でスペースキーを押すと、かな部分が漢字に変換されます。変換結果が意図したものと違う場合は、下に示したキー操作で、正しい結果が得られるまで変換を繰り返します。

<code>SPC, C-n</code>	次候補を表示する
<code>C-p</code>	前候補を表示する
<code>C-i</code>	カタカナにする
<code>RTN</code>	モード切替え(ローマ字、半角英数字、全角英数字)
<code>Shift-h</code>	ひらがなにする
<code>Shift-k</code>	文節区切りを縮める
<code>Shift-l</code>	F1 あるいは ALT-1 F2 あるいは ALT-2
<code>C-c</code>	文節区切りを伸ばす
<code>C-g</code>	後の文節へ移動
<code>qq</code>	[左矢印], C-b [右矢印], C-f
	前の文節へ移動
	先頭文節にカーソルを移動
	最終文節にカーソルを移動
	記号入力(一覧が表示される)

ここで、`SPC` とはスペースキーのこと、`RTN` とはリターンキーのことです。

4.1.3.2 Mac OS X 端末の Emacs

Mac OS X 端末の Emacs で、`C-\` と入力すると、SKK のモードになります。SKK では以下のように動作します。

<code>SPC, C-n</code>	次候補を表示する
<code>C-p</code>	前節を表示する
<code>C-i</code>	文節の長さを短くする
<code>C-o</code>	文節の長さを長くする
<code>C-l RTN</code>	すべての変換を確定する
<code>C-g</code>	すべての変換モードを中止する

4.1.4 Input Method を利用した日本語入力

Emacs には専用の日本語入力の仕掛けが用意されていますが、その他のアプリケーションでは Input Method を利用して日本語を入力します。Solaris 端末では `kinput2(+gamma)` と `iim(+atok)` を利用します。Mac OS X 端末では X window 用の Mac-UIM(`Shift+Space`) と Aqua インターフェース用のことえり(Command+Space) を利用します。

4.1.4.1 kiinput2 のキー操作

Solaris 端末では X Window System の日本語に対応したいいくつかのアプリケーションに対し `kiinput2` を利用して日本語入力を行います。あらかじめ terminal 等の端末内で `kiinput2 &` を実行して動作させます。

<code>shift-SPC</code>	開始、終了
<code>C-w, SPC</code>	漢字に変換
<code>SPC, C-n</code>	次候補を表示する
<code>C-p</code>	前候補を表示する
<code>C-l, RTN</code>	確定
<code>TAB</code>	モード切替え(ローマ字、半角英数字、全角英数字)
<code>F1</code>	カタカナにする
<code>F2</code>	ひらがなにする
<code>shift-[左矢印], C-i</code>	文節区切りを縮める
<code>shift-[右矢印], C-o</code>	文節区切りを伸ばす
<code>[左矢印], C-b</code>	後の文節へ移動
<code>[右矢印], C-f</code>	前の文節へ移動
<code>C-a</code>	先頭文節にカーソルを移動
<code>C-e</code>	最終文節にカーソルを移動
<code>shift-ESC</code>	記号入力(一覧が表示される)

4.1.4.2 iim の利用

X Window System の日本語に対応した様々なアプリケーションに対して iim が日本語入力機能を提供します。Solaris 端末ではログイン時に動作しています。何かの事情で動作していない場合には terminal emulator(gnome-terminal) 内で `iimstart` を実行して動作させます。(終了は `iimstop` です。) Gnome Desktop System を利用していれば Gnome Panel 内に `En` ボタンが表示されます。iim の日本語入力では Atok を利用できます。図(4.2)が



図 4.2. iimsatt

C-SPC	開始、終了 漢字に変換
SPC	次候補を表示する
C-p	全角ローマ字にする
C-o	半角カタカナ、ローマ字にする
C-n, RTN	確定
F7 あるいは C-i	カタカナにする
F6 あるいは C-u	ひらがなにする
[左矢印], C-k	文節区切りを縮める
[右矢印], C-l	文節区切りを伸ばす
[左矢印], C-b	後の文節へ移動
[右矢印], C-f	前の文節へ移動

4.1.4.3 ことえり

Mac OS X 端末では Aqua 対応のアプリケーションは「ことえり」を利用して日本語入力可能で、X Window System 上で動作するアプリケーションの場合は「ことえり」を利用して直接、日本語を入力することはできません。

- Aqua 対応のアプリケーション
- Mac OS X 端末の Emacs や Firefox, Terminal, MS-Office など
- X Window System 上で動作するアプリケーション
- Mac OS X 端末の Sylpheed や Tgif, xfig など

SPC, C-n	次候補を表示する
C-p	半角ローマ字にする
C-l	全角ローマ字にする
C-k	カタカナにする
C-j	ひらがなにする
C-f	直前の文節にカーソルを移動する
C-b	直前の文節にカーソルを移動する
C-i	変換対象部分の長さを短くする
C-o	変換対象部分の長さを長くする
RTN	すべての変換を確定する

4.1.4.4 Gnome-terminal への日本語入力

Gnome-terminal は iim 対応です。起動時は UTF-8 になっています。
次に Input Method は Internet/Intranet Input Method に設定しています。

したがって **[Ja]** の状態で C-SPC を押すと AtoK を利用して日本語入力が可能になります。

Gnome Desktop System ではログイン時の Language を Japanese UTF-8 にした場合にのみ、File Browser が、Rename で日本語を使用できる等の点で、UTF-8 ファイル名に対応しています。しかし、Solaris 端末で使用できる Emacs22 は、日本語ファイル名でファイルを保存すると不正な Unicode になる等、UTF-8 ファイル名を正しく扱えない様です。gedit は問題なく動作します。

Mac OS X では Finder が UTF-8 ファイル名に対応しています。Mac OS X 端末に導入されている Emacs は UTF-8 ファイル名を使用できます。

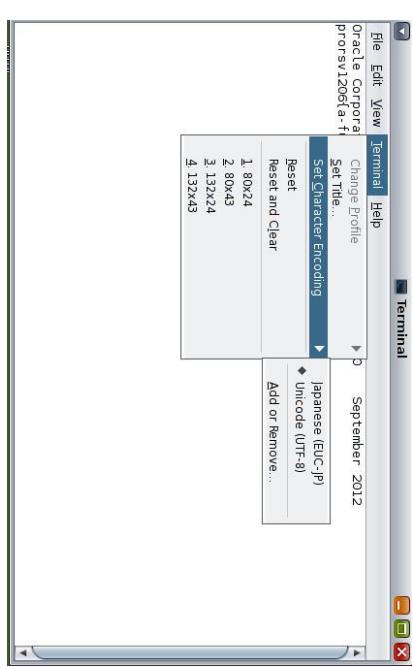


図 4.3. gnome-terminal-set-character-encoding

c	メタキャラクタでない文字 c はその文字自身を表す
\m	メタキャラクタ m の特別な意味を消し、その文字自身を表す
\n	行頭を表す
\r	行末を表す
\$	任意の一文字を表す
.	任意の一文字を表す
[c1c2...cn]	文字集合 c1...cn 中の任意の 1 文字を表す。かつこの中ではメタキャラクタは評価されないが、特別に - がメタキャラクタになる。- を普通の文字として指定するには --- とハイフンを 3 つ連ねると - から -までの文字を表すので、結局 - を表すことになる
[c1..c2]	c1 から c2 までの範囲の文字中の任意の 1 文字を表す。(例えば [a..z])
[r1..rn]	r1..rn のどれでもない任意の 1 文字を表す。
\N	N 番目のタグつき表現(後述)がマッチした文字列を表す。N は数字。
r*	正規表現 r の 0 回以上の繰り返しを表す。
r1r2	正規表現 r1 が正規表現 r2 が続く並びを表す。
\(\r\)	タグつき正規表現 r(正規表現のグループ化)を表す。ネストも可能。

表 4.1. 正規表現

Windows PC や Mac 等では日本語フォルダ名などが当然の様に使用されています。しかし、演習室の様にマルチプラットフォーム環境で共通のホームディレクトリを使用する場合は、現在の状況では ASCII 文字を使用したファイル名を使うことで、上で説明した様な混乱を避けることができます。

4.2 正規表現 Regular Expression

正規表現とは文字列を表現するルールを決めたものです。コンピュータのプログラムや通信で文字列を扱う上で一定の規約に従めておけば、そのルールに従って動作する仕組みを提供したり、利用したりすることが出来て非常に便利です。正規表現は規約そのものですが、これは各種プログラミング言語で実装、あるいはライブラリが提供されて利用されています。

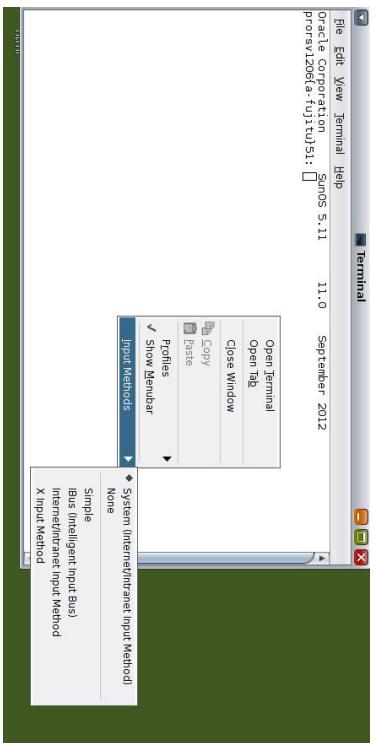


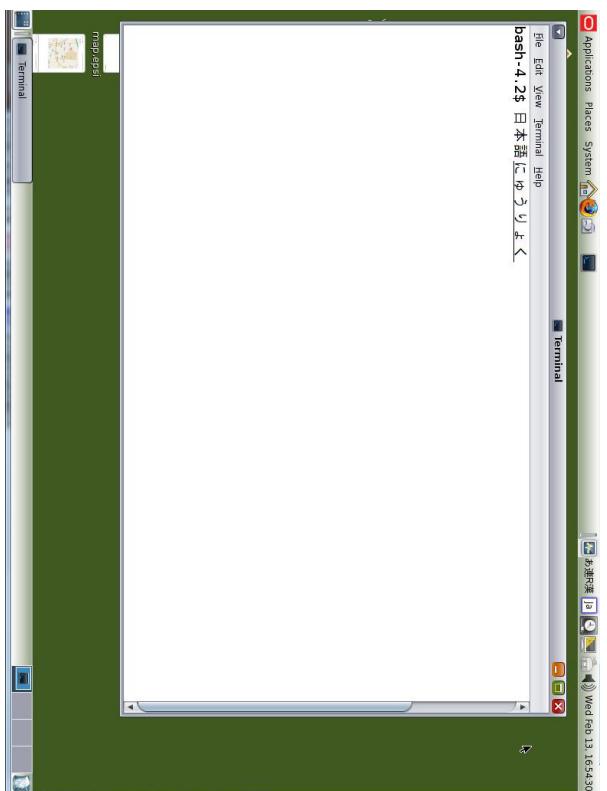
図 4.4. gnome-terminal-select-input-methods

4.2.1 正規表現のポイント

- 正規表現 = 通常の文字 + メタキャラクタ
- 正規表現で表現されるのは文字の集合
- 正規表現で表現された文字の集合にある文字列が含まれることを match したという。
- 最も長い文字列に match する。
- `(.*)` という文字列は、`f1(f2(f3))` に対しては、`(f2(f3))` にマッチする。

```
%_cat temp1
f1(f2(f3))
%_sed -e "s/(.)/a/" temp1
f1(f2(f3))
%_sed -e "s/(.*)/a/" temp1
f1a
```

図 4.5. gnome-terminal-nihongo



4.3 タグつき表現

以下の様にタグつき表現を使用できます。

- `\.pp`
(ピリオド) に `pp` が続きます。`\(パックスラッシュ)` によって、.(ピリオド) のメタキャラクタとしての働きがエスケープされています。
- `\w*`
1つ以上の空白の繰り返しを表します。

4.2.2 正規表現の例

- `x.y`
`x` と `y` の間に任意の 1 文字が入ります。

```
%_echo abcdefg | sed 's/(abc)\(\def\ef\g\)/\2-\1/'
defg-abc
```

- 例 3 大文字に変換(\uだと最初の 1 文字のみ大文字にする)

```
%_echo abcdefg_|sed_s/^(abc\(\defg\))/\^2-\U1/'  
defg-ABCDEFG
```

- 例 4 小文字に変換 (\l だと最初の 1 文字のみ大文字にする)

```
%_echo_ABCDEFGL_|sed_s/^(\ABC\(\DEFGL\))/\^2-\L\1/'  
DEFG-abcdef
```

4.4 sed

sed は Stream EDitor の略です。いろいろな使い方がありますが、ここでは、よく用いる 1 行コマンドとしての使い方について、文字列の変換に限定して示します。

```
sed_e "s/文字列 1/文字列 2/g" ファイル名
```

すると、ファイル名で示されたファイルの中の文字の中で、文字列 1 に該当するものが文字列 2 に変換されます。“で囲まれた中の s は代入 (=substitute) の意味です。/ は区切り文字としてしばしば使われますが、他の文字でも構いません。s の直後の文字が区切り文字となります。最後の g は全て変換するという意味のフラグです。sed は行ごとに処理しますので、g がないと、各行の中で最初に match しただけが変換されます。

区切り文字の例

```
%_sed_e "s/(.)/a/" temp1  
f1(f2(f3))  
%_sed_e "s+(.)+a+" temp1  
f1(f2(f3))
```

4.4.1 例

- int を unsigned int にします。

```
sed_e "s/int/unsigned.int/" ファイル名
```

```
%_cat temp4  
int.c;  
%_sed_e "s/int/unsigned.int/" temp4  
unsigned.int.c;
```

- 行頭のピリオド(.) 全てをカンマ(,) に変換します。

```
%_sed_e "s/^\.,/" ファイル名
```

4.5 パターンマッチ

文字列置換以外にも正規表現は利用されます。grep コマンドは **-e** オプションで正規表現を利用した文字列検索ができます。以下の例はファイル名が a,b または c から始まり t で終るものリストします。

```
%_ls | grep_e '^abc[.] .*t$'
```

perl や awk を始めとして様々なプログラミング言語で正規表現を利用したパターンマッチをプログラムの制御に利用できます。

第5章 ファイル管理, 情報収集

誤って削除してしまったファイルを元に戻すことは不可能です。重要なファイルなどは自分の手で、USBメモリー等の外部記憶メディアにバックアップとして保存しておくことを勧めます。

File BrowserやFinder等のGUIを利用してファイル等をゴミ箱フォルダ(\$HOME/.Trash/)に移動することができます。「ゴミ箱フォルダに移動させる」ことは削除ではなく、あくまでも「移動」です。rm コマンドはゴミ箱フォルダに移動させる命令ではありません。

5.1 ファイル管理

ファイルの管理について説明します。

5.1.1 ファイル名とワイルドカード

UNIXのコマンドにファイル名を与える場合、ワイルドカードという仕組みを利用できます。ワイルドカードとは、任意の文字、文字列に一致する特別の文字のことです。以下で説明するように正規表現とは異なるので混同しない様にしましょう。

- * 任意の文字列に一致するワイルドカード
- ? 任意の一文字に一致するワイルドカード

例:

*.tex	拡張子に.tex をもつ全てのファイル
kadai?.tex	kadai1.tex, kadai2.tex など、?のところに任意の一文字を入れた全てのファイル
kadai[1234].tex	kadai1.tex, kadai2.tex, kadai3.tex, kadai4.tex
kadai[1-4].tex	kadai1.tex, kadai2.tex, kadai3.tex, kadai4.tex

使用方法:

ls i*	名前がiから始まるファイルをリストする。
rm #*	名前が#から始まるファイルを削除する。
grep all *	カレントディレクトリの全てのファイルから all という文字列を含む行を表示する。

などのように使用します。

5.1.2 ファイル削除の注意

rm 命令でワールドカードを使用するときは誤って削除したいファイルとは別のファイルを削除する失敗が発生する危険性が高くなります。失敗を避けるために削除する作業をファイルひとつひとつに対して削除の確認するオプション(-i)を利用すると良いでしょう。

rm -i #*

の様に使用します。

5.2 情報の収集の方法

ワークステーションを利用して様々な情報にアクセスすることができます。

ターミナル(=gnome-terminal, terminal, xterm 等)内で実行した命令は記録して再利用することができます。これをヒストリと呼びます。ターミナル内で動作しているシェルの種類によってその利用方法は異なります。C-Shell(=csh)やbashの場合はhistory コマンドを実行するとヒストリを参照できます。

5.1.5 ファイル名補完機能

ファイル名の綴りをすべてタイプする代わりに、綴りを補完する機能があります。これも利用しているシェルの種類により利用方法は異なります。

- C-Shell(=csh) の場合は ESC キーをタイプすると、カレントワーキングディレクトリにあるファイルの名前にマッチする綴りを補完します。

- bash や tcsh の場合は TAB キーをタイプすると、コマンド名補完したり、ファイル名を補完します。

5.2.1 アプリケーションについての情報

5.2.1.1 man

ワークステーションには多くの命令がインストールされています。それらをすべて把握することは不可能ですから、大抵の場合は命令の説明をターミナル(=gnome-terminal等) 中で見ることができますようにあります。それがオンラインマニュアルであり、manという命令です。

」マンント名が4つあるといふので、man 「マンント名」を実行します。manは環境変数MANPATH を参照します。キーワード検索は「man -k キーワード」を実行します。

- 便用例

- man -k demo
キーワードを検索します。

man の利用方法を表示します。

- man -s 1 chmod セクション1を選択して chmod を表示します。 chmod 命令の説明。
 - man -s 2 chmod セクション2を選択して chmod を表示します。 C 言語で利用できる chmod 関数の説明。
 - man -s 3 chmod /usr/local/pgsql/man 内に含まれているマニュアルを表示します。

- 日本語表示の例

- env LANG=ja_JP.UTF-8 man man
manの利用方法を日本語で表示します。

`find` コマンドを利用して `/usr` 以下のディレクトリやファイルの中で `man` という綴りのものを検索すると以下の様にリストされます。

- ```
% find /usr -name
...
/usr/local/man
...
/usr/j2se/man
...
/usr/apache/man/
...
```

`find` コマンドを利用して `/usr` 以下のディレクトリやファイルの中で `info` という綴りのものを検索すると以下の様にリストされます。

No next, prev or up links — This is the top of the INFO tree  
**(dir)Top**

File Edit Options Buffers Tools Info Help

## The Info Directory

The Info Directory is the top-level menu of major Info topics. Type `q` to return to the Info Directory. Type `q q` to exit Info. Type `q q q` to get a list of Info commands to visit an Info tutorial. Type `m` to choose a menu item; for instance, `m Emacs` returns visits the Emacs manual.

In Emacs Info, you can click mouse button 2 on a menu item or cross reference to follow it to its target. Each menu line that starts with a \* is a topic you can select with `"*`. Every third topic has a red \* to help pick the right number to type. ■

\* **Menu:**

**Message User Agent**

\* **[Page] [Next]**

**Textinfo documentation system**

How to use the documentation browsing system.

FFTW User's Manual, format, The GNU documentation format, Read Info documents without Emacs.

Compile Info customizations.

\* **install-info(textinfo)**

Update info/dir entries.

\* **makeinfo(textinfo)**

Translate Texinfo source.

\* **pdfTeXinfo(textinfo)**

PDF output for Texinfo.

\* **texi2html(textinfo)**

Print Texinfo documents.

\* **texinfof(texinfo)**

PDF output for Texinfo.

\* **texindex(texinfo)**

Sort Texinfo index files.

**Emacs**

\* **[Page] [Next]**

The extensible self-documenting text editor.

INFO TOC Top TOC14 About Emacs Version 24.2.88

/usr/local/gnu/info 内に存在する info ファイルを見る場合には

```
% info -d /usr/local/gnu/info
```

またいくつかのアプリケーションの説明は info で見ることができます。info は Emacs の中で (M-x info) あるいは (C-h i) を実行して使用できます。Emacs の説明や UNIX で利用できる様々なコマンドの説明を見る事ができます。他には Bash, CVS, Octave, R, Make, Gdb などの説明もあります。

The Info Directory

The Info Directory is the top-level menu option in easier Info topics. Type `cd` into the Info directory to see a list of Info commands, or type `info` to visit an Info tutorial. Type `info info` to choose a generic item, for instance. In easier Info, you can click the mouse menu icon or cross reference to follow it to its target. Each help entry that starts with a \* is a topic; you can select with `"*`. Every third topic has a red \* to help pick the right number to type. []

場合には Emacs の起動設定ファイル `.emacs.el` 等に以下の様に Info ディレクトリを追加するといいでしよう。

```
;; Append info directories
(setq Info-default-directory-list
 (append Info-default-directory-list
 ('("/usr/local/sicstus4/lib/sicstus-4.2.3/doc/info"))
))
```

**注意** 各種プログラムの設定ファイルを編集する場合には慎重に行ってください。充分に内容を把握していない場合は変更を思いとどまることが必要です。また設定ファイルの書き間違いでプログラムが思うように起動できない場合は初期設定に戻すことも考慮しておきましょう。`.emacs.d/init.el` 等の標準環境設定ファイルは `~stdskel/` に配置してあるので参考にすると良いでしよう。

### 5.2.1.3 other documents

また、その他のアプリケーションの説明はどこかにドキュメントがある筈です。find コマンドを利用して `/usr` 以下のディレクトリやファイルの中で `doc` という綴りのものを検索すると以下の様にリストされます。

```
% find /usr -name doc
/usr/share/libjato/doc
/usr/share/sgml/docbook/docbook-xsl-stylesheets-1.60.1/doc
/usr/share/sgml/docbook/docbook-xsl-stylesheets-1.60.1/extensions/doc
/usr/share/sgml/docbook/dsssl-stylesheets-1.77/doc
/usr/share/doc
/usr/share/jedit-4.1/doc
find: /usr/lib/cc-cfw: Permission denied
/usr/openwin/lib/X11/fonts/CSL/doc
/usr/openwin/lib/X11/fonts/doc
/usr/sfw/share/ghostscript/7.07/doc
/usr/sfw/share/gimp-print/doc
/usr/sfw/share/doc
/usr/X11/lib/X11/doc
/usr/X11/share/doc
/usr/jdk/packages/javax.help-2.0/doc
/usr/local/lib/R/doc
```

`doc` ディレクトリ内のファイルの形式は PDF であったり、テキストだったり、HTML だったりと特に決っていません。

## 5.2.2 個人所有の情報 – Home Directories

`/usr/local/doc` などの共通に開かれたディレクトリを覗いて有益な情報を得ることができます。ですが、他のユーザのディレクトリにも有益な情報があるかもしれません。しかし、いくら興

味があるからといって無断で他人のディレクトリに入るのは失礼な行為であって、泥棒呼ばわりされても仕方ありません。

利用者の中にはファイルのアクセス許可を出しておいても許可をしていないという考え方の人があります。その考え方が良いことなのか悪いことのかは別にして、実際に多くの利用者が利用しているネットワークの中では起こり得る現実です。覗きたい場合にはまず問い合わせるのが常識的な方法と考えるのが妥当です。

また、何でもオープンにしたいからと言ってメールのフォルダまでアクセス可能にするのは良いことはありません。他人から届いたメールを勝手に公開することは手紙を書いた人のプライバシーを無視することになります。

## 5.2.3 世界中から発信される情報 – Internet

インターネットで接続された環境を利用している場合には様々な情報を手に入れることができます。

**E-mail** 電子メールは非常に便利なインターネットの機能です。通常の手紙と大きく違つて、同じ内容の手紙を一度に多くの人々に送ることが容易です。そこで次の様な行為は譲ります。

- To:all-student 宛や Cc:all-student 宛にアルバイトの情報やクラブの勧誘、個人的なやりとりを送りつける。

以前は `all-student` という名前のメーリングリストが存在しましたが、不適切な利用が多かったので廃止されました。メーリングリストとはある一つのメールアドレスを多くのメンバーが共有して、同じ内容のメールをやり取りする仕組みです。会津大学では授業別のメーリングリストが提供されています。

<http://web-intt.u-aizu.ac.jp/official/students/sad/stsa@1.html>

**News** News は電子掲示版と呼んでも、インターネット内の壁新聞と呼んでも良いでしょう。

- u-aizu.XXX では学内の人だけが閲覧できます。

- u-aizu.announce は学内向けのアナウンスが掲載されます。
- u-aizu.gakuseibu は学生部からのアナウンスが掲載されます。
- u-aizu.休講、就職説明会、セミナー
- u-aizu.course は講義や演習の情報が掲載されます。また、自ら質問を投稿すれば他の学生にとっても、役に立つ情報が回答される可能性があります。

`u-aizu.announce.gakuseibu` には学生部からの情報が投稿されます。これは学生部の業務の一部であり、これを見て情報を得るのは学生の義務です。`u-aizu.announce.gakuseibu` の情報を取得せずに、不利益を被った場合は学生自身に責任があります。

ネットニュースでは、関係者が質問に対する的確なアドバイスを与えてくれる場合もありますが、一般的には、ニュースに投稿することは彼らの仕事ではない場合が殆んどです。Netnewsの利用は現状では非常に衰退しています。多くのインターネットプロバイダも「かしながら、Netnewsのサービスを終了しました。これをもって、「publicニュースの機能が不要である」という理解は的を得ていません。

- ・匿名性の高いWeb掲示板の利用に多くのユーザーが移動した。
- ・ニュースグループのボランティアベースの運営は継続するのが困難だった。

他にもあるでしょうが、皆さんも理由を自分自身で考えてみてはいかがでしょうか?もしもサービスが流行る流行らないが予言できるのであればビジネスチャンスを確実にものにできるでしょう。

Googleグループ(<http://groups.google.com/>)ではネットニュースに代わり記事を閲覧、検索、投稿できます。

**World Wide Web** World Wide Webは非常にポピュラーです。学内の情報は非常に重要です。

- <http://web-int.u-aizu.ac.jp/labs/istc/ipc/index.html> 情報センタ公式のページなので日頃から目を通す習慣をつけて下さい。
- <http://web-int.u-aizu.ac.jp/labs/istc/ipc/faq/faq.html> 情報センターが提供するFAQ(Frequently Asked Questions)です。

検索エンジンを使いこなすことができれば効率よく情報を収集できます。

- Google <http://www.google.co.jp/>
- Goo <http://www.goo.ne.jp/>
- Yahoo <http://www.yahoo.co.jp/>

Webを基盤技術として、次々と次世代の情報サービス研究開発され、提供されています。Web 2.0などというキーワードで語られる場合もありました。以下に一部をリストします。

#### SNS (Social Network Service)

- ブログ (Weblog, Blog)
- RSS
- Twitter
- facebook

**情報の利用に関する注意** 書籍やネットワーク等を利用して取得した情報を利用する場合には以下に注意しましょう。

- 取得した情報の著作権を侵害しない。
- 取得した情報を噛みにすることは”うそ”に騙される危険があり、またそれを利用することとは”うそ”を拡散させる。

・著作権を主張していないフリーの素材であっても、情報をそのまま利用すると問題が発生する場合がある。例えば大学のレポートのために Wikipedia や百科事典等をそのまま書き写した場合は採点の対象と認めない授業が多い。

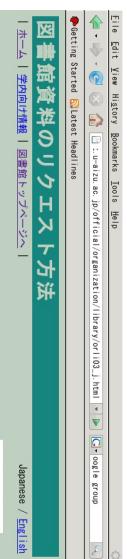
・取得した情報の意味や内容、インパクトなどを考えたりすることで自分の脳細胞を刺さり深く記憶に刻むことができる。それらを省略してレポートに貼り付けるのは単純作業に限りなく近く、結局何も身につかない。

#### 5.2.4 図書館の利用

会津大学図書館のホームページです。  
[http://web-int.u-aizu.ac.jp/official/organization/library/orli\\_j.html](http://web-int.u-aizu.ac.jp/official/organization/library/orli_j.html)

学習や研究生活において情報を活用するためにインターネットを利用できるようになり、大変便利になりました。そのため図書の利用が必要無くなると予想され、書店や図書館の衰退が予想されます。しかし、果たしてこれは本当でしょうか?その様なことを考えつつも、現状では図書館はとても役に立つ機能を提供しているので、これを利用しないのは損という物です。また図書館には文献検索の専門家である図書館司書という専門家がいるので、彼らに相談することもできます。これは、レファレンスサービスといって図書館の重要なサービスの一つです。是非役立ててください。以下に図書館の機能をリストしておきます。

- 調べもの相談(=レファレンス)サービス(司書さんに相談してみましょう。)  
[http://web-int.u-aizu.ac.jp/official/organization/library/orli12\\_j.html](http://web-int.u-aizu.ac.jp/official/organization/library/orli12_j.html)
- 本の貸し出し
- 藏書の検索  
<http://libleopsv.u-aizu.ac.jp/top/index.do?method=change&langMode=JPN>
- 文献検索、取り寄せ
- 文献複写
- 図書館資料のリクエスト  
[http://web-int.u-aizu.ac.jp/official/organization/library/orli03\\_j.html](http://web-int.u-aizu.ac.jp/official/organization/library/orli03_j.html)
- 学術機関リポジトリ  
会津大学ではまだ、本格的に運用が開始されていませんが、全国的には開始されています。  
<http://www.nii.ac.jp/irp/list/>  
学内では卒業論文などの成果物は  
[http://web-int.u-aizu.ac.jp/official/students/sad/sts15\\_j.html](http://web-int.u-aizu.ac.jp/official/students/sad/sts15_j.html)
- DVD 等の視聴覚資料



うに様々な科学の分野を扱っています。  
arxivは一般的の学術論文雑誌と異なり、厳格な査読を通して論文を審査しないので、一定の基準をみたしていれば投稿した論文はすぐに掲載されます。arxivの以下の点で役に立ちます。

- 研究結果を誰よりも早く公開すること
- 最初に誰が発見したか、発表したかの証拠にできる。
- 最先端の研究についての情報交換のスピードが早くできる。
- 学術論文雑誌にはその後でゆっくり掲載する。

arxivに掲載される論文は学術論文雑誌に掲載される前の論文であり、これをプレプリント呼びます。Perelmanが「ボアンカレ予想」を解決したという論文はarxivに投稿され掲載されました。しかし、学術論文雑誌には掲載されていません。

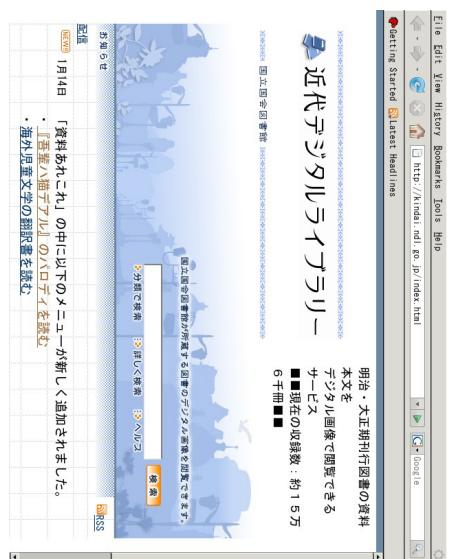


図 5.1. 国書館資料のリクエスト方法

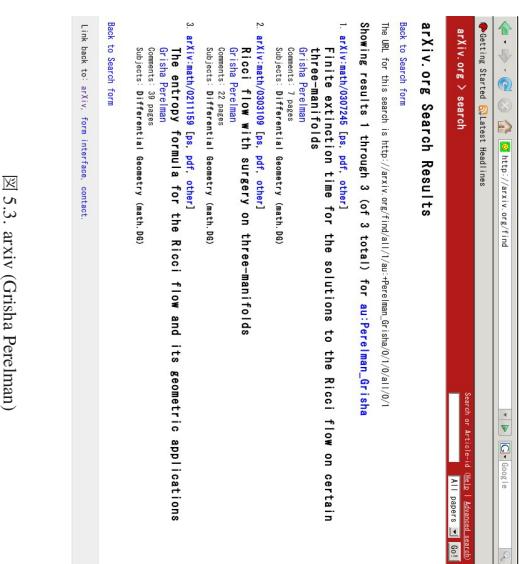


図 5.3. arxiv (Grisha Perelman)

## 5.2.5 国立国会図書館

http://www.ndl.go.jp/ 国立国会図書館では様々な資料を閲覧できます。

## 5.2.6 http://arxiv.org/

arxivは電子論文のサービスです。“Open access to 586,462 e-prints in Physics, Mathematics, Computer Science, Quantitative Biology, Quantitative Finance and Statistics.”この記述にあるよ

図 5.2. 国立国会図書館の電子図書館

## 第6章 ネットワーク利用のエチケットと倫理

- 1995年1月 インターネット接続ホスト数が4,800,000 hosts に増加。
- 1995年7月 インターネット接続ホスト数が6,000,000 hosts に増加。
- 1998年7月 インターネット接続ホスト数は36,739,000台。

### 6.1 Network Etiquette = Netiquette

ネットワークはとても便利です。海外への電子メールは数分の単位で届きます。これは大きな事実です。人類は格違いの量と速度で情報を共有できる時代に突入しようとしています。これは今までの制度や国境を越えて、初めて地球レベルでの文化の共有が実現されるということを意味します。ネットワークを利用する場合に守るべきことがあります。それらは、法律や条例で定められているものもあれば、明文化されないまでも、社会の常識としてある程度の幅を持ちながら理解されているものもあります。Netiquette をキーワードとして、Web で検索すると様々な解説を見ることができます。

#### インターネットは通信か放送か（出版か）

人類の歴史において通信手段の発展はとても重要なものでした。最初は太鼓の音、ノロシ、鏡の光で少ない量の情報を比較的遠方まで伝えるものだったでしょう。交通機関が発達した後では量の多い情報を手紙として伝えることができるようになりました。放送の技術は、ラジオ、テレビです。出版の技術は本、新聞、雑誌、映画、レコードなどがあります。

<https://en.wikipedia.org/wiki/Internet>

で見ると次のように書いてあります。

- ARPANET が 29 Oct 1969 に始まり、
- 1972 年には 40 台のホストが ARPANET に接続され、翌 1973 年には英國とノルウェーに接続されました。
- 1982 年 (230 hosts) には TCP/IP というプロトコルがインターネットの標準通信プロトコルとして確立されます。
- 1984 年 (1,000 hosts)、Domain Name Server が導入されました。
- 1990 年に ARPANET は正式にその役割を終えます。
- 1992 年 (1,000,000 hosts) に CERN の研究者 Tim Berners-Lee が World Wide Web を作りました。
- 1993 年 4 月、NCSA が Mosaic を発表しました。

インターネットでは個人間の通信手段として電子メールは早い時期から利用されてきました。ネットニュースも早い時期から利用されてきました。ネットニュースはやや、新聞やテレビの報道に比較的近く、マスマディアとしての性格を有しています。しかし、一般的なユーザが少なかった 1993 年以前にはミニコミ雑誌程度の影響力しかありませんでした。1993 年 4 月、画像を扱える World Wide Web のブラウザ（掲示ソフトライ）が発表されると、「マルチメディア情報システムが登場された。」という認識が広まり、様子が一変しました。インターネットが一般的なテレビ放送の性格を持つ可能性が出てきました。現在では、電子商取り引き、電子マネー、オンラインショッピング、電子出版など実際の経済活動が平行な形で行われています。また情報検索をするためのサーチエンジンの開発／サービスも行なわれていて、World Wide Web を電子図書館として利用することもある程度は可能となり便利になってきました。

従来の、新聞や雑誌、テレビなどマスコミュニケーションは送り手と受け手が明確に別れています。従来から、送り手は社会に対して非常に大きな影響力を持っています。誰でも情報を誰もが見ることとは無いので、放送としては不十分な状態です。また、「誰でも情報の発信者になれる」というこれまでの情報システムには無かった性格があります。

でも」ということは、「どんな異常な人格の人でも」という意味の場合があります。カルト教団でも、テロリストでも、変質者でもと言うことです。もしも、全く無防備でないにしても、これまでテレビを見ていた気持ちと同じ程度の気持ちでページを見る。

の場合にはそのページに書いてある内容に強く影響され洗脳される確率が高くなります。これは勿論「とても素晴らしいページ」の場合には良い効果となることです。つまり、どの様に判断するかが全く「受け手」の側に押しつけられてしまします。

従来の、新聞や雑誌、テレビなどマスコミュニケーションはある程度世論に審査され続けるものですから、「評判」というものが定まってきます。そこで「受け手」は態度を選択できるのですが、World Wide Web ではあまりに多様な「送り手」の中から情報を得る状態なので「評判」を知る前に、大きな影響を受けてしまう事件が発生しています。1997 年 3 月に発生した「Heaven's gate」の集団自殺事件では、カルト教団がホームページを作り、信者を集め、ほんの数ヶ月前にページを見て入信した人が集団自殺に巻き込まれました。しかし、「ホームページを規制していただきがない」、「個人の問題」などと様々な意見があるでしょう。「受け手」として注意しましょと呼びかけるのは簡単です。しかしながら、会津大学の一員、あるいはコンピュータサイエンティストであれば、情報システムとしてどうあるべきかを考えしていく必要があります。

World Wide Web は程度に違いがあつても、そこに掲載される情報を放送としたならば、出版、放送に関するルールを守る必要があるように考えることができます。一方で、World Wide Web を含めてインターネットを個人間の通信としたならば、個人が有する表現の自由を守る必要があるように考することができます。ところが、実際は、インターネットとは公共の道路の様なもので、そこでどの様なサービスを行なうかによって、個人の通信なのか、放送なのかという性格が大きく違ってくると言えましょう。従って、サービス毎に利用のルールを検討していく必要があります。

## インターネット利用による被害

インターネットの利用によりコミュニケーションの革新が起り、様々な便利なサービスが提供されています。間違いなく我々現代人はそれらの利便性を享受していますが、その一方で被害を被る人もいます。それらの被害について悪い人が原因であるのはまずは間違いありませんが、法律やITの技術などのシステムに起因する部分を我々自身が考えることも必要です。

- ・ネットといじめや裏サイトなど
- ・薬剤のネット販売
- ・出会い系サイトによる児童売買春
- ・アダルトサイトの未成年者による閲覧、利用
- ・大麻種子やドラッグ等の個人輸入
- ・自殺サイト
- ・個人情報の流出による財産、生命の危険
- ・個人情報の流出による財産、生命の危険
- ・最近の事件についての追加です。

- ・反社会的画像や動画の公開
- ・コンビニのアイスの冷蔵ケース内で寝転ぶ写真 → コンビニ閉鎖

- ・国家的組織による盗聴
- ・スノーデン事件

## 法律

今日現在、まだインターネットの利用に関しては法の整備が遅れていると言われています。なぜ遅れているかといえば、それはインターネットがまだ比較的新しくて、やっと普及し始める段階にあるからです。普及し始めて始めて様々な問題が公に知られる機になり、世間が大騒ぎになったその後で、解決方法を探すという事態も少なくありません。参考として表6.1に米国、日本、欧洲でこれまでに著作権や知的財産権について検討されて来た経緯を紹介します。

表6.3にるように日本においても法律の整備や改正が行われています。

また、ネットワークを利用する場合の指針を提供しています。(Request For Comments)RFC1855;FYI23として IETF(Internet Engineering Task Force) がまとめた Netiquette Guidelines が翻訳も添えてあります。一度は読んでみることを勧めます。

<http://www.cgh.ed.jp/netiquette/>

|                  |          |                                                                                                                                                               |
|------------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1992/3<br>1992/4 | 日本<br>EU | 文化庁が著作権審議会マルチメディア小委員会を設置<br>データベースの法的保護に関する閣僚理事会指令の検討を開始                                                                                                      |
| 1993/11          | 日本       | 文化庁のマルチメディア小委員会が第一次報告書を公表<br>第4回 SOFTIC(Software Information Center)シンポジウムで「マルチメディア環境と著作権法」を検討                                                                |
| 1994/2           | 日本       | 知的財産研究所が「マルチメディアを巡る新たな知的財産ルールの提唱(公開草案)」を公表                                                                                                                    |
| 1994/7           | 米国       | ITF(Information Infrastructure Task Force, クリントン政権の権力機関)が「知的財産と NII(National Information Infrastructure)(グリーンペーパー)」を公表<br>文化庁が「マルチメディア小委員会ワーキンググループ検討経過報告」を公表 |
| 1995/2           | 日本       | 「データベースの法的保護に関する閣僚理事会指令についてのコモンポジション」を採択<br>「情報化社会における著作および著作隣接権に関するグリーンペーパー」を公表                                                                              |
| 1995/7           | EU       | EUが「マルチメディア小委員会ワーキンググループ検討経過報告」を公表                                                                                                                            |
| 1995/9           | 米国       | ITFが「知的財産と NII(ホワイトペーパー)」を公表                                                                                                                                  |
| 1995/11          | 日本       | 第5回 SOFTIC シンポジウム「ネットワーク環境下における知的財産権問題」開催                                                                                                                     |

表6.1. 1995年までの各国の政府レベルの検討

|         |                                                                                                                                             |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------|
| 2001/11 | サイバー犯罪国際会議において、インターネットなどを利用した国際犯罪に対処するためのサイバー犯罪条約に、日米欧などの30カ国が署名している。この条約では、インターネットなどを利用した犯罪について、国際的に共通的な定義つけを行い、犯罪捜査方法、刑事訴追方法などについて規定している。 |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------|

表6.2. 最近の条約

## マナー

インターネットが普及していくならば人々のコミュニケーションは大変、密度の高いものになる可能性があります。人間の行動には失敗がつきものです。これは統計的に見れば、一定の割合で人間の行動には失敗が発生するということです。そうなると、コミュニケーションの増大に比例して人々の間の摩擦も非常に多くなると推測することができます。私達が快適に暮らしていくにはあまりに多くの摩擦を人間関係の中で起こす事は好ましくないでしょう。ですから、なるべく、お互いに気持ちよくネットワークを利用するためには守るべきマナーがますます必要になってきます。

通常、あまり話題には出て来ませんが、ネットワークの資源は無限ではありません。一人の人が勝手なことをすると周りの人々に迷惑をかける事になります。そのような事を意識してネットワークを利用する事が、マナーとしても重要です。つまりネットワークに不要なトラフィックを増やしてはいけないということです。

|            |                                        |
|------------|----------------------------------------|
| 1999/10/1  | 著作権法、「不正コピーを補助するソフトの配布禁止を定めた著作権法の修正」   |
| 1999/8/13  | 不正アクセス行為の禁止等に関する法律                     |
| 1999/8/18  | 犯罪捜査のための通信傍受に関する法律                     |
| 2000/5/31  | 電子署名及び認証業務に関する法律                       |
| 2000/12/6  | 高度情報通信ネットワーク社会形成基本法                    |
| 2001/11/30 | 特定電気通信役務提供者の損害賠償責任の制限および発信者情報の開示に関する法律 |
| 2013/1/1   | 著作権法、「違法ダウンロード刑事罰化などの修正」               |

表 6.3. 日本の法律

## セキュリティ

会津大学内では多くのサーバーにあるファイルを共有することによって知的情報をシェアしています。つまりネットワークファイルシステム(NFS)によって他のホストにあるファイルにも容易にアクセスできます。また、多くのユーザが登録されて利用しており、各ホストにはプライベートな情報や研究や大学の管理内容に関係する機密性の高い情報もあるでしょう。一般的にファイルの内容はユーザ、グループ、その他のそれぞれに対するアクセス(読みだし、書き込み、実行)の許可(permission)、不許可によって制御できます。したがって、ファイルのパーミッションが自分自身を含めたユーザ全てに対してアクセスの内容を表示する事になります。ですから、自分の情報をちゃんと管理するにはパスワードに注意する必要があります。また、他人の秘密のファイルのパーミッションが偶然にアクセス可能になっている場合不用意に内容を取得したり、利用する事は良い事ではありません。

会津大学内の計算機資源はリモートで利用可能です。学生の場合、目の前にあるワークステーションにログインすれば、そこからどの演習室のどのホストにでもリモートでログインできます。リモートログインした先のホストのCPUをそのまま利用できるのですから、なんとおよそ500台程度の計算機をその場で利用できるのです。利用可能な状態なのですから、もしもそのような利用の場合には他人に迷惑をかけないように配慮することが必要です。

会津大学の計算機システムは以上に説明したように高度にネットワーク化されています。つまり、一人のユーザーがいろいろな情報にアクセス可能な状況にあります。このような場合、もしもたった一人のユーザーのパスワードが見破られて、悪質なコンピュータ犯罪者に知られたらどんな事が起きてるのでしょうか?おそらく、大学内のユーザ全員に迷惑のかかることになるでしょう。ですから、パスワードの管理はとても大切なマナーです。

## 6.1.1 技術

- 電子メール、ネットニュースの技術
  - 一行に(半角の場合)80文字以内に納める。
  - サブジェクトに漢字コードを書かない。
  - 巨大な内容を避ける。
  - シグネチャは簡略にする。
  - 大事なメールを送る時には、自分宛にCarbon Copy をしておく。

## 6.1.2 マナー

- 電子メール、ネットニュースの共通のマナー
  - 誤解を受けそうな表現は避ける。
  - 匿名や他人の振りをしたメールや投稿はしてはいけない。
  - 不要な添付ファイルを付けない。
  - 電子メールのマナー
    - 通常の手紙と同様に、礼儀正しく書く。
    - 不特定多数宛へのダイレクトメールはしない。
    - チェーンメールはしない。
    - メールの内容を勝手に公開しない。
    - 引用を公開する場合には送り主の了承をもらう。
    - メール転送の設定は間違いの無いように動作確認まで行う。
  - ネットニュースのマナー
    - プライベートなやりとりのみの投稿は控える。
    - 亂暴な言葉使いは避ける。
    - 失言を認めたら素直に謝罪する。
    - また聞きの話を利用しない。
    - 引用をする場合は引用元を明らかにする。
    - ニュースグループのテーマに沿った内容を投稿する。
    - 個人を誹謗中傷(ひぼうちゅうしょう)しない。
    - 無闇やたらに挙げ足を取るような態度は控える。
    - 議論している相手だけが投稿する対象ではないと割り切るなど、気持ちの整理をする。
    - 自分で責任を取れない発言を繰り返さない。
    - 質問に対する返事のお礼をする場合には、どのような質問であったか、回答がどうであったかの内容を簡単にまとめてお礼を添えて再投稿する。

- リモートアクセス(telnet, rlogin, ssh 等)のマナー
- アカウントを持っていないサイトに無闇にリモートアクセスしてはいけない。

- ローカルマナー

- 端末から長時間離れる時は logout する。
- 端末から長時間離れる時は xlock をかけるのではなく logout する。
- 他人の端末をいたずらしたり、覗きこんだりしてはいけない。
- 管理者から指定されたディスクの使用量を守るように気を配る。
- 長時間重いプロセスを実行する場合には nice をかけて、cpu の使用優先度を低く設定して、他のユーザに迷惑をかけない様にする。
- パスワードの管理に注意し、更新のルールに従う。

### 6.1.3 アカウントロック

会津大学では以下の条件の1つが発生した場合、利用者のアカウントを停止します。

- 利用者講習会、あるいはリテラシーの講義を履修していない場合。
- パスワードの更新をしない場合。
- .forward の設定にミスがあった場合。

その他、他のユーザの不利益の原因になる場合や、セキュリティ上緊急性がある場合。アカウントロックはペナルティの意味で実施されることはできません。上記の条件が解決されればロックは即時解除されます。アカウントロックにより困った場合には SSB で相談すると良いでしょう。アカウントロックの状況は <http://web-int.u-aizu.ac.jp/labs/istc/ ipc/topic/locklist.html> で確認できます。

### 6.1.4 情報センターが管理しないコンピュータの利用について

大学内にあるコンピュータで情報センターが管理しないものとその管理責任は以下の通りです。

|                              |               |
|------------------------------|---------------|
| 私物のコンピュータ<br>事務局で利用されるコンピュータ | 所有者<br>研究室の教員 |
|------------------------------|---------------|

それらを会津大学のネットワークである AINS に接続して利用することができますが、以下の条件の全てを満たさない場合は接続を解除し禁止します。

- AINS のグローバルアドレスを使用する場合は接続申請所を提出し、許可を得る必要がある。
- 私物のコンピュータを AINS に接続するために、認証ネットワークを利用してユーザー認証を得る必要がある。

- コンピュータのセキュリティを利用者自身が管理すること。
- 不正な利用をしないこと、および、不正な利用をされないこと。

NAT(=Network Address Translation)機能を持ったルータ等を設置して、ローカルアドレスによる接続を提供している場合は、NAT 機能を持ったルータを接続申請した教員に対して、下流に接続されている全てのコンピュータについて管理責任が発生します。

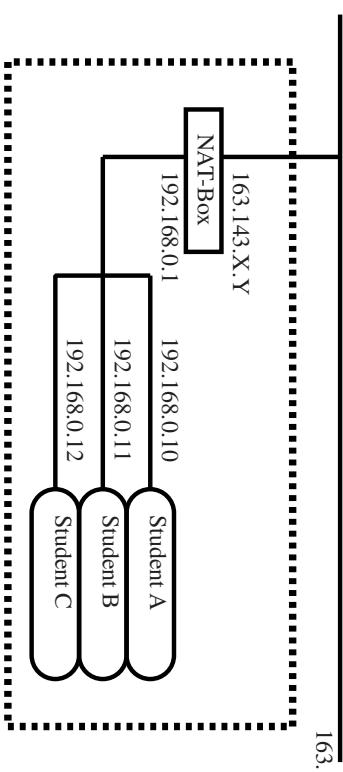


図 6.1. NAT の責任の範囲

### 6.1.5 アクセス制限

ネットワークアクセスを利用して他のコンピュータに対して攻撃的なアクセスをしてはいけません。会津大に対して攻撃的なアクセスした学外のコンピュータやネットワークは一定期間アクセス制限をしています。アクセス制限の状況は <http://web-int/labs/istc/ ipc/Security/access-list.html> で確認できます。

### 6.1.6 コラム 1 —————

研究室では教員がコンピュータを接続申請するけれど管理は学生に任せっきりで、熱心な学生ならウイルスチェックもセキュリティアップデートもきちんととするけれど、代々引き継がれて行くうちに、放置されて、ボロボロに… そして、事件が起きるのであります。

## 第7章 文書作成 $\text{\LaTeX}$ の基礎

### 7.1 $\text{\LaTeX}$ の基礎

$\text{\LaTeX}$  によって機能強化された  $\text{\TeX}$  です。 $\text{\TeX}$  は Stanford University の Donald E. Knuth 教授が、自分の著作を電子出版するための余技に始め、製作した文書整形システムです。このシステムには、文書のレイアウトだけでなく、数式の記述、表などに関するコマンドが豊富に揃っています。ユーザーの目的に合わせて文書設計を楽しめるように設計されています。

$\text{\TeX}$  は会津大学の卒業論文を作成する場合のツールとして指定されています。また、多くの学会で  $\text{\TeX}$  を使用した論文の投稿を指定しています。

$\text{\TeX}$  は Windows や Mac OS X でもインストールし利用できます。

#### 7.1.1 $\text{\TeX}$ とは

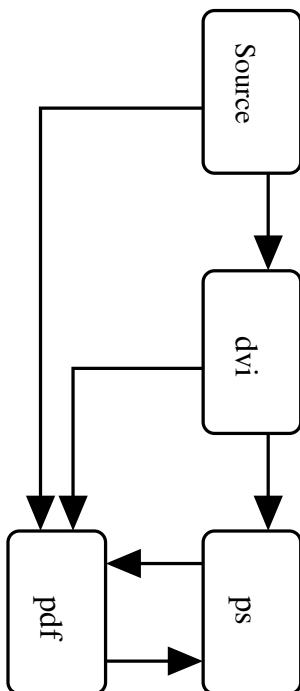
$\text{\TeX}$  は非常に便利なシステムであり、ユーザーの目的を満足させる印刷を得ることが出来る一方で、コマンドの使用が多岐に渡るとともに、非常に基本的な組版機能しか持っていない点が問題です。例えば、何センチ空白を空けるとか、文字の種類を変えるというような基本機能はありますが、章見出しを作るという機能はありません。このような機能は、既存の機能を組合せて作り、 $\text{\TeX}$  に追加登録することができます。このようにして追加登録した新機能のことを macro といいます。

$\text{\TeX}$  自身も、自分の本を書くためにたくさんのマクロを作りましたが、その中で必要最小限のものを集めたものが plain $\text{\TeX}$  というマクロパッケージ(マクロ集)です。この plain $\text{\TeX}$  マクロパッケージで機能強化した  $\text{\TeX}$  のことを plain $\text{\TeX}$  といいます。この plain $\text{\TeX}$  を少し変更した plain $\text{\TeX}$  を土台にさらに  $\text{\TeX}$  マクロパッケージを加えたものがここで学ぶ  $\text{\TeX}$  です。

$\text{\TeX}$  の特徴を簡単にいうと、文書の論理構造を指定するだけで組版が出来るようにつくれたソフトと言うことが出来ます。例えば、「ここは章の見出しがある」というのが論理構造であり、それに対して「ここは12ポイントの明朝体で2行どり左寄せで...」というのが視覚的な構造です。 $\text{\TeX}$  では、著者が「ここは章見出しだる」とだけ指定すれば、あとは書き方を指定したファイル(スタイルファイル)に従って文字の大きさや書き始めの位置といった細かなことが自動的に決定されます。もちろん、すでに登録されている組版が気に入らないなら、登録内容を変更することができます。

このように、テキスト入力と文書フォーマッティングの完全な分業スタイルをバッチ方式といいます。これに対して、テキスト入力と同時に出来上がりのレイアウトを目で確認しながら文書等を作成する方式を WYSIWIG (What You See Is What You Get) 方式と言います。論文や報告書のような定型的な文書作成には文章の論理構造に考え方を集中できるバッチ方式が適していると考えられます。このとき使用される  $\text{\TeX}$  のような、一種のプログラミング言語をマーカアップ言語と呼びます。

#### 7.1.2 $\text{\TeX}$ の処理の流れ



Source まず、Emacs 等のテキストエディタで「.tex」という末尾(拡張子)を持つファイルを作成します。このファイルをソースファイルあるいは単にソースと呼ぶことにします。

Source→dvi: platek sample.tex

Source→pdf: pdflatex sample.tex

pdflatex は日本語のソースには非対応です。

dvi ソースを  $\text{\TeX}$  で処理をします。そのソースに書いてある命令に従って整形・組版を行ない、その結果を dvi ファイルと (Device Independent) 呼ばれるファイルに出力します。この作業をコンパイルと言います。命令が正しく記述されていないと、コンパイルが失敗し dvi ファイルは生成されないか、想定した通りのイメージは得られません。その場合はソースファイルをテキストエディタで手直しし、再度コンパイルします。

dvi→ps: dvips sample.dvi -o sample.ps

dvi→pdf: dvipdfmx sample.dvi

dvipdfmx はパッケージ pstricks には非対応です。

ps, pdf コンパイルに成功したらプレビューを行ないます。これは印刷イメージをウインドウシステムの中で表示し目で確認することです。テキストの誤りや組版の結果が満足いくものでなければソースを修正します。こうした修正を繰返すことによって最終的なものを作り上げていきます。

ps→pdf: ps2pdf sample.ps

pdf→ps: pdf2ps sample.pdf

印刷 必要ならプリンタを使って印刷します。紙も重要な資源の一つですから無駄なプリント出力はしないでください。

以下の節で、具体的な  $\text{\TeX}$  の使い方について説明します。

#### 7.1.3 印刷までの手順

ソースファイルを Emacs 等のテキストエディタで用意します。Emacs で sample.tex を編集します。ソースファイルの名前は「.tex」という拡張子を必ず付けます。ソースのすべて

を入力してセーブしたら、 $\text{platex}$  を使用してそれをコンパイルします。このコンパイルがうまくいかない場合、 $\text{platex}$  の処理の途中で指示を求めてプロンプト「\*」を表示しますが、もしそうなったら「x」とタイプし、コンパイル作業を中止してシェルに戻れます。うまくコンパイルされたならば、以下に示すファイルができます。

```
sample.aux sample.dvi sample.log
```

`sample.aux` は補助 (auxiliary.aug) ファイルと呼ばれ、テキストファイルですから、エディタで読むことも出来ます。aux ファイルは  $\text{\LaTeX}$  の「相互参照」という機能のために次回のコンパイル時に利用します。`sample.log` はログ (log) ファイルといって、コンパイルの実行の途中で画面に表示されるメッセージや実行状態についての情報が書き込まれます。これもテキストファイルであり、エラーが生じたときその原因を調べるために使いますが、消してもかまいません。`sample.dvi` は dvi ファイルと呼ばれ、ソースを元に組み上られた画像です。画面上で出来上がりを確認することをプレビュー (preview) と言います。またそのためのプログラムをプレビュア (previewer) といいます。dvi を画面に表示するツール `xdv` がありますが、日本語フォントには非対応なので、ps や pdf に変換して表示します。

- ポストスクリプトに変換し `gv` で表示する。

```
dvips sample.ps -o sample.ps
gv sample.ps &
```

- さらに PDF に変換し pdfstudio、xpdf や evince を使用して画面に表示する。

```
ps2pdf sample.ps
pdfstudio sample.pdf &
```

プリントへの出力はシステムによって違いますが、例えば会津大学のコンピュータ演習室 1 に設置されている stdlpr1 では

```
lpr -Pstdlpr1 sample.ps
```

によって印刷します。

## 7.1.4 ソースファイルの構造

ソースファイルの全体の構造は次のようになっています。

```
\documentclass[12pt]{jarticle}
アリアンブル
\begin{document}
本文
\end{document}
```

最初の `\documentclass[12pt]{jarticle}` は、レイアウトとして `jarticle` を指定

しています。和文の文章ならば、次のいずれかをレイアウトとして選んでください。

- `jarticle` … 論文、短い報告書など、いくつかの「節」 (section) からなる文書

- `jreport` … 長い報告書など、いくつかの「章」からなる文章で、奇数偶数ページのデザインが同じもの
- `jbook` … 書籍など、いくつかの「章」からなる文書で、奇数偶数ページのデザインの異なるもの

欧文の文書は `article`、`report`、`book` とします。  
`[12pt]` は本文の文字を 12 ポイントでコンパイルしてほしいという指定です。他に 10.11 ポイントが選べます。省略すると 10 ポイントになります。ポイントの長さは国によって若干の違いがありますが、日本では 0.3514mm を 1 ポイントとしています。 $\text{\TeX}$  では  $1\text{point} = \frac{1}{72.27}\text{inch}$  と定めています。

`\begin{document}` と最後の `\end{document}` はペアで使われます。この範囲に本文を書きます。本文は改行を勝手にしてかまいません。行の割り付けは  $\text{\TeX}$  が自動的にやってくれます。また、段落の切れ目には 1 行以上の空白をはさみます。 $\text{\TeX}$  のコマンドは「\」で始まる単語です。JIS キーボードでは「¥」で、英語圏のキー ボードではバックスラッシュ「\」です。これらが入り混じって出てきても同じ文字だとみなしてください。

## 7.2 数式

この章では一般的な数式の扱い方を説明します。  
 数学者や物理学者、コンピュータサイエンティストを満足させるだけのテクニックはこの章だけでは十分には紹介できませんが、 $\text{\TeX}$  には彼らから十分な支持を得られるだけの機能が備っています。興味のある人は章末の文献案内を参照して下さい。

### 7.2.1 2つの数式モード

$\text{\TeX}$  では数式の取り扱いは次の 2 つに大きく別れています。

- 本文中の数式モード
- 別組の数式モード

#### 7.2.1.1 本文中の数式モード

文章中に数式を書き入れるにはこのモードを使います。ソースファイルに単に `a` と書けば `a` のようにローマン体で出力されますが、これを `\(t)` で囲んで `\(t)` と書くと `t` のように数式用の書体になります。`\$` で囲んでもかまいません。

|                                  |                               |
|----------------------------------|-------------------------------|
| ソース                              | 出力                            |
| 曲線のパラメータを <code>\(t)</code> とする。 | 曲線のパラメータを <code>t</code> とする。 |

### 7.2.1.2 別組の数式モード

別組の数式とは

$$P = P(t)$$

のように改行して行の中央に数式を書くためのモードです。別組の数式は、`\[`と`\]`で囲みます。`$$`で囲む方法もあります。別組の数式モードは、本文中の数式モードとは使われる文字の大きさや見栄えが多少異なりますが、書き方はほとんど同じです。

**ソース**

```
\sqrt{a = \int_a^b [f(x)]^2 dx}
```

この無限集合の中から、”最も滑らかな”関数を求める。ただし、”最も滑らかな”というのには、積分が可能な限り小さいことを意味する。この意味での最良補間関数がただ一つ存在し、それが $\sqrt{(2k-1)}$ 次のスプライン関数であることが後でわかる。

**出力**

この無限集合の中から、”最も滑らかな”関数を求める。ただし、”最も滑らかな”というのは、積分が可能な限り小さいことを意味する。この意味での最良補間関数がただ一つ存在し、それが $2k-1$ 次のスプライン関数であることが後でわかる。

$$a = \int_a^b [f(x)]^2 dx$$

この例の様に`\[`と`\]`で数式モードを利用した場合には、括弧の対応でモードの開始と終了を判別でき、Emacs 等の括弧の対応を確認する機能のあるエディタ等を利用すると便利です。

### 7.2.1.3 Examples

**ソース**

数式は数式モードに入ると手軽に書くことができます。数式モードに入る方法は、主に`\$`と`\$`で囲む方法、`$$`と`$$`で囲む方法、

```
\begin{equation}
y = x^2 + 1
\end{equation}
```

の様に`equation`環境を使う方法、そして

```
\begin{eqnarray}
y &=& x^2 + 1 \\
&=& 2^2 + 1 \quad (x = 2 を代入) \\
&=& 5
\end{eqnarray}
```

`\end{eqnarray}`

の様に`eqnarray`環境を使う方法(これは(`\ref{lec:siki}`)式に`x = 2`を代入した結果です。)等があります。

```
Index
$ a_{ij} b^{ijk} = c_{i1^k}
Fractional number
$ \frac{x^2 - y^2}{x+y} = x - y
```

```
Vector
$ \vec{x} = (1, 0), \quad \vec{y} = (0, 1), \quad \vec{x} \cdot \vec{y} = 0,
\quad \vec{a} = \overrightarrow{AB} + \overrightarrow{BC} = \overrightarrow{AC}
Sum, Integral and limit
$ \lim_{N \rightarrow \infty} \sum_{i=1}^N f(x_{i-1})(x_i - x_{i-1}) =
\int_0^\infty f(x) dx
```

```
Greek
$ \lim_{\epsilon \rightarrow 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}
放射線にはその実体として、ヘリウム原子核、電子、高エネルギー光子があり、それぞれα線、β線、γ線と呼ばれている。\\
Square root
$ \sqrt{x^2 + 2x + 1} = |x + 1|
```

```
Integration
\int_a^b \left(-1 + \frac{2}{(1+x)(1-x)} \right) dx
```

```
Matrix
\$ \left(\begin{array}{ccc|c}
a_{11} & \dots & a_{1n} & \\
\vdots & \ddots & \vdots & \\
a_{m1} & \dots & a_{mn} &
\end{array} \right)
```

```
\hline
&&c
\end{array} \right)
```

**出力**

数式は数式モードに入ると手軽に書くことができます。数式モードに入る方法は、主に $x^2$ の様に`\$`と`\$`で囲む方法、

$$y = x^2 + 1$$

の様に`\$`と`\$`で囲む方法、

$$y = x^2 + 1$$

の様に`equation`環境を使う方法、そして

$$\begin{aligned} y &= x^2 + 1 \\ &= 2^2 + 1 \quad (x = 2 を代入) \\ &= 5 \end{aligned} \tag{7.1}$$

の様に`eqnarray`環境を使う方法(これは(7.1)式に $x = 2$ を代入した結果です。)等があります。

Index

$$a_{ij} b^{jk} = c_i^k$$

$$\frac{x^2 - y^2}{x + y} = x - y$$

Vector

$$\vec{x} = (1, 0), \quad \vec{y} = (0, 1), \quad \vec{x} \cdot \vec{y} = 0, \quad \overrightarrow{AB} + \overrightarrow{BC} = \overrightarrow{AC}$$

Sum, Integral and limit

$$\lim_{N \rightarrow \infty} \sum_{i=1}^N f(x_i)(x_{i+1} - x_i) = \int_{x_0}^{x_\infty} f(x) dx$$

Greek

$$\lim_{\epsilon \rightarrow 0} \frac{\frac{f(x+\epsilon) - f(x)}{\epsilon} - f'(x)}{\epsilon} = \frac{d}{dx} f'(x)$$

Square root

Integration

Matrix

$$\int_a^b \left\{ -1 + \frac{2}{(1+x)(1-x)} \right\} dx \quad (7.5)$$

$$\left( \begin{array}{c|cc} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \\ \hline c & & \end{array} \right)$$

7.3 箇条書

L<sup>A</sup>T<sub>E</sub>Xには、箇条書き用に, itemize, enumerate, description の3つの環境が用意されています。単に項目をならべるだけなら itemize 環境を、各項目の通し番号をつけるには, enumerate 環境を用います。各項目の先頭には, \item コマンドをつけます。enumerate 環境については、54 ページの 8.1.4 例を見てください。

```
\begin{itemize}
\item itemize
\item enumerate
\item description
\end{itemize}
```

- itemize
- enumerate
- description

7.4

LaTeXには`tabular`という便利な表組みの機能があります。これを使えばワープロで罫線を引いて表を作るよりも簡単にきれいにそれらを作ることができます。まず、例を見て下さい。

**itemize** 単に各項目をならべます

**description** 各項目に項目名をつけてます

出力

```

\hline & 講義と演習 & 林, 程 \\
& 演習 & 寺園, 浅井 \\
& 演習 & 寺坂, 出井 \\
& 演習 & 藤津, 村上 \\
\hline
\end{tabular}

```

\begin{tabular} と \end{tabular} が表を出力する命令です。この命令は

```
\begin{tabular}{列指定}
表本体
\end{tabular}
```

効率的に簡単に書けるような環境です。列指定は

- C ... 中央寄せ (center)

- r ... 右寄せ (right)

$\text{\phantom{string}}$  と並べます。| は罫線を意味しています。  
 表本体で、列の区切りは &、行の区切りは \ です。表の最下行の終りには \ は付けません。  
 $\backslash hline$  コマンドは横線を入れる指示です。行と行の間に罫線を入れたいときには  $\backslash hline$  を書き込む場所は、ソースの中では比較的自由です。ソースをもう一度よく見て下さい。  
 表をつくるときに、ある文字列と同じ幅の空白をつくりたいことがあります。そのようなときには、 $\backslash phantom{string}$  とすると、string と同じ幅の空白ができます。 $\backslash underline$ などを組み合わせると便利です

## 7.5 図

$\text{\LaTeX}$  には簡単な線画を描く機能があります。かなり複雑な図を書くこともできますが、それらとは別のソフト(例えば tgif, xfig など)を使って描く方が一般的です。図を書くのに便利なパッケージの一覧を紹介しておきます。

- グラフ理論のグラフなどを書く … pstricks, pst-node など
- 電子回路図 … circ など
- 楽譜 … musixtex など

他にもいろいろなものがあります。直接図などを書くためのものではないですが、 ifthen パッケージや calc パッケージも図を書くのに重宝します。 $\text{\LaTeX}$  に図などをいれる際には、[3, 11] が参考になります。

### 7.5.1 図のための空白の作り方

$\text{\LaTeX}$  で作成する文書に図などが入るスペースを空けておくことがあります。縦方向のスペースを 1cm 空ける命令には  $\backslash vspace\{1cm\}$  があります。ただし、このコマンドではページの上端に来るスペースは省略してしまうので、その場合には  $\backslash vspace*\{1cm\}$  を使います。

## 7.6 カスタマイズ

$\text{\LaTeX}$  では、コマンドを新しく定義したり、書式を書き換えたり、自分専用の書式を作ったりできます。それには、通常、マクロを定義する方法が用いられます。

```

\newcommand{\cube}[1]{#1^3}
\begin{document}
\begin{equation}
\begin{array}{c}
\text{\cube{a}}=5 \\
\end{array}
\end{equation}
\end{document}

```

## 7.7 参考文献

文献 [4] や [8] は  $\text{\LaTeX}$  についての参考書です。[9] は、Lamport 自身による  $\text{\LaTeX} 2\varepsilon$  の解説書。[1–3, 5, 10, 11] は、 $\text{\LaTeX} 2\varepsilon$  について、上級者も含めて、好評な参考書です。[2] は、 $\text{\LaTeX}$  と HTML や XML との統合について書かれた良い参考書です。[5] は、 $\text{\LaTeX} 2\varepsilon$  についての参考書です。[15], [16] は、 $\text{p}\text{\LaTeX} 2\varepsilon$  を使って様々にできる事を示しています。[4] は、数式を書く上で、大変参考になります。[6, 7] は、 $\text{\LaTeX} 2\varepsilon$  とは直接は関係ありませんが、最終フォーマットとして重要な PDF についての資料です。

## 関連図書

- [1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The  $\text{\TeX}$  Companion*. Addison-Wesley, 1994.
- [2] Michel Goossens and Sebastian Rahtz.  *$\text{\TeX}$  Web Companion : Integrating  $\text{\TeX}$ , Html and Xml*. Addison-Wesley, 1999.
- [3] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The  $\text{\TeX}$  Graphics Companion : Illustrating Documents With Tex and Postscript*. Addison-Wesley, 1997.
- [4] George Grätzer. *MATH Intro  $\text{\TeX}$* . Springer Verlag, third edition, 2000.
- [5] Patrick W. Daly Helmut Kopka. *A Guide to  $\text{\TeX} 2\varepsilon$* . Addison-Wesley, 1994.
- [6] Adobe Systems Inc. *PDF Reference version 1.4*. Addison-Wesley, third edition, 2001.
- [7] Adobe Systems Inc. *PDF Reference Version 1.3*. ピアソン・エデュケーション, 2001.
- [8] Leslie Lamport. *文書処理システム  $\text{\TeX}$  . ピアソン・エデュケーション*, 1999.
- [9] Duane Bibby Leslie Lamport. *A Documentation Preparation System User's Guide and Reference Manual*. Addison-Wesley, 1993.
- [10] Alexander Samarin Michel Goossens, Frank Mittelbach.  $\text{\TeX}$  コンパニオン . アスキー出版局, 1998.
- [11] Alexander Samarin Michel Goossens, Frank Mittelbach.  $\text{\TeX}$  グラフィックスコンパニオン . アスキー出版局, 1999.
- [12] 中野賢. 日本語  $\text{\TeX} 2\varepsilon$  ブック. アスキーオブン局, 1997.
- [13] 磯崎秀樹.  $\text{\TeX}$  自由自在. サイエンス社, 1992.
- [14] 奥村晴彦.  $\text{\TeX} 2\varepsilon$  美文書作成入門 改訂版. 技術評論社, 2000.
- [15] 藤田眞作.  $\text{\TeX} 2\varepsilon$  階梯(第2版). ピアソン・エデュケーション, 2000.
- [16] 藤田眞作.  $\text{\TeX} 2\varepsilon$  入門・縦横文書術・ピアソン・エデュケーション, 2000.

## 第8章 文書作成 $\text{\LaTeX}$ の参照, 作図

### 8.1.3 pageref コマンド

`pageref` コマンドは, `label` が定義された場所のページの番号を生成します. 文法は,

`\pageref{labelname}`

です.

例えば, 図 2 というのが 5 ページに来るのなら,

`P.\pageref{fg2}` の図`\ref{fg2}`を見よ.

### 8.1 相互参照

$\text{\LaTeX}$  では, `label` を介して章や式, 図表などの番号を参照できます. また, 参考文献番号も自動的に参照します. `label` を介することによって, 章立ての変更や, 図, 表, 式, 文献の追加や削除によって, 本文中の参照部分を一々変更する必要がなくなります.

#### 8.1.1 章, 式, 図表等の参照

章, 式, 図表等の参照には, `label`, `ref`, `pageref` のコマンドを使います.

##### 8.1.1.1 label コマンド

`label` コマンドは, 指定された `labelname` に, 番号を与えます. `label` コマンドの文法は,

`\label{labelname}`

です.

`label` コマンドが, `labelname` に与える番号に影響を及ぼすコマンドと環境は,

`section` 等, `table`, `picture` 環境と `caption`, `equation` 環境, `eqarray` 環境, `enumerate` 環境 (後述)

等です. `table`, `picture`, `equation` 環境に対しては, 環境内 (`\begin{...}` と `\end{...}` の間) に `label` コマンドをおこことで有効になります (例参照).

`enumerate` 環境の項目番号`\item`などの `label` をつける時には, `\item` の直後に, `\label` をおきます。

##### 8.1.1.2 ref コマンド

`ref` コマンドは, その場所に, `labelname` に与えられた番号を貼りつけます. 文法は,

`\ref{labelname}`

です.

`ref` コマンドは, 番号しか生成しません. そのため, 例えれば `section` を参照するなら, 第`\ref{labelname}`節, 図を参照するなら, 図`\ref{labelname}`, のように指定しなければなりません.

##### 8.1.1.4 例

`ソース`

```
\section{序}
\label{intro}
この文章では, \LaTeX における label の扱いについて述べる.
label を扱うコマンドは, label, ref, pageref 等がある.
詳しく述べは, 第\ref{usage}節を見よ.
\par
```

```
\section{label の使い方}
\label{usage}
section の参照は, 第\ref{intro}節, のように行う.
P.\pageref{intro} のようになど, ページで参照することも出来る.
\par
```

```
図, 表の参照は, 図\ref{fig1}, 表\ref{table1}, のように行う.
\begin{figure}[htbp]
\begin{center}
\fbox{ここに図が入る.}
\end{center}
\caption{図の参照の例}
\label{fig1}
\end{figure}
```

```
\begin{table}[htbp]
\caption{表の参照の例}
\label{table1}
\begin{center}
\fbox{ここに表が入る.}
\end{center}
\end{table}
```

```
\begin{table}[htbp]
\caption{表の参照の例}
\label{table1}
\begin{center}
\fbox{ここに表が入る.}
\end{center}
\end{table}
```

箇条書きの参照は, 以下のとおりである.

```
\begin{enumerate}
\item これは 1 項目である.\label{item1}
```

```
\item これは 2 項目目である.\label{item2}
\item これは 2 項目目と 3 項目目の間に後から挿入した.\label{item2.5}
\item これはもともと 3 項目目であった.\label{item3}
\end{enumerate}
ここで、\ref{item2.5} を見よ、\ref{item3} を参照せよ、のように参照する。
```

**出力**

この文章では、 $\text{\LaTeX}$ における label の扱いについて述べる。label を扱うコマンドは label, ref, pageref 等がある。詳しくは、第 8.2 節を見よ。

## 8.1 序

section の参照は、第 8.1 節、のように行う。P.55 のように、ページで参照することも出来る。図、表の参照は、図 8.1、表 8.1、のように行う。

ここに図が入る。

図 8.1. 図の参照の例

表 8.1. 表の参照の例  
ここに表が入る。

箇条書きの参照は、以下のとおりである。

1. これは 1 項目目である。
2. これは 2 項目目である。
3. これは 2 項目目と 3 項目目の間に後から挿入した。
4. これはもともと 3 項目目であった。

ここで、3 を見よ、4 を参照せよ、のように参照する。

## 8.1.2 相互参照を有効にする

$\text{\ref}($  や  $\text{\cite}$ ) を使ったときは、正しい出力を得るために  $\text{\plataex}$  を 2 回実行しなければなりません。1 回目の実行では、次のような警告メッセージが画面に現れます。

```
LaTeX Warning: Reference 'Intro' on page 1 undefined.
LaTeX Warning: Citation 'boehm' on page 1 undefined.
LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.
```

つまり「正しい相互参照を得るためにもう一度実行してください」ということです。もしこの警告を無視して dvi ファイルを印刷すると、節番号・引用のところが伏せ字 („„) になってしまいますのでもう一度コンパイルします。

## この講義の目的

技術的なレポートの書き方、特に技術的なレポートでは不可欠な図の書き方について学びます。ここでは、グラフィックエディタ tgif の使い方とそれを用いた  $\text{\LaTeX}$  ファイルへの図の取り入れ方について説明します。

## 8.2 グラフィックエディタ

グラフィックスエディタには、いろいろありますが、現時点では皆さんができるもの

- tgif
- xfig
- gimp

このうち、tgif について、簡単な説明を加えます。

注意

英語の論文などに使う画像を作成する際には、日本語化されていないグラフィックエディタを使う方がより安全です。日本語化されたものから出力できる PostScript や encapsulated PostScript のデータには、日本語フォントの情報が含まれてしまい、日本語フォントに対応していないプリンターでは出力できない可能性があります。

### 8.2.1 機能概要

tgif は、X ウィンドウ上で、会話的に図形の作成・操作を行なう作図ツールで、PostScript プリンタに直接出力したり、 $\text{\LaTeX}$  で書かれた文書内に取り込むことが可能な形式の PostScript ファイル (encapsulated PostScript) を生成したりすることができます (カラーもサポートしています)。

描画できるもの (描画プリミティブ) としては、テキスト、四角形、橢円、ポリライン (連結した線分列)、ポリゴン (多角形、橢円弧、角丸四角形 (角が円弧になった四角形)、ビットマップ (1 画素ごとに色を指定して図形を作る)) があります。

## PostScript とは?

PostScript<sup>1</sup> はページ記述言語(page description language=PDL)の一つです。ページ記述言語は、文書のページイメージを表現するための言語のことです。フォントの指定、文字の指定、およびその文字の印位置の指定等に分解してとらえると、それらに対応する手続きや演算子を備えた言語によりページの内容を記述できます。PostScript では、記述言語そのものをプリンタに対して送出し、通信の手段に依存しないで簡単にページ表現を送ることができます。ハーフトーンを含んだグラフィックス機能や、文字の変形操作ができるなどの機能も備わっています。最近では、Level3 も現れるなど、新しい機能が追加されてきています。特に、Font に関して、CID(Character Identifier)-keyed encodingへの変更がなされ、古いファイルによっては、プリンタやソフトが読むことができないといった事態が起きています。

## 8.2.2 tgif の起動

tgif を起動するには、まず X ウィンドウを起動し terminal emulator ウィンドウから

`tgif [option...] [filename] &`

と入力します。tgif で描いた図は、ファイル名に「.obj」という拡張子が付いたファイル(.obj ファイル)に保存されます。tgif 起動の際に指定するファイル名 [filename] には、.obj ファイルのファイル名から拡張子を省略したファイル名を指定します。

主なオプションとしては以下のものがあります。

- -reverse (または、-rv) … 画面を反転表示します (デフォルト=default : オプションを指定しない場合は、フォアグラウンド色が黒、バックグラウンド色が白)。

- -eps … デフォルトのプリントアウト先を「.eps ファイル」にします (デフォルトは、プリンタ)。

- -hidedragcursor (または、-hc) … ドラッグ中のマウスカーソルを表示しない。
- -savewitheps (または、-swe) … ファイルのセーブ時に".eps ファイル"も生成します。

- tgif -print -eps … obj ファイルを eps ファイルに簡単に変換できます。

## 8.2.3 起動時のウィンドウ

tgif を起動すると 56 ページの図 8.2 ようなウィンドウが開かれます。

1. Message Window … tgif からのメッセージが表示されるウィンドウ。エラーメッセージ等も表示されます。表示されたメッセージは、マウスの「左ボタン」でスクロールアップ、「右ボタン」でスクロールダウンします。

2. Choice Window … 現在の描画モードと描画属性(線の幅、フォント、色、塗りつぶしのパターン、ペンシル、線の端点形状、破線パターン等)を表示します(57 ページ図 8.3 参照)。

3. Canvas Window … 実際に描画を行なうウィンドウ。

4. Scrollbar … 「Canvas ウィンドウ」の内容をスクロールします。

5. Ruler … 目盛りと現在のグリッドの間隔を表示します。

<sup>1</sup>PostScript は、Adobe Systems Inc. の商標です

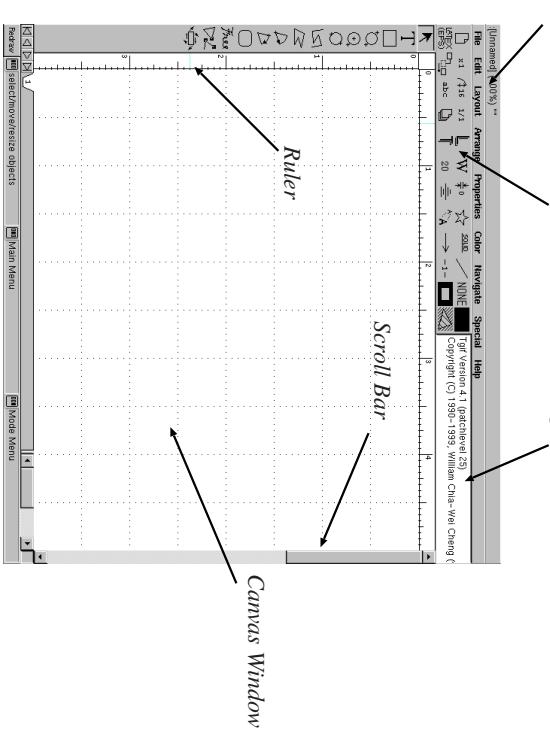


図 8.2. サブウィンドウの名前

## 8.2.4 作図

### 8.2.4.1 メインメニューの表示

「中ボタン」を「Canvas ウィンドウ」の中で押し続けると、図 8.4 のようなウィンドウが表示されます。これがメインメニューです。メインメニューからアイテム(Mode, File, Edit...)を選択することで、tgif のすべての機能を使用することができます。

1. 「カーソル」を「Edit」という文字に持っています。

2. 「左ボタン」を押し続けます。
3. 「Edit メニュー」が表示されます。
4. 「左ボタン」を押したまま、「Edit」メニューの上をなぞると、アイテムが次々と反転表示します。
5. 例えば、「SelectAll」の上で、ボタンを離すと「オブジェクトをすべて選択する」が実行されます。

このように、マウスボタンを押しながらマウスを動かし、目的のアイテムのところで、ボタンを離すことを、「メニューを選択する」といいます。メニューは、良く使うメニューなので、以下の方法で画面に貼り付けて置くと良いでしょう。

1. メインメニューを表示した状態で、マウスを右上に動かしていって下さい。
2. しばらく動かすと枠が抜け出します。
3. その後で、「右ボタン」を離すとメニューが貼りつきます。
4. 移動させる時は「右ボタン」でドラッグ、消す時は「中ボタン」でクリックします。

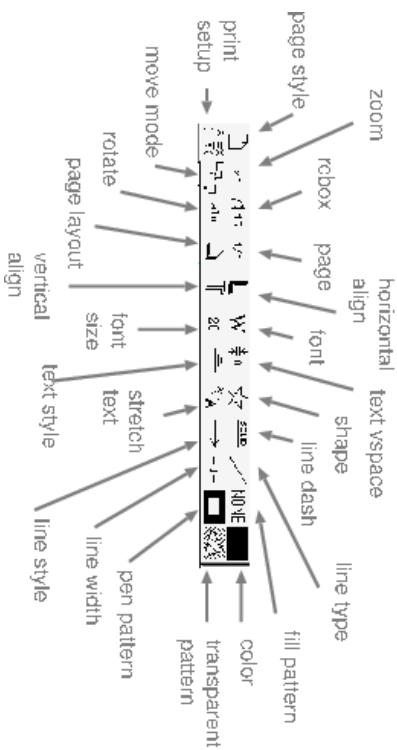


図 8.3. Choice ウィンドウ

## 8.2.5 例えば、四角形を描く

四角形を描いてみます。

1. 「Mode メニュー」から「四角形」を選択します。
2. カーソルが白くなります。
3. 「Canvas ウィンドウ」の左下の適当な場所で、「左ボタン」を押しドラッグして下さい。
4. 四角いラバーバンドが表示されます。

なお、一度描画モードになると、「Mode メニュー」から「矢印」を選択して「操作モード」にするまでは(つまり、カーソルが「白い」うちは)、同じ图形が続けて描画できます。操作モードになるとカーソルは「黒く」なります。

## 8.2.6 マウスボタンの意味

tgifは、3つのマウスボタンを仮定した設計になつとおり、それらの意味は以下のようにまとめられます。

|      |                                         |
|------|-----------------------------------------|
| 左ボタン | オブジェクトの描画、選択、変形                         |
| 中ボタン | ポリライン、ポリゴンの描画中<br>文字列の描画中               |
| それ以外 | 領域の指定                                   |
| 右ボタン | tgifのメインメニューのポップアップ<br>Mode メニューのポップアップ |

表 8.2. マウスボタンの意味



図 8.4. メインメニュー

## 8.2.7 作図のまとめ

- 描画

「Mode メニュー」から描画したいオブジェクトを選択して行ないます。

- オブジェクトの選択  
「左ボタン」をクリックして行ないます。
- オブジェクトに対する操作(属性を変える、移動する、拡大する等)操作対象のオブジェクトを選択してから行ないます。  
- 属性の変更  
選択したオブジェクトに対してメニューで行ないます。
- 拡大(縮小)  
選択したオブジェクトの周りの「小さな四角」を「左ボタン」でドラッグすることで行ないます。  
- 移動  
選択したオブジェクトを「左ボタン」で、「小さな四角以外」の部分をドラッグすることで行ないます。

## 8.2.8 保存

$\text{\LaTeX}$  に描いた図をファイルに保存します。「File」メニューから「Save」を選択します。新たに「Choice ウィンドウ」の「プリント先」アイコンを「左ボタン」で「 $\text{\LaTeX}$  アイコン」になるまでクリックして下さい。

1. 「File メニュー」から「Print」を選択して下さい。
2. 「File メニュー」から「Print」を選択して下さい。
3. カレントディレクトリ→、「*fileName.eps*」というファイルが生成されます。

## 8.2.10 終了

「File メニュー」から「Quit」を選択すると *tgif* は終了します。

## 8.3 $\text{\LaTeX}$ への図の取り込み

*tgif* や *xfig* で描いた図を、 $\text{\LaTeX}$  で書いた文書に取り込むには次のようにします（例として図「drawing-sample」を取り込んでみます）。図をとりこむには「.eps ファイル」（この例では、「drawing-sample.eps」）を用います。

以下のような  $\text{\LaTeX}$  のソースファイル *example.tex* を作ります。

ソース

```
\usepackage[dvipdfmx]{graphicx}

\documentclass[12pt,a4paper]{article} % A4,12pt 取り込み
\usepackage{dvipdfmx} % graphicx package (graphicsではない)
\usepackage{psfrag} % ps ファイル中の text を変換

\begin{document}
この文章には、tgif で描いた図が入っています。
\psfragscanon
\psfrag{\alpha}{\$\\alpha\$}
\begin{figure}[htbp]
\begin{center}
\includegraphics[scale=0.8]{% 例として /home/course/lit1/pub/literacy/FIGURE/drawing-sample.eps
\caption{tgif で描いた図}
\label{example}
\end{center}
\end{figure}
\end{document}
```

ここで、「\includegraphics」というのは encapsulated PostScript 形式等のファイルを  $\text{\LaTeX}$  に取り込むためのコマンドです。これを使うには、プリアンブルに

\usepackage{graphicx}

を入れて、graphicx パッケージを使用することを宣言します。*graphicx* パッケージでは、現在は、eps フォーマットの他に pdf, png, jpeg がサポートされています。一部の商用の  $\text{\LaTeX}$  では tiff フォーマットなどもサポートされています。各種画像フォーマットから eps フォーマットへの変換には、

• convert

- bmeeps
- imconv<sup>2</sup>
- ps2eps

などのコマンドが利用できます。また、pspicture パッケージなどを用いて  $\text{\LaTeX}$  で絵を書いて、dvips を-E オプションを使って eps ファイルを作る事もできます。xfig で作成した図を fig2epic と epic パッケージで利用することができます。

ここで、「\psfrag」というのは、encapsulated PostScript 形式等のファイルを  $\text{\LaTeX}$  に取り込んだ後に eps ファイル中の文字（tgif などで text として入力したもの）を  $\text{\LaTeX}$  の文字列に置き換えるコマンドです。これを使うには、プリアンブルに

\usepackage{psfrag}

を書きます。これによって、tgif などでは入力が困難な、数式やギリシャ文字などを図の中に簡単にいれる事が、可能となります。文字列は、ps ファイルの中で文字列の埋め込み――

```
gsave
 608 127 moveto (sample.tex) show
 restore
```

の様に記述されていて、この場合、*sample.tex* が文字列になっています。psfrag は、個々の文字列全体が指定した文字列と合致したものだけ変換しますので、

\psfrag{sample}{\bf sample}

では、上記の部分を変換しません。この部分を変換するには、

```
\psfrag{sample.tex}{\bf sample.tex}

\psfrag{sample}{\bf sample}
```

実際に図を取り込む命令は \includegraphics です。graphicx パッケージを利用してい場合にこの命令を利用できます。

<sup>2</sup>Use 'imconv -help' for help.

イル」を更新し忘れれば、その図を取り込んでいる $\text{\LaTeX}$ の文章をプリントアウトしても図は修正されないということになってしまいます。  
以下に tgif で描いた図を取り込んだ例を示します。

```
\usepackage{graphicx}
\includegraphics[height=2.4cm, width=1.2cm]{%

/home/course/lit1/pub/literacy/FIGURE/fig1.eps}
```

とすると、高さ 2.4cm で、幅 1.2cm で取り込まれます。全角文字 5 文字分の大きさにする場合には width=5zw を使用します。

```
\usepackage{graphicx}
\includegraphics[scale=0.3]{aizu.eps}
```

とすると、元の画像の大きさの 0.3 倍になります。これらの方法で、大きさを調節できます。  
画像を回転させるために、angle=45 を使用することもできます。回転の中心は origin=c  
とすれば図形の中心を指定したことになります。

```
\usepackage{graphicx}
\includegraphics[origin=c, angle=60]{aizu.eps}
```

### 8.3.1 figure 環境

figure 環境を用いると図を管理し適切にレイアウトを実行します。また、図の説明のコメ  
ントである Caption を記述できます。オプションの h はここに配置、 b はページのボトムに  
配置、 t はページのトップに配置、 p は独立したページを指定し、書いた順に優先されます。  
挿入箇所に図を確実に配置するためには

```
\usepackage{float}
```

を利用して H をオプションとして使用します。

### 8.3.2 from dvi to ps

tgif で描いた図を取り込んで、出力するためには、dvi ファイルを PostScript ファイルに  
変換します。(文字も、図も PostScript にして合わせます)

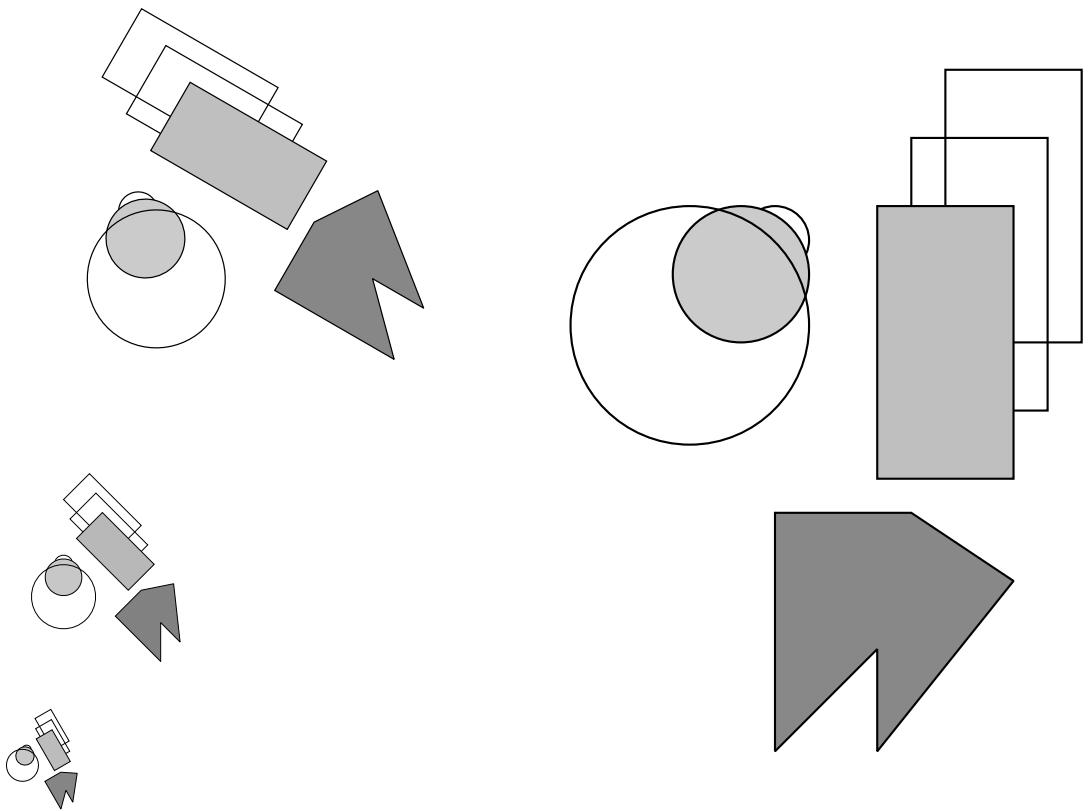
```
dvi2ps example.dvi -o example.ps
```

「.ps ファイル」は、プレビューウィンドウ gv でプレビューできます。

印刷するために、このファイルをプリンタに送ります。

```
lpr -Pprintename example.ps
```

printename にはネットワークプリンターのどれかを指定して下さい。  
 $\text{\TeX}$  のファイルに tgif で作成した図を取り込むのに使うのは、「.obj ファイル」では  
なく、「.eps ファイル」であることに注意して下さい。ファイルをセーブしても、「.eps ファ



## 第9章 学術論文の作成, スライド

**本論** 本論を書く場合の注意に關しては、後で述べます9.2.2  
**結び** 本論の結論をまとめたもの、および残された問題点やその研究の今後の展望など  
 を述べます。

### 9.1 学術論文

技術文章の中でも典型的のが、学術論文でしょう。この講義では学術論文の構成を通して、一般的な技術文章の書き方を学びます。

#### 9.1.1 学術論文とは

「情報処理学会」や、IEEE<sup>1</sup>、ACM<sup>2</sup>といった団体では、論文誌とよばれる雑誌を発行しています。これらの雑誌に掲載される論文は、学術論文とよばれ、広く公開され学問の発展に寄与します。学術論文と呼ばれるには、最低でも以下の二つの条件を満たしているが必要<sup>3</sup>です。

- ・専門家の間に広く公開され、誰でも読むことができるのこと。
- ・その分野の専門家による査読(その論文が、雑誌等に掲載される価値があるかどうかを審査すること)がなされること。

#### 9.1.2 学術論文の一般的な構成

学術論文の構成は、掲載される雑誌によって異なった形式が指定されます。実際に論文を執筆する際には、その指定に沿った書き方をしなければなりません。ここでは、大抵の論文に共通する、一般的な構成を書きます。学術論文に限らず、一般的な技術文章は、このような構成をとると読みやすく、また格調が高くなります。

タイトル できるだけ簡潔に、かつ論文の内容が良く分かるように書きます。

著者名 所属や連絡先を併記する場合もあります。

**概要/abstract** 論文の内容を短くまとめたものです。長さは、雑誌により随分と違いますが、英文で100語前後、日本文で500字前後を目安とするといいでしょう。ここには、目的と結論を明記しておかなくてはなりません。これを短くまとめるのは、結構大変です。

本文 本文は、大きく次の3つに分けられます。

序論 その研究の目的と、その研究の背景(その論文より前になされ、発表された研究(これを先行研究といいます)と、その研究とのかかわり)を述べます。その論文のこの後の構成を述べることもあります。

**文献リスト** 論文中で参照したことによって、冗長になり論文の論旨が不明瞭になるおそれのある長い表や証明、大きな図などは、まとめて付録として、文末に添付します。プログラムリスト等も付録にすることが多いようです。

### 9.2 科学技術文章の要件

よい技術文章は、次の6つの要件を満たしていないなくてはなりません。

1. 個別の事実、主張：各式や、定理、実験結果等が、正しくなければなりません。
2. 論理構成：論文の論理の流れに無理があつたり、混乱していてはいけません。
3. 表現：紛らわしい表現や、分かりにくい文章を書いてはいけません。
4. 新規性：その研究が、新しい事実、結論を含んでいないなくてはなりません。
5. 有用性：その研究は、役に立つものでなければなりません。
6. 整合性：論文中で論理が矛盾していたり、過去に知られている事実に反するものであつてはいけません。

このうち、1, 4, 5, 6は、どちらかといふと研究内容に関する事なので、ここではあまり触れません。ここで重要なのは、2と3です。

#### 9.2.1 良い技術文章の必要性

研究者の中には、立派な研究さえしていればそれで良いのだと考えている人も中にはいるようですが、それは明らかに間違いで、どんなに優れた研究も、それが世の中に広く公開され、理解されなければ何の意味もありません。立派な研究をしながら、なかなか世に認められない研究者の大部分は、この人に知つてもらう、理解してもらうというこの努力を怠つていよいです。つまり、良い研究をして、なおかつそれを分かりやすい形で公表できて、はじめて一人前の研究者といふわけです。

さらに、論理構造というのも大変重要です。技術文章は、良い内容を良い表現方法で、つまり、読みやすい名文を書き、図や表を駆使して見栄えをよくし、視覚的にもわかりやすくしただけで、良い文章が出来るとは限りません。このあたり、論理構造というものをもっと重視すべきです。ただし、正しくかつ理解しやすい論理構造を構築するのは大変です。幾つかの方法論もありますが、これといった決め手はないようです。やはり、自分で文章を書き、他人の文章を読んで、経験を積むといった方法にまさるものは無いでしょう。

<sup>1</sup>the Institute of Electrical and Electronics Engineers, Inc., 米国電気電子学会

<sup>2</sup>Association for Computing Machinery, 米国計算機学会

<sup>3</sup>實際には、もっと細かい条件が付されます。例えば、本学では学術論文の要件として、5項目が挙げられています。

## 9.2.2 文を書く上での注意

文章を書く上での注意を挙げておきます。

- 誰がやったことが明確にする  
ある事実、あるいは主張が、前から知られている（公知）ことか、自分が新たに発見、証明、考案したことかをはっきりさせなくてはなりません。既に知られていることが、自分が新たに発見したことのように読めたりする、その分野の専門家から見れば、単に知られている事実を列挙したにすぎない文に見えてしまいます。
- 客観的事実か考案のかを明確にする  
前項とも関係しますが、客観的事実なのか、それとも自分の考え、考案結果を述べたもののか明確にしなければなりません。
- 引用は明示する  
他の論文や、本などからその内容を引用したり、既に知られている事実を参照する時は、その引用のもととなる文献を必ず明示しなければなりません。文末の文献一覧に載せておくのはもちろんのことです。
- 根拠を明示する  
ある主張をする時は、必ずその根拠を明示しなければなりません。既に知られている事実を述べる時は、参考すべき文献がその根拠となります。前項の引用の明示は、この事とも関係します。観察、実験の結果を述べる時は、その観察なり実験が根拠となります。から、どんな条件の元で、どんな観察、実験を行ったのか明記する必要があります。根拠を述べなくともよいのは、事項の定義をする場合のみです。
- 仮定、条件をはっきりさせる  
どんな技術文章を書く時も、そこに書いてあることが、どんな条件の元で、どういう仮定をした時に成り立つことなのかははっきりさせなければなりません。仮定をはっきりさせないと、すべての場合に例外なく成り立つと思われてしまいます。大抵の場合、それは嘘をついていることを意味します。
- 用語の定義を明確にする  
技術文章の場合は、専門用語がたくさん使われます。まず書き手がそれらの用語の定義を、全てきちんと理解していなければなりません。その上で、必要な用語の定義はきちんとしておくことが必要です。もちろん、その分野の専門家が読むことを仮定している場合は、その分野で一般的に用いられている用語の定義は省略される場合が多くなります。しかし、用語の定義が多くなるほど、その文章を読んで理解できる人が少なくなります。知識や事実を一般に広く公開するという趣旨からすれば、用語の定義は、紙面に許される限り述べておくべきです。論文中で始めて使うところで、述べるのがいいでしょう。略号や略語も同様です。
- 記号の定義を明示する  
技術文章では、多くの記号が用いられますが、それらの記号は、四則演算や初等関数のような極めて一般的なものを除いて、全て定義されなければなりません。前項の用語の定義と同様に、ある専門分野で一般的に用いられる記号の定義は省略されることがありますが、専門用語の定義と違い、記号の定義は省略することは、原則として許されません<sup>4</sup>。

<sup>4</sup>例外的に、特定の組織の内部でしか読まれない文章や、紙面が極端に限られている場合は、この限りではありません

## 9.3 L<sup>A</sup>T<sub>E</sub>X の論理構造命令

L<sup>A</sup>T<sub>E</sub>X には、文章の論理構造を表すコマンドが用意されています。というより、L<sup>A</sup>T<sub>E</sub>X では、文章の論理構造を指定すれば、適当なレイアウトで出力されます。

### 9.3.1 section コマンド

既に講義資料や、演習資料の例題に出てきましたが、章立てに用いられるコマンドに、section コマンドがあります。ここでは、イタリック体の section コマンドというのを、以下に述べる幾つかのコマンドの総称として用いています。

ここでは、jarticle スタイルファイルで用いられる section コマンドを解説します。section コマンドには、次のようなものがあります。

section subsection subsubsection paragraph subparagraph

これらのコマンドの出力例は、この後の例を見てください。section コマンドは、節、小節の番号を自動的に生成します。

【入力例】

```
\section{節}
\subsection{小節}
\subsubsection{小小節}
\paragraph{段落}
\subparagraph{小段落}
\subsubparagraph{小段落}
\subsubsection{小小節}
\subsubsubsection{小小小節}
\section{節}
\subsection{小節}
\subsubsection{小小節}
各節、小節、段落のレイアウトと節番号を観察してください。
```

**出力例****1 節****1.1 小節****1.1.1 小小節**

段落 段落の内容

小段落 小段落の内容

**1.1.2 小小節****2 節****2.1 小節****2.1.1 小小節**

各節、小節、段落のレイアウトと節番号を觀察してください。

**9.3.2 学術論文の論理構成と L<sup>A</sup>T<sub>E</sub>X**

L<sup>A</sup>T<sub>E</sub>X を用いて学術論文を書くと、一般的に次のような構成になります。

```
\documentclass[jarticle]
\bibliographystyle{plain}
\title{タイトル}
\author{著者名}
\date{}

\begin{document}
\maketitle
\abstract{概要の内容}
\section{序論}
\subsection{本論文の目的}
\subsubsection{論文の目的}
\subsubsection{背景}
\subsubsection{背景の説明}
\subsubsection{本論}

\begin{tableofcontents}
\end{tableofcontents}
```

**9.3.2.1 プリアンブル**

\documentclass[jarticle] から、\begin{document} までの間を、プリアンブルと呼びます。ここには独自マクロの定義や使用するパッケージの指定を書いたりします。  
 \usepackage{graphicx}  
 \usepackage{fnewtxttext}  
 \usepackage{cite}  
 などが便利で基本的なパッケージです。

**9.3.3 論理構成に便利な使い方**

9.2.1で、論理構成法の決め手はないと述べました。しかし、論理構成を考える上で、その手助けとなる方法は幾つかあります。L<sup>A</sup>T<sub>E</sub>X の機能のなかには、論理構成を考える上で便利なものがあります。

**9.3.3.1 目次コマンド**

L<sup>A</sup>T<sub>E</sub>X では、目次を自動的に生成するコマンドがあります。目次を生成するコマンドは、\tableofcontents です。 \tableofcontents コマンドは、その場所に目次を挿入します。正しく目次を生成するには、2回、最悪の場合は3回コンパイルする必要があります。  
 目次には、節番号とタイトルが表示されます。これをア utolineプロセッサ代わりに使うことが出来ます。ア utolineプロセッサとは、文章の論理構造を決定するために、その文章の骨組みだけを最初に組み立てて、あとからその骨組みに肉付けをして文章を書いていくためのツールの総称です。L<sup>A</sup>T<sub>E</sub>X で文章を書く場合は、目次を文章の骨組みとして参照し、論理構造を考える一助に出来ます。また、文を書いている途中や、書き上げた後で、論理構造を変更することも出来ます。このあたりが、節番号や相互参照を自動的に生成する L<sup>A</sup>T<sub>E</sub>X の真骨頂といつてもいいでしょう。

**9.3.3.2 目次に関する設定**

デフォルトでは、目次に出力されるのは subsubsection までです。これでは目次をア utoline インプロセッサ代わりに使うのは不便です。L<sup>A</sup>T<sub>E</sub>X では、目次に出力される section のレベルを設定できます。プリアンブル部分に、\setcounter{tocdepth}{5} と指定すると、subparagraph まで目次に出力されます。

**9.4 文献の参照**

次に文献の参照の仕方を説明します。論文を書く場合には、参考にした文献や関連する論文を、参照文献として論文の最後にリストすることができます。これは、自分の論文が参考文献の成果の上に成立していることを表します。そして、より数多くの論文に参考文献として引用されているということが論文としての価値の指標になるという考え方がある意味で“成り立ちます。したがって、研究者の間では引用はとても厳しく扱われる場合があり、適切に引用がなされないヒュームをつけられることがあります。

## 9.4.1 BibTeX を用いた参照の方法

文献の参照は、文中で \cite というコマンドを用います。文法は、  
\cite{labelname}  
です。ここで、*labelname* は、BibTeX の文献情報ファイル内で設定したラベル名です。

### 9.4.1.1 BibTeX の文献情報ファイルの例

説明のために、文献情報ファイルは *bunkn1.bib* というファイル名（拡張子は *bib*）とします。以下の例の様に文献情報を *bunkn1.bib* に書き込みます。この例では 2 つの論文 (*article*) と 1 冊の本 (*book*) が記述されています。それぞれのラベル名は *ritchie1974unix*, *knuth1975*, *shannon1962* です。（url を文献情報として利用することも出来ます。）

```
@article{ritchie1974unix,
 title={The UNIX time-sharing system},
 author={Ritchie, D.M. and Thompson, K.},
 journal={Communications of the ACM},
 volume={17},
 number={7},
 pages={365--375},
 year={1974},
 publisher={ACM}
}

@ARTICLE{knuth1975,
 author={Knuth, D.E. and Rao, G.S.},
 journal={Computers, IEEE Transactions on},
 title={Activity in an Interleaved Memory},
 year={1975},
 month={sept.},
 volume={C-24},
 number={9},
 pages={943 - 944},
}

@book{shannon1962,
 title={The mathematical theory of communication},
 author={Shannon, C.E. and Weaver, W.},
 volume={19},
 number={9},
 year={1962},
 publisher={University of Illinois Press Urbana}
}
```

これらのラベル名を利用するためには、

```
\bibliographystyle{jplain}
\bibliography{bunkn1}
\bibliography{bunkn1,bunkn2}

と記述します。利用する bib ファイルが複数の場合は、ファイルを列挙します。 bunkn1.bib と bunkn2.bib を使うときは、
```

```
\bibliography{bunkn1,bunkn2}

と記述します。bib ファイルが複数の場合、ファイルを出力する個所（通常は最後尾）に
```

実際に文献情報を組み込んだ出力を得るためにには次の順に命令を実行します。ここでは LaTeX のソースファイルを *ronbun.tex* として説明します。

#### 1. *platex ronbun.tex*

最初に *platex* を用いてコンパイルし、*ronbun.tex* 内の *cite* コマンドで引用するラベルのリストを *ronbun.aux* ファイルに記録する。

#### 2. *pbibtex ronbun*

*ronbun.aux* 内にリストされている情報を利用し、*bunkn1.bib* から *ronbun.tex* の為の文献情報ファイル *ronbun.bbl* を読み込んで \bibliography{bunkn1} と書かれている個所に文献リストを記述したものを *ronbun.dvi* に出力する。*ronbun.aux* に文献リストの情報が書かれる。

#### 3. *platex ronbun.tex*

*ronbun.bbl* を読み込んで \bibliography{bunkn1} と書かれている個所に文献リストを記述したものを *ronbun.dvi* が出力される。

#### 4. *platex ronbun.tex*

*ronbun.aux* の文献リストを利用し *cite* の参照番号を生成した完成版を *ronbun.dvi* に出力する。

## 9.4.2 thebibliography

BibTeX を利用して生成される *bbl* ファイルの中身は、*thebibliography* 環境で記述された文献リストです。BibTeX を使用しないで、本文中に直接、*thebibliography* 環境を用いて文献リストを記述することも可能ですが、

- 論文を作成する度に文献リストを直接書き込むことになり面倒です。

- 公開されている *bib* ファイルを利用する方が便利です。

• スタイルを統一せずに書けるが、それは良くない。

と言えます。*bbl* ファイルは *cite* に応じて生成されるので、1 つの *bib* ファイルを複数の *tex* ソースで利用することができます。  
*thebibliography* 環境は以下の様に書かれます。

```
\begin{thebibliography}{widest_label}
\bibitem{labelname} bibliography

```

ここで, `widest_label` は, 通常, `{99}` や `{999}` のように指定します。これは, 各項目の文献番号に何桁の数字が必要かを示すもので, 99なら2桁, 999なら3桁の数字の欄を用意します。

標準のスタイル (`jplain`) で生成される `thebibliography` では以下の通りです。文献は, 論文と本に大きく分けられます。論文の記述は,

```
著者名, 論文タイトル, 論文誌名, Volume No., 記載ページ, 年
```

の順に, コンマで区切って書きます。論文タイトルの部分は, 通常イタリック体で書きますが, アンダーラインをつけたり, 引用符(‘や’')で括ったりもします。本の場合は,

```
著者名, 本のタイトル, 出版社, 出版社の所在地, 出版年
```

の順に, コンマで区切って書きます。本のタイトルは, 通常論文のタイトルと同じ扱いにします。場合によっては, 本の総ページ数を書いたり, 出版社の所在地を省いたりします。

## 9.5 スライド Beamer

学会や研究会等で自分の研究成果を発表する場合に、スライドを利用することができあります。`LATEX` を利用してスライドを生成する方法を紹介します。以下のサンプルを注意深く観察してください。

```
\documentclass[10pt]{beamer}

\usepackage{Madrid}
\usefonttheme{serif}
\usepackage{color} % color
\usepackage{graphicx} % image のとりこみなど
\usepackage{pst-node} % 図を書く
\usepackage{pst-node} % pdf 向きのフォント (圧縮される) 帯広め
\usepackage{pxfonts} % pdf 向きのフォント (圧縮される) 帯広め
\title{beamer サンプル}
\subtitle{literacy I}
\author{literacy 教員}
\email{lit@u-aizu.ac.jp}
\institute{会津大学}
\begin{document}
\frame{\titlepage}

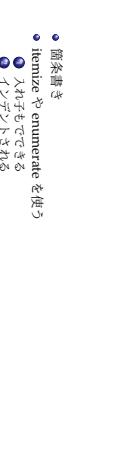
\frame{\frametitle{pst-node の例}}
\begin{figure}
\begin{center}
\begin{pspicture}(-0.5,-11.5)(8.5,0)
\dotnode(0,-3.5){D1}\rput(0,-3.2){1}
\dotnode(1,-3.5){D2}\rput(1,-3.2){2}
\dotnode(2,-3.5){D3}\rput(2,-3.2){3}
\ncline{D1}{D2}\ncline{D2}{D3}\rput(2,-4.8){graph}
\end{pspicture}
\end{center}
\end{figure}
\end{document}
```

作成したスライドを使ってプレゼンテーションを行うために PDF ファイルにしておくと多くの環境で利用できるので便利です。

```
\begin{itemize}
\item {\ttitemize} や {\ttenumerate} を使う
\item 入れ子もができる
\item インデントされる
\end{itemize}
```



January 8, 2014



January 8, 2014



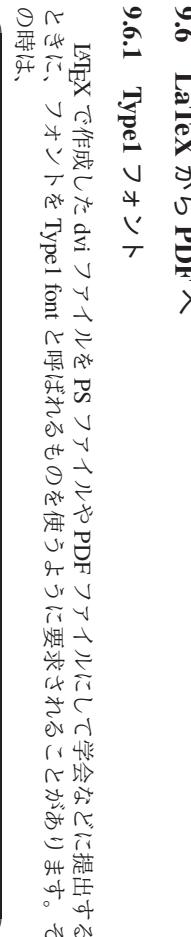
## 9.6.2 フォントの埋め込み

LaTeX で生成された dvi ファイルから PDF ファイルを生成する場合にフォントを PDF ファイルに埋め込んでおくと、表示がくずれません。PDF ファイルで使用されているフォントを調べるには pdffonts を利用します。`-f` オプションで開始ページ、`-l` オプションで終了ページを指定できるのでページ毎に解析できます。

```
pdffonts -f 10 -l 10 pdf-file.pdf
```

フォントが埋め込まれていなければ pdf を表示するアプリケーションが代りのフォントを使用する場合があり、正しく表示できないことがあります。フォントを埋め込むための PDF ファイルの作成方法は、主に以下の 3 つの方法があります。

- dvips foo.dvi -o foo.ps で ps ファイルを作成し、  
ps2pdf -dPDFSETTINGS=/prepress foo.ps で foo.pdf を作成する。
- dvips -Pdownload35 foo.dvi -o foo.ps で ps ファイルを作成し、  
ps2pdf foo.ps で foo.pdf を作成する。
- dvipdfmx foo.dvi で foo.pdf を作成する。何らかの理由で日本語フォントを用いている場合には、  
dvipdfmx -f ptx-ipa-ttf.map foo.dvi で foo.pdf を作成する。(現在のところ  
pstricks を使用した場合には dvipdfmx は利用できません。)



```
dvips -P cm file.dvi -o file.ps
```

というように、dvips で `-P cm` を指定すると、bluesky の cm (computer modern:  $\text{\LaTeX}$  の標準フォント) を使うと良いです。PDF ファイルに変換するにはさらに ps2pdf を利用します。

```
ps2pdf file.ps file.pdf
```

## 第10章 著作権およびコンピュータやネットワーク利用に関連する法律、セキュリティ

- ・ 学校や研究所など公共の施設のネットワーク接続は商用インターネットワークと比較して、接続利用料金が優遇されています。また、接続利用、接続回線の使用料は税金から支払われています。従って、公共の施設は公共のサービスを目的としてネットワークを利用するべきであるとする制限があります。従って個人的な商業行為や、他人の商業行為に関係する行為を個人的な施設で行ってはならないでしょう。
- ・ 個人的な商業行為:
  - \* fleamarket を装って通信販売をしてはいけない。
  - 他人の商業行為に関係する行為:
    - \* ホームページに商用ネットのページをリンクするなど、広告活動をしてはいけない。
- ・ 不正アクセス
  - \* 「不正アクセス行為の禁止等に関する法律」が平成 11 年 8 月 13 日に公布され平成 13 年 12 月 22 日に最終改正の後、平成 13 年 1 月 6 日から施行されています。違反の内容によっては「一年以下の懲役又は五十万円以下の罰金」あるいは「三十万円以下の罰金」に処するとあります。
- ・ 犯罪捜査のための通信傍受
  - \* 「犯罪捜査のための通信傍受に関する法律」が平成 11 年 8 月 18 日に公布され平成 13 年 12 月 12 日に最終改正の後、平成 14 年 3 月 1 日から施行されています。裁判所により傍受令状が発布された場合には通信事業者等は正当な理由がない場合は傍受を拒んではならない。(通信事業者等の協力義務)
- ・ 肖像権
  - 写真、絵画、映像、音声など、肖像となる個人の許可を得ないで、肖像を利用することはできない。
  - ・ 名誉毀損
    - 不当に個人や団体の名誉を甚だしく傷つけた場合には、名誉毀損として訴えられることがある。
  - ・ 犯罪的陳列罪
    - 犯罪的な画像や音声、文章を不特定多数の人間に配布および陳列できるような形態でネットワークに載せる事は犯罪的陳列罪で検挙される。
  - ・ 范行為の制限
    - ねずみ講と呼ばれる商売の方法は法律で禁止されており、参加者も厳しく处罚されます。

- インターネットはすでに良識のある人だけが利用しているような状態ではありません。通常の社会人が遭遇するような、様々な誘惑や勧誘に出会う可能性があります。インターネットへの接続を提供するプロバイダや、あるいは会津大学のような学校、その他の公共機関は利用者に対して接続の機会を提供するだけで、決して利用者を保護したり、あるいは利用者の行動の保証をするようなことはありません。利用者は自分自身が自身の行動に伴うリスクに対して責任をとる義務があります。
- 著作権に関するまことしやかな「例外」を翻弄みにしてはいけない。
- \* 教育用に複製しても良い。等
- 偶然みつけてしまった情報でも、それを公開した時点で違法行為となる事があります。
- ・国や文化による著作権意識の違いにも留意が必要。

#### 参考 URL

- <http://law.e-gov.go.jp/cgi-bin/idxsearch.cgi>  
法令データ提供システム、総務省 行政管理局。

平成24年に著作権法の一部が改正され、2013年1月1日より施行されています。詳細は文化庁の説明を参考にすると良いでしょう。  
[http://www.bunka.go.jp/chosakukan/24\\_houkaisei.html](http://www.bunka.go.jp/chosakukan/24_houkaisei.html)

1. いわゆる「写り込み」(付随対象著作物の利用)等に係る規定の整備
2. 国立国会図書館による図書館資料の自動公衆送信等に係る規定の整備
3. 公文書等の管理に関する法律等に基づく利用に係る規定の整備
4. 著作権等の技術的保護手段に係る規定の整備
5. 違法ダウンロードの刑事罰化に係る規定の整備

特に違法ダウンロードが刑事罰の対象として明文化されました。

#### 10.1.1 著作権関係の webpage

|                                                                   |                                                        |
|-------------------------------------------------------------------|--------------------------------------------------------|
| <a href="http://www.cric.or.jp/">http://www.cric.or.jp/</a>       | 社団法人 著作権情報センター                                         |
| <a href="http://www.accs.jp.or.jp/">http://www.accs.jp.or.jp/</a> | 社団法人 コンピュータソフトウェア著作権協会                                 |
| <a href="http://www.jasrac.or.jp/">http://www.jasrac.or.jp/</a>   | /社団法人 日本音楽著作権協会                                        |
| <a href="http://www.bunka.go.jp/">http://www.bunka.go.jp/</a>     | 文化庁                                                    |
| <a href="http://www.copyright.org/">http://www.copyright.org/</a> | Motion Picture Association<br>(アメリカ映画協会)               |
| <a href="http://www.riaa.com/">http://www.riaa.com/</a>           | Recording Industry Association of America<br>全米レコード工業会 |

#### 10.2 Security

##### ・セキュリティの要素

- 物理的セキュリティ、環境的セキュリティ  
情報システムの設置場所、設置形態の管理（物理的損傷防止、盗難防止）など
- 人的セキュリティ  
システムを利用したり管理する人の行動に関わる部分。倫理、責任、義務、権利など
- 技術的セキュリティ  
コンピュータ(ハードウェア、ソフトウェア)管理、ネットワーク管理、ユーザのアクセス制御、

##### ・ISO/IEC 13335 [? ] の技術的セキュリティについての定義

- Confidentiality (機密性・秘匿性) 権限がないものが触れないことデータ及び情報が、正当と認められるときに正当な方法で利用している正当な行為者及び手続きにのみ公開されること
- Integrity (一貫性・完全性) データ及び情報が正確かつ完全であり、その正確かつ完全な状態が保たれること
- Availability (利用可能性・可用性) データ及び情報が、正当な行為者によって要求されたときいつても、アクセス可能かつ利用可能であること
- Authenticity (信憑性・真正性) 利用者、処理方法、システム及び情報が、その実体通りに識別されることを保証すること
- Accountability (透明義務・責任追跡性) 行為の形跡から行為者を一意に特定できることを保証すること
- Reliability (信頼性) 意図した動作と結果が、実際の動作と結果に矛盾しないこと

#### OECD 8 原則

1. 収集制限の原則：個人データの収集には、制限を設けるべきである。いかなる個人データも、適法かつ公正な手段によってのみ、収集されるべきである。データ主体に知らしめるないしは、データ主体のは同意を得ることが適当な場合には、データ主体に知らしめるか、または同意を得た上で、収集されるべきである。
2. データ内容の原則：個人データは、その利用目的に適合していないければならない。個人データは、その利用目的に必要な範囲内で正確、完全であり最新なものに保たなければならない。
3. 目的明確化の原則：個人データの収集目的は、収集時よりも遅くない時点において明確化されていなければならぬ。データ収集目的を明確化した後のデータの利用は、当該収集目的の達成又は当該収集目的に矛盾しないこと。目的を変更した場合は、その変更毎に明確化された目的の達成に限定した利用とするべきである。
4. 利用制限の原則：個人データは、下記の例外を除いて、「目的明確化の原則」により明確化された目的以外の目的のために開示・利用その他の使用に供されるべきではない。

- ・データ主体の同意がある場合
- ・法律の規定による場合

5. 安全保護の原則：個人データは、その紛失もしくはデータへの不当なアクセス、データの破壊・データの不正使用・データの不正な修正・開示等の危険に対して、合理的な安全保護措置により保護されなければならない。

6. 公開の原則：個人データに係る開発、慣行、運用及び政策については、一般的な公開の政策が適用されるべきである。個人データの存在、性質及びその主要な利用目的とともにデータ管理者の識別、データ管理者の（通常の）住所を容易に明確にすることができないなければならない。

#### 7. 個人参加の原則：個人は次の権利を有する。

- ・データ管理者が自己に関するデータを有しているか否かについて、データ管理者又はその他の者から確認を得ること。
- ・自己に関するデータを、i) 合理的な期間に、ii) もし必要なら、過度にならない費用で、iii) 合理的な方法で、かつ、iv) 自己にわかりやすい形で自己に知らしめられること。
- ・上記(a)及び(b)の要求が拒否された場合には、その理由が与えられる(示される)こと。理由の開示の拒否に対して異議を申立てることができること。
- ・自己に関するデータに対して異議を申立てができること及びその異議が認められた場合には、そのデータを消去、修正、完全化、補正させること。

#### 8. 責任の原則　データ管理者は、上記の諸原則を実施するための措置に従う責任を有する。

### 10.2.1 セキュリティ維持の為のユーザの義務

コンピュータやネットワークを安全に利用するためには、利用の際の危険性について何があるかを把握して、その危険性を低減するように努力しなければなりません。パソコンやコンピュータ等の自分自身がが管理しているコンピュータを安全に利用するには次のことに注意する必要があります。

- ・セキュリティの為のソフトウェアを利用する。
- ・ファイアウォールを設定する。
- ・利用しているシステムのセキュリティホールを塞ぐ。Microsoft 社の Windows であれば "Microsoft Update" をきちんとする。
- ・ウイルスに感染している可能性のある怪しいソフトウェアを使用しない。
- ・個人情報の不正取得を狙ったスパイウェアの感染を防ぐ。(Windows Defender などを利用する。)
- ・メールによって配布されるウイルスへの対策をする。

- ・個人情報の不正取得や詐欺を狙った詐称メールや詐称ウェブページによるフィッシング(Phishing)に騙されないように注意する。
- ・USB メモリ経由で感染するウイルスがあります。USB メモリを管理の怪しいパソコンに差し込んだり、怪しい USB メモリを自分のパソコンに差し込むことは感染の危険があります。

自分が所属している組織のネットワークシステムやコンピュータを安全に利用するには次のことに注意する必要があります。

- ・パスワードは自分以外に知られてはいけない。
- ・アカウントを共有してはいけない。
- ・パスワードをタイプしている手元や口元を覗いてはいけない。
- ・ファイルの管理
  - 秘密にしたいものにはアクセス許可を出さない。
  - アクセス許可があるファイルでも共有を前提としない他人のファイルは覗かれない。見たい場合には了承を得る。
- ・所属組織のネットワーク利用規定を順守すること。

#### 10.2.1.1 会津大学情報センター(情報処理センター)利用規程

会津大学情報センターの利用者が守るべきルールが

<http://web-int.u-aizu.ac.jp/labs/istc/rpc/regulation.html>  
に掲載されています。なにか迷うがあれば、読んでみましょう。一部を転載します。

(利用に關する規律)  
第8条 利用者は、AINS 資源の利用にあたっては、次に掲げる事項を遵守しなければならない。

- 他の利用者の迷惑となる行為をしてはならない。利用など不適切な利用を行ってはならない。
- 公序良俗に反する利用及び法会、学則等に反する利用など不適切な利用を行ってはならない。
- アクセス可能かどうかにかかわらず、他の利用者の許可なく、他の利用者のファイル等の AINS 資源を利用してはならない。
- アカウントなどの利用資格を転貸してはならない。
- パスワードなどの機密を保持し、他者に開示してはならない。
- AINS 資源を利用する際、身分を偽つてはならない。
- AINS 資源を汚損・損傷してはならない。
- その他センター長が別に定める事項に違反してはならない。  
2 利用者は、所定の認証手続きを経てなされたAINS 資源の利用の結果について責任をもたねばならない。

## 第 11 章 情報管理

### 11.1 File Types

ファイルには、人間が読めるファイルと機械語などで書かれて、人間がそのままでは読めない(読むのが困難)なものがあります。

```
pros34{takafumi}114: file ex1.dvi
ex1.dvi: data
pros34{takafumi}115: file ex1.tex
ex1.tex: ascii text
```

の様に、file コマンドで、その種類を知ることができます。人間が読めないファイルを binary(バイナリ) ファイルと呼びますが、実行ファイル、画像ファイル、音声ファイルなど様々なものがあります。

### 11.2 File Convert

世の中には様々な目的のファイルが様々な format で利用されています。個々のソフトによって、利用可能なフォーマットは限られています。そこで、フォーマットの変換が必要になってしまいます。例えば、

- ・ 画像データを L<sup>A</sup>T<sub>E</sub>X のファイルに取り込みたい場合などは、eps 形式。
- ・ 画像データを HTML のファイルに取り込みたい場合などは、png 形式、または jpeg 形式。

などの様に、主な組み合わせについては、変換方法を知つておく必要があります。

#### 11.2.1 画像ファイル

画像データの保存形式には多くの種類があります。例えば ImageMagick という画像を扱うツールの convert について、「man convert」を実行すると  
`/usr/local/share/doc/ImageMagick-6.3.0/www/convert.html`  
 や `http://www.imagemagick.org/script/convert.php`

において 83 種類の画像フォーマットがリストされているのを見ることができます。これらが画像フォーマット全てを網羅しているという訳ではありませんが、現在、流通している free software では、もともと対応しているフォーマットの種類が多いものの一つです。モリを多く消費する事や、変換速度が遅いなどの欠点もありますが、convert<sup>1</sup> が便利なツールであると言うことはできるでしょう。convert が扱うファイル形式を表にしてみました。

| Type   | Description                                                                                |
|--------|--------------------------------------------------------------------------------------------|
| AVS    | AVS X image file.                                                                          |
| BIE+   | Joint Bi-level Image experts Group (JBIG) file interchange format.                         |
| BMP    | Microsoft Windows bitmap image file.                                                       |
| BMP24+ | Microsoft Windows 24-bit bitmap image file.                                                |
| CMYK   | Raw cyan, magenta, yellow, and black bytes. 他に C から始まるものは 2 種類。                            |
| DIB    | Microsoft Windows bitmap image file.                                                       |
| DICOM  | Medical image file.                                                                        |
| EPDF   | Encapsulated Portable Document Format.                                                     |
| EPI    | Adobe Encapsulated PostScript Interchange format.                                          |
| EPS    | Adobe Encapsulated PostScript file.                                                        |
| EPS2   | Adobe Level II Encapsulated PostScript file.                                               |
| EPSF   | Adobe Encapsulated PostScript file.                                                        |
| EPSI   | Adobe Encapsulated PostScript Interchange format.                                          |
| EPT    | Adobe Encapsulated PostScript Interchange format with TIFF preview.                        |
| FAX    | Group 3.                                                                                   |
| FIG    | Flexible Image Transport System.                                                           |
| FITS   | FlashPix Format.                                                                           |
| FPX    | CompuServe graphics interchange format; 8-bit color. 他に G から始まるものは 4 種類。                   |
| GIF    | CompuServe graphics interchange format; 8-bit color (version 87a).                         |
| GRAY   | Raw gray bytes. 他に G から始まるものは 2 種類。                                                        |
| HDF+   | Hierarchical Data Format.                                                                  |
| HPGL   | HP-GL plotter language. 他に H から始まるものは 2 種類。                                                |
| JBIG+  | Joint Bi-level Image experts Group file interchange format.                                |
| JPEG   | Joint Photographic Experts Group JFIF format; compressed 24-bit color. 他に J から始まるものは 1 種類。 |
| ICO    | Microsoft icon.                                                                            |
| LABEL  | text image.                                                                                |
| MIF++  | Magick image file format. MNG Multiple-image Network Graphics.                             |
| MPEG   | Motion Picture Experts Group file interchange format. 他に M から始まるものは 3 種類。                  |
| PBM    | Portable bitmap format (black and white).                                                  |
| PCD    | Photo CD. The maximum resolution written is 512x768 pixels.                                |
| PDF    | Portable Document Format.                                                                  |
| PGM+   | Portable graymap format (gray scale).                                                      |
| PICT   | Apple Macintosh QuickDraw/PICT file.                                                       |
| PIX    | Alias/Wavefront RLE image format.                                                          |

<sup>1</sup> See 'man convert': converts an input file using one image format to an output file with the same or differing image format.

| Type | Description                                                    |
|------|----------------------------------------------------------------|
| PNG  | Portable Network Graphics.                                     |
| PNM  | Portable anymap.                                               |
| PPM  | Portable pixmap format (color).                                |
| PS   | Adobe PostScript file. 他に P から始まるものは 9 種類。                     |
| PSD  | Adobe Photoshop bitmap file.                                   |
| PS2+ | Adobe Level II PostScript file.                                |
| RAD  | Radiance image format.                                         |
| RGB  | Raw red, green, and blue bytes. 他に R から始まるものは 4 種類。            |
| SGI  | Irix RGB image file. 他に S から始まるものは 2 種類。                       |
| SUN+ | SUN Rasterfile.                                                |
| TGA+ | Truevision Targa image file.                                   |
| TIFF | Tagged Image File Format. 他に T から始まるものは 5 種類。                  |
| TTF  | TrueType font file.                                            |
| UIL  | X-Motif UIL table.                                             |
| UYVY | 16bit/pixel interleaved YUV (e.g. used by AccomWSD).           |
| VIFF | Khoros Visualization image file. 他に V から始まるものは 1 種類。           |
| WIN  | select image from or display image to your computer screen.    |
| XPM  | X Windows system pixmap file (color).                          |
| XWD  | X Windows system window dump file (color). 他に X から始まるものは 3 種類。 |
| YUV  | CCIR 601 4:1:1 file.                                           |

注意  
画像ファイルを取り扱う際には、copyright に留意して下さい。

画像フォーマットが同じでも、ソフトウェアやプラットフォームによって読みない事があります。特に、tiff などの様に、フォーマットの自由度の高いものでは、ソフトウェアによっては、すべての仕様に対応していないものがあります。また、byte-order や圧縮方法の相違が原因で読みない事があります。PPM, SGI, PICT など、比較的、問題なく複数のプラットフォームで利用できます。

最近の ImageMagick の出力する ps や eps のデータは品質が悪く、そのままでは ghostscript 等の標準的なツールで取り扱えない場合があります。その様な場合には他のツールを使用します。

- bneps - Converter from PNG/JPEG/NetPBM to EPS and PDF

以下に使用例をリストします。オプションの詳しい説明は bneps --help で表示されます。

- bneps -leps2, color=no input.png output.eps  
creates an EPS file for use with LaTeX. The result after latex/dvips is to be printed on a PS2-grayscale printer.
- bneps -1ps2,m,s=A4 input.png output.ps  
creates a print file which can be send directly to a printer. The image optimally fits an A4 page.
- bneps -1pdf input.png output.pdf  
creates a PDF-file for use in a \*.tex document which is processed by pdflatex. This is useful for images containing an alpha channel.

## 11.2.2 音声ファイル

音声データの保存形式にも多くの種類があります。例えば'man sox'<sup>2</sup> を実行すれば種類の音声フォーマットがリストされているのを見ることができます。最近は、比較的、AIFF フォーマットが多く使われる用です。画像ほど多くはありませんが、audio のフォーマットも増えてきています。最近では、mp3(MPEG Layer3) も良く用いられ、多くのオーディオ装置でも利用されています。

注意  
AUDIO ファイルを取り扱う際には、copyright に留意して下さい。

| Type  | Description                                                                                                                                                  |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| .aiff | AIFF files used on Apple IIc/IIgs and SGI.                                                                                                                   |
| .au   | SUN Microsystems AU files.                                                                                                                                   |
| .hcom | Macintosh HCOM files.                                                                                                                                        |
| .ogg  | Ogg Vorbis Compressed Audio.                                                                                                                                 |
| .raw  | Raw files (no header).                                                                                                                                       |
| .ub,  | These are several suffixes which serve as a short-hand for raw files with a given size                                                                       |
| .sb,  | and style.                                                                                                                                                   |
| .uw,  |                                                                                                                                                              |
| .sw,  |                                                                                                                                                              |
| .ul   |                                                                                                                                                              |
| .sf   | IRCAM Sound Files.                                                                                                                                           |
| .voc  | Sound Blaster VOC files.                                                                                                                                     |
| .auto | This is a "meta-type": specifying this type for an input file triggers some code that tries to guess the real type by looking for magic words in the header. |
| .cdr  | CD-R CD-R files are used in mastering music Compact Disks.                                                                                                   |
| .dat  | Text Data files                                                                                                                                              |
| .smp  | Turtle Beach SampleVision files.                                                                                                                             |
| .wav  | Windows 3.0 .WAV RIFF files.                                                                                                                                 |
| .maud | An Amiga format                                                                                                                                              |
| .vwe  | Psiion 8-bit alaw                                                                                                                                            |
| .mp3  | MP3 Compressed Audio                                                                                                                                         |

## 11.2.3 メールへの添付ファイルについて

実行形式のプログラムや画像や音声のファイルをメールやニュースに発信する場合にはすべてのデータは印字可能な ASCII 文字列に変換されテキストデータとして添付されなければなりません。7bit にしなければならない事に注意して下さい。  
実際にメールのメッセージファイルをテキスト表示させるとすべてがASCII文字列で記述されていることがわかります。これを適切なツールで扱えば元のファイルに戻すことができます。

添付ファイルは必要な場合にのみ利用します。以下の場合などは添付を避けましょう。

- ファイルサイズの大きなファイル → メール以外の方針を使用する。

<sup>2</sup>See 'man sox': SOund eXchange - universal sound sample translator

- Web や Netnews でアクセスできるようなファイル → メールに URL 等のポインタを書く。

### 11.2.3.1 MIME

ASCII 文字列に変換するための方法は MIME<sup>3</sup> インターネットメールフォーマット (RFC2046/2045) として規格化されています。そして多くのアプリケーションは、この規格に従ったメッセージを生成、処理するようになります。Thunderbird や Sylpheed は MIME に対応したメールツールであり、添付ファイルの扱いも解りやすく使い勝手が良いでしょう。

### 11.2.3.2 他のツール

mime メッセージファイルを処理する CUI のプログラム reformime があります。mime メッセージファイルから添付ファイルを抽出する場合は次の様に実行します。

```
cat file | reformime -x
```

### 11.2.3.3 uuencode

MIME が規格化される以前は uuencode<sup>4</sup> を利用して画像や音声、プログラム等のバイナリファイルをテキスト形式に変換してメールやニュースのために利用していました。uuencode は、標準でなく、消滅しつつある、などと言われていますが、現在でも uuencode を利用したメッセージは一部で利用されています。これをを利用して得られるテキストデータは以下のように begin で始まり end で終了します。

```
begin 644 login.jpg
MjU/WS7^A]3D4]I>ZR6\FD4TMIS('#[[93T,G"$IV@F3-58/T70^@$-!P;R-
M0;O5G)C#'+*~DD8A!-&I=XR8H11>[Q\G0;FNRHY/Y]A<4"?F7C-$T]A-
...
[中略]
...
MR06\Q2]IVSTM?15-(K8;:8(^0."]T^5V-3Z1W(HWS;Z)"[9(=?1@07<E$7)&
end
```

これを Mail などのメッセージに取り込んで利用します。MIME encoding されたファイルを用意に変換し、確認せずに表示などを行うと、思わぬ結果になることがあります。PostScript などは、一度、ファイルに変換してから、安全なツールで表示をするようにして下さい。

### 11.2.4 PostScript

PostScript(PS) は、もともとは、Adobe 社のプリンタの出力用の言語（ページ記述言語）でしたが、様々な場面で、原稿や図の最終仕上がり等のファイルフォーマットとして用いられています。現在、最も良く使われているのは、level3 ですが、level2 もまだ多くの場所で使われています。

<sup>3</sup>See 'man mime': Multipurpose Internet Mail Extensions

<sup>4</sup>uuencode, uudecode - encode a binary file, or decode its encoded representation

### 11.2.5 EPS

Encapsulated Postscript の事です。派生しているフォーマットがいくつもあるので、注意が必要です。EPS ファイルは、普通の PS(PostScript) ファイルに、preview 用の画像や、L<sup>A</sup>T<sub>E</sub>Xなどのソフトで読み込むときに必要な、サイズや位置の情報を含んでいます。このサイズの情報は、BoundingBox という部分に次のように書かれていて、直接 more 等で確認できます。

```
%%BoundingBox: 38 347 607 774
```

左から lower-left-x lower-left-y upper-right-x upper-right-y の座標値です。或は gs を使って表示することもできます。

```
gs -sDEVICE=bbox file.ps
```

しかし、EPS を生成するソフトによって、余白の取り方が異なっていたり、中には、サイズを誤って出力するものもあり、トラブルの原因になっています。これを解決するには、eps2eps コマンドを使います。

```
eps2eps input.eps output.eps
```

使用できるオプションは man eps2eps で確認できます。

```
Ghostscript Tools
NAME
ps2ps, eps2eps - Ghostscript PostScript "distiller"
```

```
SYNOPSIS
 ps2ps [options] input.ps output.ps
 eps2eps [options] input.eps output.eps
```

```
DESCRIPTION
 ps2ps uses gs(1) to convert PostScript(tm) file "input.ps"
 to "output.ps". Normally the output is allowed to use
 PostScript Level 2 or Level 3 constructs, but the -dlang-
 ageLevel=1 option restricts the output to Level 1.
 eps2eps performs the equivalent optimization for Encapsu-
 lated PostScript (EPS) files.
```

解像度を高くする場合は以下のようにします。

```
eps2eps -r6000 input.eps output.eps
```

## 11.2.6 互換性

ポストスクリプトも進化してきている事、また、構文が必ずしもわかりやすく無いことから、主に古いフリーソフトによる出力ファイルが、新しいPSのプリンタで出力できない事があります。**ghostview**の様な、互換ソフトは、比較的、エラー処理が甘いので、そういうソフトで出力できるからといって、純正のPSプリンタで出力できるとは、限りません。また、PSファイルは、多数のフォントを使っている事が多いのですが、対応しているフォントが無いと、出力されなかったり、文字が化けたりします。こういったトラブルの際に、使える一つの方法を紹介しておきます。

```
gs -dBATCH -dNOPAUSE -s -sDEVICE=psmono -soutputFile=out.ps in.ps
```

これで、in.ps から、out.ps に変換されます。常に有効とは限りませんが、多くの場面で有効です。また、level2 のファイルを level1 のプリンタにかける時にも使えます。

## 11.2.7 PDF

「フォントが無いと出力できない」というPSの欠点は、様々な場面で、障害となっていました。また、PSファイルの一部を編集するのは、かなり面倒でした。これは、もともとがPSがプリンタ用の言語という性格に起因するものですが、最近のDTP(DeskTop Publishing)事情にはそぐわないものになってきています。そこで、AdobeはAcrobat/PDF(Portable Document Format)というファイルフォーマットを作りました。これは、フォントが無くて表示できないという事もなく、PSでは困難ないマニュアルの配布などに広く使われるようになりました。そのため、非常に短い時間で、PDFを初めあらゆるプラットフォームでPDFが使用されます。

### 11.2.7.1 PDFの表示

表示には、**pdfstudio**、**xpdf**や**evince**を利用します。その他のツールとして**gv**等も使用できます。Mac OS X 端末にはAcrobat Readerが導入されおり、利用できます。

### 11.2.7.2 PDFの生成

PDF ファイルを生成するには、**pdfstudio**や**ps2pdf**を利用できます。

### 11.2.7.3 文字列検索

**pdfstudio**、**evince**、**xpdf**や**Acrobat Reader**ではPDF文書内の文字列検索ができます。

## 11.3 Compress

ファイルもたまると、ディスクの容量を圧迫します。画像ファイル、AUDIOファイルやプログラムのソースファイルなどは、圧縮をすることで、そのサイズを小さくすることができます。こういった目的には、通常 lossless 圧縮法を用います。これは、一度圧縮をしたファイルを extend した時に、元にもどる圧縮方法です。逆に、見た目や聞いた時の感じがほぼ同じであれば良い様な目的には、画像や音声ファイルを lossy 圧縮します。これは、不要だと思われる情報を落として、圧縮率をあげているので、extend しても、完全にもどにはもどりません。注意が必要です。

**compress**, **uncompress**  
compress コマンドで圧縮をかけ、uncompress で圧縮を解きます。

**gzip**, **gunzip**, **zcat**  
gzip コマンドで圧縮をかけ、gunzip で圧縮を解きます。

**bzip2**, **bunzip2**, **bzcat**  
bzip2 コマンドで圧縮をかけ、bunzip2 で圧縮を解きます。

**xz**, **unxz**, **xzcat**  
xz コマンドで圧縮をかけ、unxz で圧縮を解きます。

**zcat** コマンドは、圧縮したファイルを表示したり、他のコマンドの標準入力に送るのに便利です。OS付属のものと、GNUのものでは、対応できるフォーマットに差があります。**/usr/local/gnu/bin/zcat** を使うと良いでしょう。

## 11.3.1 アーカイブ

たくさんのが細かいファイルを持っていると、それを管理するのが難しくなるだけではなく、ディスクの容量も無駄に消費することになります。そんな時には、archiveして、圧縮tar)を使うのも良いでしょう。

```
tar cf littex.tar litt*tex
compress littex.tar
tar cf littex.tar litt*tex
gzip littex.tar
gtar zcf littex.tgz litt*ps
```

## 11.4 Backup Method

### 11.4.1 リムーバブルメディア

ホームディレクトリのハードディスクは限られた資源です。

- ・頻繁にアクセスする必要のないファイルは、USBストレージ(USBメモリー、外付けHDD等)に、バックアップする様に心掛けましょう。USBストレージの機器は各自で用意したものを利用します。

- ・また、オペレーションミスなどで、ファイルを誤って消した場合に備える意味で、必要に応じて、バックアップをとる事が大事です。USBメモリーの使い方は、ISTCのホームページに説明の記事があるので参考にしましょう。

<http://web-int.u-aizu.ac.jp/labs/istc/isc/release/howto/howto.html#usb>

- ・USBのコネクタは耐久性が低く壊れやすいので、無理に抜き差ししてはいけません。500回程度の抜き差しが使用限界と言われていますので、WS本体のUSBコネクタはなるべく使用しないでください。演習室のキーボードがUSBハブとしての機能を提供していますので、キーボードにUSBのコネクタがあれば、これを利用してください。Mac端末にはUSBハブが用意されています。

- ・以下にUSBストレージの利用の大まかな流れを説明します。

- ・装着(Mount)  
まず、USBストレージを差し込みます。すると自動的に装置のファイルシステムをMountします。これには、装置の容量により、しばらく時間がかかることがあります。

- 演習室1.2、CALL1.2、中講義室、大講義室のSunray端末を使用している場合はターミナル内で

```
% /opt/SUNWmut/bin/utdiskadm -l
Device Partition Mount Path

disk1 disk1p0 /tmp/SUNWmut/mnt/a-fuji/tu/noname
disk1 disk1p1
```

を実行し、マウントを確認できます。

- Mac OS X端末の場合はデスクトップにVolumeフォルダが表示されます。

- ・利用  
ファイルなどの書き込みは通常通りできますが、フォルダのアクセス制限がotherに対するも読み出し可、書き込み可となる場合があるので個人情報の扱いには注意しましょう。

- ・アンマウント(Unmount)

- Sunray端末の場合は /opt/SUNWmut/bin/utdiskadm -l コマンドで得られるデータベース情報を利用してアンマウントします。

```
/opt/SUNWmut/bin/utdiskadm -e disk1
```

等とし、アンマウントします。

- Mac OS X端末の場合は「フォルダマイコンをゴミ箱に入れる」か、ファインダーの「アンドウを開いて ejectするか、マウスの右ボタンをクリックしてメニューを出して ejectを選択するかのいずれかの方法でアンマウントします。

- ・取り外しアンマウントが完了した後に取り外します。

リムーバブルメディアの最大の欠点は置き忘れや、紛失、盗難が発生することです。個人情報を記録したメディアは特に注意して、無くさないように、そして、他人に利用されないように慎重に取り扱ってください。

#### 11.4.1.1 USBの規格

USBの規格は1.1と2.0、そして3.0があります。USB1.1はキーボードやマウス等の通信速度の低いデバイスを扱う規格です。大容量のUSBメモリやHDDを扱うのは不適切です。Solaris WSやMac OS X端末のUSBは2.0なので、比較的大容量のストレージを扱うことができます。演習室にはUSB3.0の機器はまだ導入されていません。

#### 11.4.2 オンライнстレージ(Online Storage)

記憶媒体を手に持たなくともインターネットで提供されているオンラインストレージを利用する事で、バックアップをしておくことが可能ですが。ただし、不利益を被らないためには、利用の規則や免責事項などを把握して、より安全なオンラインストレージを選択する必要があります。オンラインストレージの利点はネットワーク端末があればどこでも利用可能だということです。USBメモリ等のリムーバブルストレージとは異なり、紛失や盗難の危険性は高くありません。ただし、パスワード等でアクセスを管理するのでパスワードを取られた場合は危険です。ネットカフェなどの端末ではキーロガーや等を仕掛けられているとパスワードを記録される危険性があります。

- ・Google Drive <http://drive.google.com/> (無料)
- ・Windows OneDrive <http://onedrive.live.com/> (無料)
- ・Yahoo ポップス <http://box.yahoo.co.jp/> (無料)
- ・Dropbox [www.dropbox.com](http://www.dropbox.com) (無料)

### 11.5 Web Browser Setting

Webのブラウザ(Browser)には、いくつかの設定が必要です。

- ・表示に使うアプリケーションの設定
- ・postscriptの表示や画像の表示に用いるApplicationの設定

cacheはウェブの閲覧内容をディスクに保存しておいて、再び同じページを開いたときにcacheの内容を再利用して閲覧の速度を見掛け上で速くする為のものです。この容量が大きすぎるとホームディレクトリのディスク使用量が無自覚のまま大きくなるので、適切な大きさに設定することが必要です。

## 11.6 その他

### 11.6.1 ファイル通信機能 **ftp**

ネットワークで接続されたコンピュータ同士で、ファイルの交換が出来ます。このとき使われるのが、**ftp**(File Transfer Program)です。ファイルを交換する、相手先のコンピュータにアカウントを持っている場合は、接続して、ログイン名とパスワードを入力すれば、ファイルのやり取りが出来るようになります。これにたいして、アカウントを持っている、持っていない、にかかわらず、だれでも **ftp** できるコンピュータも存在します。この形式の **ftp** を **anonymous** にして、パスワードに、メールアドレス(例えば、s1250000@u-aizu.ac.jp)を入力します。本学には、**ftp.u-aizu.ac.jp** という名前の、**anonymous** **ftp** のホストがあります。ここにはいろいろ有用なソフトウェアがあります。有効に利用してください。

### 11.6.2 MagicPoint

**mpg** は X Window System 用のプレゼンテーションツールです。**mpg2ps** でスライドイメージを postscript に出力できます。MagicPoint には 商用の物にはない便利な機能があります。また、plain text ファイルを入力ファイルとして使っているので、通常のエディタで編集できます。また、フォーマットが、ブラックボックスな商用のものでは、バージョンによって互換性がないこともあります。有益だと思われます。(フォーマットの解析も可能ですし、様々なメディアで解析情報が得られる事もありますが、copyright に抵触する事もあり、要注意です。)

### 11.6.3 MATLAB

**MATLAB** は、信号処理、画像処理、制御系設計、最適化、シミュレーションなどに使うことを想定して作られたソフトウェアで、高度な行列演算機能とシミュレーションを簡単に実現する機能を有しています。無線 LANなどのシミュレーションなどにも利用できます。MATLAB の互換ないしは、その代替ソフトに、**Octave**<sup>5</sup> や **Scilab**<sup>6</sup> があります。Scilab は Mac OSX 端末で利用できます。Octave は Solaris 端末と Mac OSX 端末の両方で利用できます。

### 11.6.4 Maxima

**Maxima** は、1960 年代後半に MIT で開発されたコンピュータ代数諸理系 **Macsyma** の子孫です。GPL の下开源ソースソフトウェアとして公開され、一般に提供されています。Windows, Linux, Mac OSX でも利用できます。

### 11.6.5 S-plus(統計ソフト)

統計ソフトとして有名な **S** の互換ソフトに、**R**<sup>7</sup> があります。通常の統計処理はほとんどこれで可能ですが、

## 11.6.6 オフィスツール

### 11.6.6.1 互換ソフト

Microsoft 社のオフィスツールである Word, Excel, Powerpoint 等の保存ファイルは本来 Microsoft 社の Office で処理するべきですが、Solaris や Linux などでは Office は動作しません。したがって、OpenOffice 等の互換性のあるソフトウェアを使用します。ただし、互換性は充分かも知れませんが、完全ではないので、表示が崩れたりします。

### 11.6.6.2 MS-Office

Mac OS X 端末では MS-Office が利用できます。OpenOffice も利用できます。

### 11.6.6.3 Microsoft Campus Agreement

会津大学では Microsoft Campus Agreement を契約しています。それにより、以下のことが可能です。

- 大学の職員学生は、売店で 1 ライセンスのみ MS-Office を安価(1000 円程度)で購入し、個人の Windows や Mac で利用可能。
- 研究室の Windows や Mac 向けには、情報センターが貸し出す MS-Office インストールメディアを利用して Ms-Office を利用可能。

<sup>5</sup>See 'usr/local/octave/info/octave.info': Interactive language for numerical computations.

<sup>6</sup>See 'usr/local/scilab/' and <http://www-rocs.jinr.ru/scilab/>: A scientific software package for numerical computations.

<sup>7</sup>See 'man R': a language for data analysis and graphics. See also <http://www.r-project.org/>

## 第12章 シェル(その1)

### 12.1 シェルの設定ファイル

ホームディレクトリにおける設定ファイルによって、個人個人が自分に好みのシェルの操作環境を作ることができます。

#### 12.1.1 .bashrc

ログインシェルが bash の場合はターミナルを開いたときに /etc/profile が実行され、次に { \$HOME/.bash\_profile, \$HOME/.login \$HOME/.profile } の何れかが実行され、その後でターミナルに bash の操作プロンプトが表示されます。これらのファイルの内容は、ユーザがキーボードで打ち込んだと同様に実行されます。この設定ファイルが無い場合はターミナルを起動する度に、シェル変数やエイリアスを定義するためにタイピングをすることになり、実際には記憶などできないので、不可能です。ログアウトする時には \$HOME/.bash\_logout が実行されます。設定ファイルが無い場合には実行は /etc/profile のみが実行されます。

新たに bash を起動すると \$HOME/.bashrc に書かれた命令が実行され bash の操作プロンプトが表示されます。会津大学では学生には \$HOME/.profile と \$HOME/.bashrc を配布します。

環境変数の設定は、.profile か、.bashrc に記述します。通常は差異が出ないようにするために、.profile 中に .bashrc の実行を記述します。これらの設定ファイルはユーザ stdskel のホームディレクトリ (~stdskel/) に置いてあるので、参考にして下さい。これらのファイルに設定ミスがあると、そのあとの作業全般に悪影響を及ぼします。また、これらファイルの設定次第で、いくらでも快適な作業環境を作りだす事ができます。

#### 12.1.2 .cshrc

ログインシェルが csh や tcsh の場合はターミナルを開いたときに \$HOME/.login は login した時に一回だけ実行され、端末の状態の設定は、一回限りで十分なので、.login に記述し、実行します。\$HOME/.cshrc は、csh や tcsh が起動される度に実行されます。csh と区別して tcsh を利用する場合には \$HOME/.tcshrc を使用します。ログアウトする時には \$HOME/.logout が実行されます。

### 12.2 シェル変数

シェルが利用する変数です。bash では代入式で定義や値を設定します。

umask

| umask | ファイル許可ビット | ディレクトリ許可ビット |
|-------|-----------|-------------|
| 0     | rwx-      | rwx         |
| 1     | rwx-      | rwx         |
| 2     | r-        | r-x         |
| 3     | r-        | r-          |
| 4     | -w-       | -wx         |
| 5     | -w-       | -w-         |
| 6     | -         | -x          |
| 7     | -         | -           |

umask を実行すると、あなたの作成するファイルの mode の default(既定/標準) 値を設定できます。これを利用することによって、ファイルを作成する度にいちいち chmod する必要が減ります。例えば、umask 022 となっていると、作成したファイルは、chmod で、g-w と o-w を指定したのと同じです。chmod で、数字で指定する時と、bit が反転することに注意して下さい。chmod 755 をしたのと同じ効果が umask 022 で得られます。但し、これは、ファイル作成時のことですので、一度 mode を変えたファイルをセーブする時には適用されません。新たにファイルを作成した時に、ファイルのモードを設定しないでもすむようにしている(だけ)のです。

umask を 000 とすると、新規作成ファイルについては、666 (rwx-rwxrwx)、新規作成ディレクトリについては、777 (rwxrwxrwx) と設定されます。

#### umask

をコマンドラインで (=ターミナル内でということ) 実行して下さい。現在の設定がわかります。次に、

```
mkdir lit2.a
umask 002
mkdir lit2.b
```

として下さい。では、出来たディレクトリのモードがどうなっているでしょうか、まず、予想してみて下さい。

```
ls -ld lit2.a
ls -ld lit2.b
umask
```

を実行して下さい。その結果について、予想はあっていましたか?

umask を使うことで、自分のファイルを自衛できますが、password を盗まれたら防ぐことはできません。password の管理には、十分気を付けて下さい。

皆さんのは、他人ファイルやディレクトリを覗いたり、書換えるようなことをしないようにして下さい。これは、コンピュータの世界の最低限のマナーの一つです。

PATH みなさんが、普段、何気なく実行しているコマンドのファイルは、様々なディレクトリに格納されています。path はそれらを探し出して実行するときに参照する変数です。

```
PATH=/usr/local/gcc/bin:/usr/local/perl5/bin
```

という様に指定します。

一行を越えるときには、\を行の終りに入れると次の行に続けることができます。このpathの設定がないと、例えば、emacsを使いたい場合には/usr/local/bin/emacsというように、常に絶対pathでコマンドを指定しなければなりません。この設定をうまくやつておけば、コマンドを実行する時に、余計なディレクトリ名まで指定しなくてもすむようになります。カレントディレクトリ(.)は最後尾に指定するが、PATHから除いておいた方がいいでしょう。カレントディレクトリ(.)をPATHから除いた環境でカレントディレクトリにあるコマンドを実行する時には、

```
./command
```

の様に「./」をつけて実行します。

自分で作成したコマンドも置き場所を決めておくことが大切です。通常は、自分のホーム下にbinというディレクトリを作ります。そして、PATHには、\${HOME}/binを追加しておきます。カレントディレクトリのコマンドを使う必要性は低いはずです。カレントディレクトリのコマンドは、誰が作ったのか不明なことがありますし、同じ名前でも違った動きをするかもしないので、大変危険です。

**alias** aliasは、別名という意味です。良く使うコマンドで長い名前をもつたものに省略名をつけたり、いつも使うオプションをつけて、コマンド名を再定義します。

```
alias rm='/bin/rm -i'
```

を入れておくと、rmとコマンドを打ち込むと、-iオプションをつけてrmコマンドが実行されます。これは、うっかり大事なファイルを消してしまったのを防ぐのに有効です。シェル変数とその値を環境変数として利用するためには変数をexportします。

```
export PATH
```

exportを引数なしで実行した場合は設定されている全ての環境変数を出力します。envでも同様の出力を得られます。

**stty** ターミナルで使用するterminal line settingsを設定します。現在の設定を表示するにはstty -aを実行します。

**コマンド実行履歴 history** HISTSIZEという変数を設定しておくことでコマンドの履歴の記憶数を設定します。

```
export HISTSIZE=1000
export HISTFILESIZE=50
```

ターミナル内でhistoryコマンドを実行するとコマンドの履歴を参照することができます。

**IGNOREEOF** IGNOREEOFを入れておくと、コマンドラインでeof(ctrl-d)を打ったときに、logoutしないで無視(ignore)してくれます。この設定を除いておくと、うっかり、ctrl-dを打つとlogoutしてしまいます。

**MANPATH** 困った時のmanコマンドですが、これも manualの書かれているファイルの格納場所は一つとは限りません。

```
man -M マニュアルのあるパス名
```

一回限りであれば、-Mオプションで指定できますが、よく見るものについては、指定しておいた方がいいでしょう。'('バッククォート')と'('バッククォート')で囲むことによって、コマンドの返してくる文字列をシェル変数に代入することができます。

```
MACHINE='uname -s -r'
```

たとえば、これを実行することによって、uname -s コマンドで帰ってくるOSネームとuname -r コマンドで帰ってくるリビジョンを MACHINEという名前のシェル変数に代入できます。

## 12.2.1 .bashrc の実例

新1年生に最初に配布されている.bashrc(`~stdskel/.bashrc`と同一です。これを参考にします)。

```
#####
$HOME/.bashrc file
Last Updated: 2017-02-22
#
The most recent file is kept under ~stdskel/.
You may revise your files by copying from there.
URL : http://web-int.u-aizu.ac.jp/~stdskel/stdskel.html
#
The University of Aizu Information Network System (AINS)
Information System and Technology Center (ISTC)
E-mail: std-help@u-aizu.ac.jp
#####
MACHINE='uname -s -r'
umask 022
PATH=/usr/local/gcc/bin:/usr/local/perl5/bin
PATH=$PATH:/usr/local/texlive/bin
PATH=$PATH:/usr/local/bin:/usr/local/gnu/bin:/usr/local/java/jdk/bin
case "$MACHINE" in
 # for Solaris 2.x
 SunOS-5.11*) PATH=$PATH:/usr/openwin/bin:/usr/bin:/usr/sbin
 PATH=$PATH:/usr/local/SolarisStudio/bin:/usr/ccs/bin
 MANPATH=/usr/local/perl5/man:/usr/local/man
```

```
MANPATH=$MANPATH:/usr/local/share/man:/usr/local/gnu/man
MANPATH=$MANPATH:/usr/local/gnu/share/man:/usr/share/man
MANPATH=$MANPATH:/usr/local/SolarisStudio/man
```

```
for Apple Mac OS X 10.x
Darwin-*) PATH=$PATH:/usr/X11/bin:/usr/bin:/bin:/usr/sbin:/sbin
```

```
MANPATH=/usr/local/man:/usr/local/share/man:/usr/local/gnu/man
```

```
;#
for Linux RHEL
Linux-*) PATH=/usr/X11R6/bin:/usr/bin:/bin:/usr/sbin:/sbin
```

```
MANPATH=/usr/share/man
```

```
;;
esac
```

```
set site path
```

```
#set PATH=$PATH:$HOME/bin
```

```
....
```

```
if [$?USER == 0 || $?prompt == 0]; then
exit
fi
```

```
LC_COLLATE=C
```

```
export PATH LC_COLLATE
EDITOR=vi
export EDITOR
```

```
unset OS REV
if ["$SHELL" = "/bin/csh" -o "$SHELL" = "/usr/local/bin/tcsh"]; then
 eval `env SHELL=/bin/sh tset -rs`
else
 eval `tset -rs`
fi
stty erase ^H kill ^U
```

```
set this for all shells
export HISTSIZE=100
export HISTFILESIZE=50
export IGNOREEOF=3
#PS1="# \ht 'whoami'\! \$ "
PS1="\h:\w \u\!\$ "
aliases for all shells
#function cd() {
command cd @@
command pwd
#}
#alias cp='cp -i'
#alias mv='mv -i'
#alias rm='rm -i'
#alias ls='ls -F'
#alias ps='ps -ef'
```

### 12.3 シェルとは

UNIXというOSのもとで実行中のプログラムのことをプロセスと言います。マルチユーザー／マルチタスクのOSですから、同時にいくつものプロセスが活動しています。psコマンドであなたの使っているワークステーションで活動中のプロセスを知ることができます。例えれば次のようにするとpsコマンドを起動したユーザ自身に関連するすべてのプロセスの一覧を得ることができます。

```
% /usr/bin/ps -ef|grep a-fujitu|more
a-fujitu 17136 1 0 Feb 01 ? 0:01 /usr/lib/notification-daemon
a-fujitu 11618 11529 0 Jan 31 ? 0:00 iim-xbe
a-fujitu 11390 10055 0 Jan 31 ? 0:00 gnome-session
a-fujitu 12346 1 0 Jan 31 ? 1:38 gnome-terminal
a-fujitu 11529 1 0 Jan 31 ? 0:00 /usr/bin/iimx -iimd
a-fujitu 11691 1 0 Jan 31 ? 0:00 /usr/lib/gvfsd-metadata
a-fujitu 24783 12346 0 Jan 31 pts/17 0:00 -csh
a-fujitu 11670 11390 0 Jan 31 ? 0:05 gnome-power-manager
....
```

この一覧に入っていない重要なシステムサービスを行なっているプロセスもたくさん働いていて、それらも表示させたいときは次のようにします。

```
% /usr/bin/ps -ef | sort +1 -n | more
UID PID PPID C STIME TTY TIME CMD
root 0 0 0 Dec 12 ? 0:02 sched
root 1 0 0 Dec 12 ? 0:34 /usr/sbin/init
root 2 0 0 Dec 12 ? 0:43 pageout
root 3 0 0 Dec 12 ? 1131:40 fsflush
root 5 0 0 Dec 12 ? 33:16 zpool-rpool
root 6 0 0 Dec 12 ? 2:52 kmem_task
root 7 0 0 Dec 12 ? 1:26 intrd
root 8 0 0 Dec 12 ? 0:11 vmtasks
root 11 1 0 Dec 12 ? 2:31 /lib/svc/bin/svc.startd
root 13 1 0 Dec 12 ? 98:49 /lib/svc/bin/svc.configd
dladm 46 1 0 Dec 12 ? 0:32 /usr/sbin/dlmgmtd
netcfg 50 1 0 Dec 12 ? 0:28 /lib/inet/netcfgd
daemon 87 1 0 Dec 12 ? 0:05 /lib/crypto/kcf
netadm 99 1 0 Dec 12 ? 0:32 /lib/inet/iphgntd
....
```

UNIXのシェルはOS側から見れば単なる1つのプロセスに過ぎません。Bシェル、Cシェル、その他便利なシェルがたくさんあります、ユーザの好みに応じて取り替えることができます。

### 12.4 シェルの対話機能 — Bashを中心に

プログラミング言語としてのシェルの特徴はファイルシステムの操作や、プロセス間協同のコントロールに最適な形にチューニングされたプログラミング言語であると言うことができます。

## 12.4.1 パイプラインの生成と出力のリダイレクト

パイプラインの生成と出力のリダイレクトはシェルのもっとも重要な機能です。そういうシェルを持っていたからこそこれだけUNIXが普及したと言うこともできます。

|         |                             |
|---------|-----------------------------|
| >file   | 標準出力を file に切り換える           |
| >>file  | 標準出力を file に追加する            |
| >& file | 標準出力と標準エラー出力を file に切り換える   |
| p1   p2 | プロセス p1 の出力を p2 の入力として実行する。 |

表12.1. リダイレクトとパイプ

入力のリダイレクトは表では省略しています。

## 12.4.2 ファイル名の展開

ファイル名を指定するのに特別の意味を持つ文字があります。これはシェルのメタキャラクタと呼ばれるもの的一部分です。これらは特にワイルドカード文字と呼ばることがあります、UNIXの他のツールと同じように正規表現を扱うためのものです。コマンドライン上では正規表現はそれにマッチするファイル名に展開されます。ただしシェルの解釈する正規表現は正確には正規表現とは呼べない特殊な部分があります。

|               |                           |
|---------------|---------------------------|
| ?             | 任意の1文字とマッチする              |
| [list]        | list中の1文字とマッチする           |
| [lower-upper] | lowerとupperの範囲にある文字とマッチする |

「?」や「\*」の意味が正規表現の場合とは少し違うのです。「a\*」という表現はシェルではaという文字の後に0個以上の勝手な文字が続くと解釈されますが、正規表現ではaという文字の0個以上の繰り返しと解釈されます。また、「.」は正規表現では任意の1文字とマッチしますが、シェルでは「?」がその代わりをします。

```
aizu% ls
```

```
dict1 dict2 dict3 sample1 sample2
```

のような状況の時に

```
cat dict[1-3] > newdict
```

と打ち込むと

```
cat dict1 dict2 dict3 > newdict
```

と打ち込んだのと同等です。ターピングの労が省ける訳です。catはファイル連結のコマンドだから、ファイルdict1, dict2, dict3を一つのファイルnewdictにまとめることがあります。

```
ls /bin/t*
```

とすれば /bin 中の t で始まるファイルの一覧を表示することになります。

```
chmod u+x *
```

とすると、カレントディレクトリにあるすべてのファイルがユーザ本人にとって、実行可能になります、と言つて大筋は正しい。しかし、\*は1文字目の「.」だけは例外としてマッチしてくれません。例えば.loginや.logoutをマッチさせなければ次のようにになります。

```
chmod u+x . * *
```

ここでシェルの解釈する正規表現に例外があることに気づくと思います。シェルにおいては「.」(ピリオド)は普通の文字として扱われます。その代りに「?」が任意の1文字にマッチします。さらに「\*」にはファイル名先頭の「.」はマッチしません。

## 12.4.3 シェル変数

変数というのはデータをしまうメモリ上の場所のことです。シェルの抱えるデータ領域にある変数は特にシェル変数と呼ばれます。シェル変数の定義はいつでも好きなときにできます。bashでは変数に値(ここでは文字列だけを扱う)を与えるには代入式を使います。cshではsetコマンドを使用します。

変数名=文字列

です。なお、「=」の前後に空白があってはいけません。空白がある場合は、命令の実行と做されるからです。

```
p=/usr/local-gnu/bin
cd $p
```

のように、変数pに代入し、その値を利用するには\$Pを使います。この最後のコマンドラインは実行される前に次のよう置き換えが行われます。

```
cd /usr/local-gnu/bin
```

また、コマンド実行履歴には実際にタイプした命令が記録されます。

```
t="This is a pen."
```

等としたと同等に現在の日付および時刻を変数 `d` にセットします。設定された内容は `$d` で参照できます。また文字列に空白がある場合には `$d[2]` 等の参照により配列としても利用できます。

```
echo $t
```

とすると、`echo` は引数の文字列を出力するコマンドなので

```
This is a pen.
```

という文字列がOutputされます。データを格納する時は変数名だけを使い、変数の中身を参照する時は `[$変数名]` とします。変数が不用になったときは `unset` コマンドで変数が使用していないメモリを解放することができます。

```
unset t
```

これは変数 `t` の使用していたメモリを解放するコマンドです。  
 bashにおいても `set` コマンドを引数なしで使うと、現在使用中のシェル変数とその値の一覧がOutputされます。多分、あなたが、セットした覚えのないさまざまな変数が表示されるはずです。シェルには組み込みの変数がたくさんあり、ユーザは気づかぬうちにそれらのお世話になっています。

## 12.4.4 コマンドの出力から文字列を作り出す

`bash` の場合は

```
d='date'
```

又は

```
d=$(date)
```

ヒすると (`date` はバッククオートで囲まれている)

```
d='Tue Sep 7 10:38:34 JST 2016'
```

と同等になります。この場合 `d` は配列ではありません。また、予め `d` が配列であった場合は `d[0]` への代入となります。  
`csh` の場合は

```
set d = 'date'
```

ヒすると

```
set d = ('Tue Sep 7 10:38:34 JST 1993')
```

## 12.4.5 コマンドの出力を他のコマンドの引数として埋め込む

コマンドを実行してその出力を文字列として他のコマンドで再利用するには

```
'コマンド'
```

のようにバッククオートでコマンドを囲みます。使用例は

```
ls -al 'which emacs'
```

このようにコマンドの出力を他のコマンドの引数として埋め込むことができます。

## 第13章 シェル(その2)

```
myhost% !! | more
ls -l | more
...
```

!!は直前に実行されたコマンド文字列全体で置き換えられます。

### 13.1 シェルの対話機能 2 — bashを中心

#### 13.1.1 履歴 (history)

コマンド実行履歴の機能は殆どのシェルで利用できます。ここでは bash の便利な機能を紹介します。

HISTSIZE=10

として、シェル変数HISTSIZEを設定することで、最近実行したコマンドを10回まで記憶させることを指定することができます。

```
myhost% history
2 who
3 cd shell
4 date
5 ls -l
6 cd ./bin
7 ls -l
8 who
9 history
```

いろいろ作業をした後で history コマンドを利用して自分自身が実行したコマンドの履歴を知ることができます。(多分、みなさんの場合 HISTSIZE の値は .bashrc の中で既に設定済みでしょう。) history はそれぞれのシェルの組み込みコマンドなのでシェル毎に違います。以下の説明はcsh,tcsh,bash で共通です。

```
myhost% !5
1s -l
...
.
```

履歴リストの5番目が実行されたわけです。

```
myhost% !
1s -l
...
.
```

これで直前に実行したコマンドが再実行されます。

```
myhost% ls -l phone.numbers
myhost% chmod o-r !$
chmod o-r phone.numbers
.....
myhost% gcc test.c -lm -o test.out
...
myhost% cp !gcc:4 ~/test.bin
cp test.out ~/test.bin
....
```

この !\$ は直前のコマンドの最後の引数を参照します。

```
myhost% gcc test.c -lm -o test.out
...
myhost% cp !gcc:4 ~/test.bin
cp test.out ~/test.bin
....
```

この例で履歴コマンドの修飾子の説明をしましょ。

| 0   | 1      | 2   | 3  | 4        |
|-----|--------|-----|----|----------|
| gcc | test.c | -lm | -o | test.out |

コマンド文字列内の各単語にはシェルによって上のように番号がふられています。  
このコマンドの実行の後に例のように test.c という文字列を取り出しだければ

```
!gcc:1
```

が test.c を表すことになります。直前の gcc から始まるコマンド文字列の第1単語という意味です。

```
!gcc:4
```

は gcc で始まる最近実行したコマンドの第4単語を指します。  
以前のコマンドを修正して使うこともできます。(キャレット)を使えば直前のコマンド文字列内の置き換えることは先の例で示しました。

```
myhost% ! gcc:s/-lm/-lcurses/
gcc test.c -lcurses -o test.out
...
```

このようにすれば履歴の上で離れているコマンド文字列の編集もできます。gccで始まる最近のコマンドの文字列内の置き換えを行いました。

```
myhost% ! gcc:gs/test/test1/
gcc test1.c -lcurses -o test1.out
...
...
```

続けてこの結果を見ると置き換えが2箇所で行なわれています。このように置き換えが2カ所以上で必要なときはsの前にgをつけます。

なお、置換文字列の指定での区切り文字は/（スラッシュ）でなくてもかまいません。そうでないとパス名の置き換えが苦しくなってしまいます。

```
myhost% ls /usr/local/bin
.....
myhost% !ls:s=al=al/gnu= <---区切り文字に注目
ls /usr/local-gnu/bin
.....
myhost%
```

例えば/usr/localを/usr/local-gnuに置き換えるときはこのようにすればうまくできます。sの直後に現れた文字が区切り文字として最後まで用いられます。

以前のコマンドを編集して使うのはとても便利ですが、置換が終わった途端に実行されたのでは予想外の間違いが起る可能性があります。それが心配な人のために実行前の確認のための手段が用意されています。:pという修飾子は履歴リストからコマンドを編集して取り出したあと表示だけして実行せず、しかし履歴リストには加えてくれます。従って次のようなことができます。

```
myhost% !gcc:gs/test1/test0:/p
```

```
gcc test0.c -lcurses -o test0.out <---- 表示のみで実行せず
myhost% !
...
```

コマンド履歴を利用して実行結果を再利用することもできます。

```
myhost% which emacs
myhost% ls -al '!',
```

まだまだ履歴の利用方法はいろいろありますが、ここまで理解できた人はオンラインマニュアルを読む力もあると思いますのでそちらに当たってみることをお勧めします。

### 13.1.2 別名 (alias)

次はコマンドの別名機能です。aliasはコマンドの別名を定義するコマンドです。bashでは/usr/bin/aliasや/usr/bin/unaliasを利用します。

```
myhost% alias lsl='ls -l'
myhost% alias lsl='ls -l'
...
...
```

としておいてls1と打ち込むとls -lに置き換えられます。

```
alias cx='chmod +x'
```

とするだけでこのcxに引数をあたえて実行することもできます。Cシェルは引数が別名の定義の後に続くものとして処理してくれます。

```
myhost% unalias cx
```

aliasをキャンセルするにはunaliasを実行します。

また、別名の設定をいつも有効にしたければ「.bashrc」に書き込んでおくのが普通です。便利な別名はあなたのものとして実験してみてください。

いつも使う別名――

いつも使う別名は「.bashrc」に書き込んでおく。

cshの場合はaliasやunalias等はcshの組み込みコマンドとして提供されており、書式が多少異ります。

### 13.1.3 クオーティングとメタキャラクタ

!, \*, :, \$,などの文字はシェルのメタキャラクタと呼ばれ特別の意味が与えられています。この特別の意味をはぎ取つて單なる文字として扱いたいときにはクオーティング、あるいはエスケーピングという操作が必要になります。クオート文字自身がクオートされたりすることがあるので、クオーティングは非常に分かりにくい操作です。

| シェルのメタキャラクタの一覧 |
|----------------|
| SPACE, TAB     |
| RETURN         |
| \$             |
| * [ ] ? { } -  |
| > < &          |
| ! ^            |
| !`             |
| \ , "          |
| &              |

<sup>1</sup> 文字をエスケープするには\マークを直前におくかシングルあるいはダブルクオートで用います。ただし次の制約があり注意が必要です。

|   |                      |
|---|----------------------|
| \ | 制約なし                 |
| , | !と\はエスケープできない、       |
| " | , \$, \, 'はエスケープできない |

メタキャラクタの利用方法の例を示しましょう。

```
myhost% cp /usr/local/tex/lib/macros/a4j.sty \
/usr/lib/tex/macros/testversions/a4jtest.sty
```

行末の\は改行がコマンドの区切りと解釈されないように改行をエスケープしています。但し単語の区切りにはなってしまって注意して下さい。

```
myhost% echo The string \"\$t\" is stored in the \$t variable.
```

```
The string "Hello!" is stored in the $t variable.
```

echo引数の\によるエスケープ特に注目してください。

```
myhost% echo *** \$t ***
*** Hello!
myhost%
```

続いて、\$tはtの値で置き換えられてしまいました。

```
myhost% echo *** \$t ***
*** $t
myhost%
```

今度は置き換えは起こりません。シングルとダブルではクオーティングの強さが違うのです。

```
myhost% grep "dict[1-4]" ch*.doc
....
```

dict[1-4]はシェルによってファイル名展開をして欲しくないのでダブルクオートしました。

```
grep dict1 dict2 dict3 dict4 ch1.doc ch2.doc ch3.doc
```

のように展開されてしまうと第2引数以後の位置が押されてしまい、期待した結果は得られません。

## 13.2 Bashシェルプログラミング

bashのプログラミングを紹介します。

### 13.2.1 for

作業ディレクトリにあるファイル名の末尾がtexのファイルの一覧が表示するために、forを使います。

```
for f in $(ls *tex); do
 echo $f
done
```

ls \*tex を実行して得られる文字列のリストを1つずつシェル変数fに代入してdoとdoneの間の処理をします。

### 13.2.2 文字列切り出し

```
for f in $(ls *tex); do
 b=${f%.tex}
 echo $f $b.backup
 cp $f $b.backup
done
```

シェル変数bにはファイル名の後の.texの4文字を削除した文字列が代入されます。この様に文字列の一部を切り出す方法があります。以下の様にいろいろあります。

|               |            |                           |
|---------------|------------|---------------------------|
| 元             | \$t        | /usr/local/bin/sample.tex |
| 頭の最小パターンを消す。  | \${#/*.}   | tex                       |
| 頭の最長パターンを消す。  | \${{##/*}} | sample.tex                |
| お尻の最小パターンを消す。 | \${{#/*}}  | /usr/local/bin            |
| お尻の最長パターンを消す。 | \${{#/*}}  | /usr/local                |
| 最初のパターンを置換    | \${{#/*}}  | /usr/local/bin/sample.tex |
| 全てのパターンを置換    | \${{#/*}}  | -usr-local-bin-sample.tex |
| 位置と長さで切り出し    | \${{2:4}}  |                           |
|               | \${{}}     | sed                       |

sedを使えば、シェルの種類に依らず、正規表現を利用した置換を利用して、文字列の切り出しや置換ができます。

```
newmoji='echo_$moji_|sed_\'s/req_pattern/mojiretu/''
又は
newmoji=$(echo_$moji_|sed_\'s/reg_pattern/mojiretu/')
```

### 13.2.3 ファイルテスト演算子

ファイルテスト演算子を利用すると、ファイルの属性に応じて処理をすることができます。

```
for f in $(*ls); do
 if [-d$f]; then
 echo $f
 fi
done
```

-d はディレクトリかどうかをテスト (test) するオプションです。オプションは他に

|                  |    |
|------------------|----|
| ディレクトリ           | -d |
| ディレクトリでない普通のファイル | -f |
| ファイルが存在するか       | -e |
| ファイルが空でない        | -S |
| 読みだし可能なファイル      | -R |
| 書き込み可能なファイル      | -W |
| 実行可能なファイル        | -X |

などがあります。

### 13.2.4 配列

bash では配列変数を利用できます。bash の場合は shift は引数変数のみが対象であり、配列変数には使用できないので、以下の様に for を使用すると配列の全ての要素を扱うことができます。

bash では function を使って、関数を定義でき利用することができます。function を利用すれば shift を使うことができます。

```
% abc=(x y z)
% echo ${abc[2]}
z

function f() {
 While["$1"!=""]; do
 echo $1
 shift
 done
}
f ${abc[@]}
f ${abc[@]}

for a in ${abc[@]}; do
 echo $a
done
x
y
z
```

コラム 2

Windows の環境変数は "システムプロパティ" の "詳細設定" の "環境変数" に設定があるのを知っている人はどのくらいいるのかな。コマンドプロンプトを開くと端末が開いてコマンドを入力できる。path とタイプしてリターンを押すと、コマンドサーキュラスが表示されます。set とタイプしてリターンを押すと、設定されている全ての環境変数が表示されます。

## 第14章 簡単なシェルプログラミング

```
chmod u+x nu
```

を実行して nu を実行可能に設定して、(14.1) の様に source を利用する代わりに

```
./nu
```

としてやるだけで nu はふつうのコマンドのように実行されます。実際には sh nu が実行されているのですが。それは意識しなくて済みます。

ところでシェルにはいろいろな種類がありますが、シェルに対するコマンドを書き並べたプログラムをどのシェルが実行するのかを、どの様な仕組みで決めるのでしょうか。実行すべきシェルを決めるのはカーネルの仕事です。どのようにかといふと、プログラムファイアルの先頭を見るのです。この先頭行が#以外の文字で始まっていたらB シェルが起動されます。先頭が#だったらC シェルが起動されるのが原則です。また、次のように

```
#! 起動するシェルの絶対パス
```

を先頭行に書いておくと指定したシェルが起動されます。さて、このままでは nu はカレントディレクトリに nu があるときしか実行できませんが、例えば nu を

```
~/.bin (あなたのホームの下の bin というディレクトリ)
```

という名前のディレクトリに移し、コマンドサーキュラーパスに ~/bin を加えるとどの作業ディレクトリに居ても nu を実行できるようになります。

シェルはユーザからコマンドを受けとるとサーキュラーパスにリストされているディレクトリの中からコマンドの本体にあたるプログラムを探ってきてそれをメモリにロードして実行するようにカーネルに指示(exec システムコール)を出します。ということはサーキュラーパスから外れているディレクトリに書かれているプログラムはシェルにそのファイル名を与えてやってもコマンドとして実行されないになります。もっとも絶対パスを指定すれば可能ですが。 いますぐ、サーチパスに ~/bin を加えたいなら次のようにします。

### 1. ログインシェルがC シェルの場合

```
set path=($path ~/bin)
rehash
```

### 2. ログインシェルがB シェルの場合

```
PATH=$PATH:$HOME/bin
```

これを、コマンドラインから打ち込めば、あなたのホストに現在ログインしているユーザの人数が画面に出力されます。シェルにファイルからコマンドを読ませることを試してみます。次のようにすればそれが可能です。

```
source nu
```

またはファイル nu のアクセスペーミションを

```
(14.1)
```

という行がいくつかあり、その中に PATH という項目があるはずです。これを確かめて

PATH=...

という行のリストに

HOME/bin

を加えます。リストの区切り文字はコロン「:」です。ここでもう一度注意して欲しいのはログインシェルとシェルプログラムを実行するサブシェルは関係ないということです。

ls -l

によりファイルのモードを知ることができます。表示されるモードの読み

方は次の通りです。-rwxr-xr--とあつたら先頭の - は通常のファイルであることを示します。(ディレクトリなら d, リンクなら l となります。) 残りの部分は 3 つずつ組にして考

えます。最初の組 rwx はこのファイルの所有者に Read, Write, eXecute すべてが許されていることを示しています。2 番目の組 r-x はこのファイルの所有者と同じユーザグループに属する人に read, execute を許可するが、書き込み (変更) は許されないことを示します。3 番目の組 r-- はその他のユーザーに read 許可のみ与えることを示します。chmod でこれらのモードを簡単に変えることができます。

chmod [ugoa][+-=][rwx] filenames

[ ] 内はどうかが選ばれるものと解釈して下さい。

|   |                   |
|---|-------------------|
| u | user ファイルの所有者     |
| g | group 所有者の属するグループ |
| o | others その他        |
| a | all 全ての人          |

のいずれかの人に対して (単に並べれば同時に指定することもできる)

- + 許可を与える
- 許可を取り消す
- = モードを示されたように初期化する

r, w, x も並べて同時に指定することもできます。例えば

myhost% chmod ug+rwx newcommand

のように使うことができます。

#### 14.1.4 コマンドの引数

ファイルのモードを実行可能に変えるためのコマンド cx を作ってみましょう。

myhost% cx newcommand

が myhost% chmod u+r newcommand の簡略表現になるようにしたいとき cx というファイルに次のように書いておきます。

chmod u+r \$1

\$1 は cx の実行に先だってシェルによって一番目の引数に置き換えられます。2つ以上の引数には \$2, \$3, ..., \$9 が使えます。(csh, tcsh の場合は \$argv[1] も使用できます。) さて、この cx 自身を実行可能にしなくては意味が無いので

myhost% chmod u+r cx

としておきます。cx に一度に複数のプログラムを処理させるためには少し改良しなければなりません。cx を

chmod u+r \$1 \$2 \$3 \$4 \$5 \$6 \$7 \$8 \$9

と書き直すのも一つの手ですが、全ての引数を意味する \$\* という表現が使えるので、

chmod u+r \$\*

とするのがベストです。lc (line count) というコマンドをファイルの行数を数えるために作ってみましょう。

wc -l \$\*

と書いてやればうまく行きます。

wc

wc に -l というオプションを付けると行数だけの報告をしてくれます。

#### 14.1.5 プログラムの出力を引数として受け取る

プログラムを実行してその出力を他のコマンドの引数にする方法があります。

myhost% ls -l 'which csh'

-rwxr-xr-x 1 root 147456 Jul 24 1992 /bin/csh\*

とすると which csh の出力 /bin/csh が ls コマンドの引数として渡されます。

## 14.2 C シェル(csh, tcsh)

### 14.2.1 foreach

csh, tcsh では `foreach` コマンドを使うと、bash の `for` の様な処理ができます。特定の規則に合致したファイルについて、同じ操作をするのに大変便利な方法です。例えば、`tex` で終るファイル名を持つファイルの一覧を作るには、

```
foreach n (*.tex)
```

と入力すると、

```
?
```

と問い合わせてきます。ただし、該当するファイルが一つもない、

```
foreach: No match.
```

と該当する(`match`)ものが無いという返事だけが帰ってきます。

```
foreach n (*.tex)
? echo $n
? end
```

とすると、一覧が表示されます。この例では、()の中に該当する名前を持ったファイル名が一つずつ順番に代入されて、`end`までに打ち込まれたコマンドを実行します。コマンドの行では、代入された `n` の内容を `$n` で参照することができます。

```
foreach f (*.tex)
? echo $f $f.backup
? cp $f $f.backup
? end
```

この例では、`tex` で終るファイルを `f` に代入します。そして、そのファイル名の末尾に `.backup` をつけたファイルにコピーしています。この例では、例えば `a.tex` は `a.tex.backup` になりましたが、`a.tex` を `a.backup` にしたい場合には次の様にします。

### 14.2.2 文字列修飾子

```
foreach f (*.tex)
? echo $f $f:r.backup
? cp $f $f:r.backup
? end
```

この様にファイル名の一部を取り出す方法は

|      |       |                           |
|------|-------|---------------------------|
| 元    | \$n   | /usr/local/bin/sample.tex |
| 基本部分 | \$n:r | /usr/local/bin/sample     |
| 先頭部分 | \$n:h | /usr/local/bin            |
| 末尾部分 | \$n:t | sample.tex                |
| 拡張部分 | \$n:e | tex                       |

となっています。csh では文字列修飾は限られた機能だけが提供されています。複雑な処理が必要であれば `sed` 等の外部のプログラムを利用します。

### 14.2.3 ファイルテスト演算子

```
foreach f (d*)
? echo $f
? end
```

この例は、`d` から始まる名前のファイルの一覧です。この中から `d` から始まる名前のディレクトリだけを表示したい時には次の様に実行します。

```
foreach f (d*)
? if (-d $f) then
? echo $f
? endif
? end
```

この例ではこの部分が重要です。`-d` はディレクトリかどうかをテスト(test)するオプションです。オプションは他に

|               |    |                  |
|---------------|----|------------------|
| ディレクトリ        | -d | ディレクトリでない普通のファイル |
| 自分が所有者であるファイル | -f | 読みだし可能なファイル      |
| 書き込み可能なファイル   | -o | -r               |
| 実行可能なファイル     | -w | -x               |

などがあります。

### 14.2.4 配列

csh, tcsh の配列の扱いは bash と比較してシンプルです。

```
foreach f (*.tex)
? echo $f $f:r.backup
? cp $f $f:r.backup
? end
```

```
% set abc=(x y z) <-- インデックスは 1 から
% echo ${abc[2]} <-- csh では配列全体
y
%
% echo ${abc} <-- csh では配列全体
x y z
%
% echo ${#abc}
3
%
% while($#abc)
? echo ${abc[1]} <--- csh では配列変数をシフトできる。
? shift abc
? end
x
y
z
```

csh,tsh では配列変数のシフトの機能があるので、bash とは異なり、配列を使った目に分かり易いプログラムを書くことができます。

## 第15章 期末試験

### 15.1 期末試験

- ・出題範囲はリテラシーIすべてです。
- ・実施教室は電子メールやニュースで案内されます。
- ・持ち込み不可。
- ・答案提出後の退室可。

#### 15.1.1 成績評価

課題や宿題と試験の総合評価です。採点の状況は  
~a-fuji.tu/education/literacy/2017/summer/1.html  
で参照できます。この表のスコアが高くても、定期試験に不参加の場合や講義や演習の出席率が低い場合は最終成績は0点Fです。

#### 15.1.1.1 再評価

提出物の再評価は通常の成績の場合は実施しません。6月末時点の成績評価で判断して、再評価の対象となる学生に対しメールで連絡します。再評価の場合は合格点を越えて、最終成績は50点C評価となります。

#### 15.1.1.2 追試

追試が実施された場合も再評価と同様に合格点を越えて、最終成績は50点C評価となります。

## 第1章 諸注意, キーボード

- 各自のホームディレクトリ以下に保存するファイルで採点する場合があります。
- 指定されたディレクトリに指定されたファイル名で保存されている場合にのみ採点します。

### 1.1 はじめに

#### 1.1.1 諸注意

本体の電源は切らない UNIX Workstation は、パソコンや他の家電製品と違つて、大変複雑な処理を行なっています。そのため、例えば操作を間違えたりシステムにトラブルがあつたりして、画面がおかしくなったり、キー入力を受け付けないからといって、急に電源を切つてしまふと、故障して次からは動かなくなってしまうことがあります（というより、大抵の場合トラブルの原因になります）。ですから、

教官や、情報処理センター員の許可がない限り Workstation 本体の電源は決して切つてはいけません。（ディスプレーは比較的電力消費量の大きい機器です。Workstation を使用しない場合には、ログアウトを確認した後でディスプレーを Power Off します。）

### 1.1.2 演習の進め方

演習の時間は、基本的に毎回与えられた課題を、各自が自由にこなしていくものとします。まず、演習の時間の始めに教官が、その日の課題などについて、5~10 分程度説明します。そのあとで、各自のベースで自由に演習を進めてください。授業時間終了前にその日の課題をすべて終了した人は、その時点で退出して構いません。

解らぬことや、トラブルがあった時などは、遠慮せずに速やかに教官や TA、もしくは情報処理センター員に申し出て下さい。また、近くの席の友達と相談したり、解らないところを教えあつたりしても、勿論かまいません。ただし、当然のことながら大声で話したり、他の人の演習の妨げになるようなことはしてはいけません。

### 1.1.3 採点のポリシー

以下に示す事柄は各自が良く確認して間違いの無いようにしましょう。

- “課題”は演習の時間内に担当教官に提出します。

- “宿題”的提出期限は、それぞれについて指定されています。

- リテラシーのニュースグループ（= u-aizu.course.list1）に投稿されるアナウンスを毎日チェックして指示を確認する習慣を身につける必要です。

<sup>1</sup>Windows パソコンや Macintosh でも定められた手順で電源を切る必要があります。

- 他のユーザや会津大学のセキュリティに対して明らかに故意に迷惑を及ぼした場合には単位を与えない措置もありうる。
- 教員に対して読み出しのアクセス許可がある場合にのみ採点します。
- （情報センタ利用規定細則により）学部学生の適切なディスクの使用量は 1000 MByte 以下とされています。十分注意することを要望する。（ファイルサーバの更新などにより変更されるので情報センターアナウンスを参照しましょう。）

### 1.2 アカウントと login

#### 1.2.1 アカウントとはなにか

UNIX Workstation では、一般的の利用者と、システムを管理する人が厳格に分けられています。一般的の利用者のことをユーザー、システム管理者のことをルート、スーパーユーザーなどと呼ぶことがあります。利用者は、まずシステム管理者に Workstation の使用許可をもらう必要があります。このことを、アカウント<sup>2</sup>を貰うなどといつたりします。ここで始めてユーザーになるわけです。ユーザーには、一人につづつログイン名が与えられます。ログイン名は、ローマ字またはローマ字と数字の組合せからなります。会津大学では、学生のログイン名は学生番号をもとに作られます。

### 1.2.2 login のやりかた

端末を使う時には、まず自分が端末を使い始めるということを、端末に宣言しなければなりません。これを“ログイン (login) する”といいます。

一斉に login するとトラブルがここで一つ注意があります。この演習室の端末は、全てがネットワークで相互接続されています。たくさんの中の端末から一斉にログインすると、このネットワークの通信量が一時的に極端に大きくなり、最悪の場合には全ての端末が使用不能になる可能性があります。そこで、今回の演習では、教官の指示にしたがって、小人数ごとにログインを実行して下さい。次回からは、演習室に到着したら授業が始まる前になるべくログインを済ませて下さい。ただし、前の授業が長引くなどして大勢一気に入室した時は、各自で適当にログインのタイミングを調節して、トラブルを避けるようにして下さい。

#### 1.2.2.1 グラフィカルなログインスクリーンが表示されている場合

図(1.1)の様なグラフィカルなログインスクリーンが表示されている場合はユーザアカウントを入力しリターンキーを押します。その後にパスワードの入力を求められるので入力します。Mac OS X の端末でも同様にログインします。

<sup>2</sup>account: 預金口座という意味があります

パスワードの変更には1時間位かかる 集中管理されているネットワークでは全ての端末で、共通のログイン名とパスワードを利用できる必要があります。スタンダードアローンの計算機の場合はパスワードの変更はすぐ反映されますが、この演習室の場合は、パスワードの変更手続きをしてから実際にパスワードが変更されるまで、1時間くらいかかります（場合によってはもっとかかる）。そこで、今回はパスワードの変更を行なわず、皆さんがもっと慣れてくる次回以降に回します。



図 1.1. login screen

### 1.2.3 パスワードについて

パスワードとはなにか ログインの説明のところで、パスワードという言葉が出てきました。パスワードとは、そのログイン名を使っている人が、ログイン名の持ち主本人であるかどうかを確認するためのもので、銀行のキャッシュカードの暗証番号と同じです。UNIX Workstation では、パスワードはローマ字、またはローマ字と数字の組合せで、6 文字以上 8 文字以下でなければなりません。

このパスワードは、絶対に人に教えてはいけません。本人が教える以外は、たとえシステム管理者でもこのパスワードを知ることはできません。（パスワードはコンピュータ上に暗号化されています。この暗号をもとに復号してパスワードを知ろうとしても、意味のある時間、以内には不可能な仕組みになっています。）ログインの時に、パスワードは入力しても画面に表示されないのも、パスワードを人に知られないためです。ですから、友達にパスワードを教えたり、手帳やノートなどに書いてはいけません。

パスワードの変更について ユーザは自分のパスワードを適切に管理する責任と義務があります。最初に配布された現在のパスワードは、システム管理者がユーザーのために仮に登録したもので、つま、第3者であるシステム管理者がパスワードを知っているわけです。そこで、パスワードを自分しか知らないものに変更しなければなりません。しかし、以下の理由で今回はパスワードの変更は行いません。

ネットワークによるユーザー管理について 演習室の Workstation は、全てネットワークで接続されています。さらに、演習室のどの Workstation を使っても同じ作業環境（=ホームディレクトリとコマンド）が提供される仕組みにてあります。例えば、前回は一番前の Workstation で演習したけれども、今回は前から 3 番目だという場合でも、前回の続きから、あたかも同じ Workstation を使うがごとくに演習を行なうことができます。この仕組みを実現しているのが LDAP です。（Lightweight Directory Access Protocol）ホームディレクトリ等を物理的に保持しているファイルサーバをネットワークを経由して透過程的に利用する仕組みは NFS（Network File System）で実現しています。

### 1.2.4 logout のしかた

Workstation の使用開始を終了するためには使用終了のための操作を実行する必要があります。これをログアウトといいます。ログアウトが完了しない今まで端末を放置すると、あなたのアカウントの権限で悪戯をされるかも知れません。ログアウトが完了するとログインを待機する画面に戻ります。

うっかりしたミスでログインしたまま端末を放置した場合や、ログインしたままずっとトイレに行く場合などのように、ログインしたまま席を離れることは、他のユーザがあなたの利用権限でコンピュータを利用する可能性が高くなります。他人による不正な利用を避け、コンピュータからログアウトした後でもプログラムを動作させ続けて計算処理が終了するまで実行させることや、サービスを提供し続けることができます。この様に、本来プログラムの動作とログイン状態とは関係ないので、ログアウトをすればその時点でき自分が実行していたプログラムがすべて完全に終了するとは限りません。ユーザの意思に反して残って動作しているプログラムを「残留プロセス」と呼びます。残留プロセスが動作し続けている可能性があるので、ログアウトする前にプロセスの終了の確認をします。

動作しているプロセスの確認

```
ps auxww|grep username または ps -elf|grep username
```

- GUI ベースのログイン

- Gnome Desktop System の場合  
画面の上部にある System Menu を開きログアウトを選択します。

- ログインウインドウがある場合  
ログインウインドウ内で exit<sup>3</sup> を実行します。

- ログインウインドウが無い場合  
Window System が提供する操作を利用してログアウトします。

<sup>3</sup>See 'man -s 1 exit'

- Mac OS X 端末の場合

iLabや演習室5,6等の教室にはMac OS X端末が導入されています。この場合のログアウトはアップルメニューを開いてログアウトを選択します。

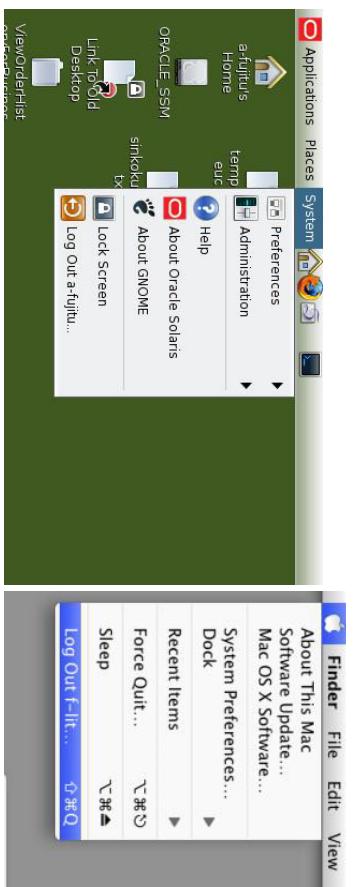


図 1.2. gnome desktop / mac os x logout

利用開始前に表示されていた90ページの図(1.1)の様なログインスクリーンが表示されたら、ログアウト完了です。

### 1.3 キーボードの操作

文字の入力を自由に行えるようになるために、キーボード操作の基本を解説します。最初に使用するのは、図1.3に示した、キーボード全体の真ん中の当たりに位置するキーのみです。当面は、他のキーは使わないものと思ってください。

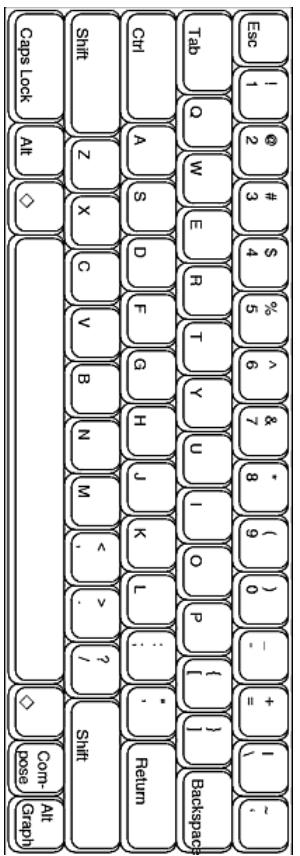


図 1.3. 主に使用するキー

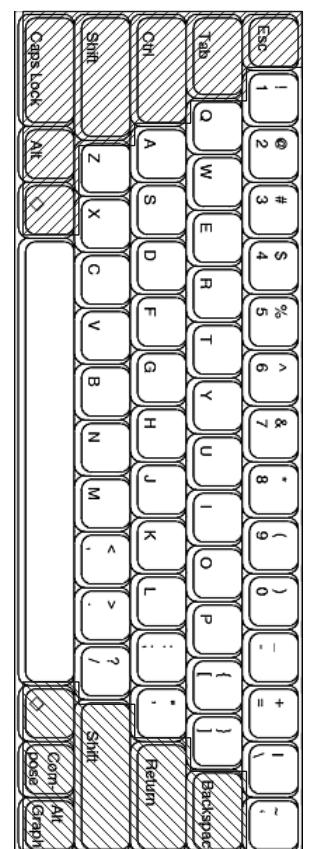


図 1.4. 制御キー

**Control** (コントロールキー) このキーを押しながら、アルファベットキーを押すことで、特殊な制御文字を入力する。ctrl-X, ^X, C-xは、何れもこのコントロールキーを押しながら、Xのキーを押すことを意味する。

**Shift** (シフトキー) キーの中には、1のキーのように、!と二つの文字が書いてあるものがいる。これらのキーをそのまま押すと、1のように下に書いてある文字が、また、シフトキーを押しながら押すと、!のように上に書いてある文字が、入力される。

また、後述のCaps Lockキーが有効になつてない状態で、アルファベットのキーを押すと、小文字のアルファベットが入力されるが、シフトキーと押しながら押すと、大文字が入力される。Caps Lockキーが有効になつている状態では、反対にそのまま大文字、シフトキーを押すと小文字が入力される。

**Back Space** (バックスペースキー) カーソルの直前にある一文字を削除する。入力間違いをした場合は、このキーで間違った文字までを消し、正しい文字を入力する。

**Return** (リターンキー) コマンド(命令)を一つ入力するたびに、入力が完了したことを示すために押す。文章の入力では、入力の区切りに押したりする。

また、左下にあるCaps Lockと書いたキーを押すとキーボードのどこかにある小さなLEDが点灯したりします。LEDが消えた状態でCaps Lockを押すと、LEDが点灯し、Caps Lockが有効であることを示します。もう一回押すと、再びLEDが消え、Caps Lockが有効で無いことを示します。Caps Lockの有効無効の意味は、Shiftキーのところで説明しました。

図 1.4 で縦掛けになっているキーは制御キーと呼び、入力の補助に使います。この中で頻繁に使うのが **Control**, **Shift**, **Backspace**, **Return** の各キーです。コントロールキーや一部の記号のキーのレインボウはキーがボードの機種によって異なります。Mac OS X端末用のキーには **Command** キーがあり、これも頻繁に使用します。

### 1.3.1 制御キー

### 1.3.2 指の割り当て

速くて正確なタイピングのためには、正しい指使いが有効です。正規のタイピングでは、各キーを押す指が決まっています。図1.5に各キーへの指の割り当てを示します。

**ホームポジション** 左手の小指から順に ASDF の位置に、右手の人差し指から順に JKL; の位置に指を置いた状態を、ホームポジションに指を置いた状態と呼びます。ホームポジションは、正しいタイピングの基本です。しっかりとマスターしてください。正しい姿勢でのタイミングは肩こりや腰痛、視力の低下を防ぎます。

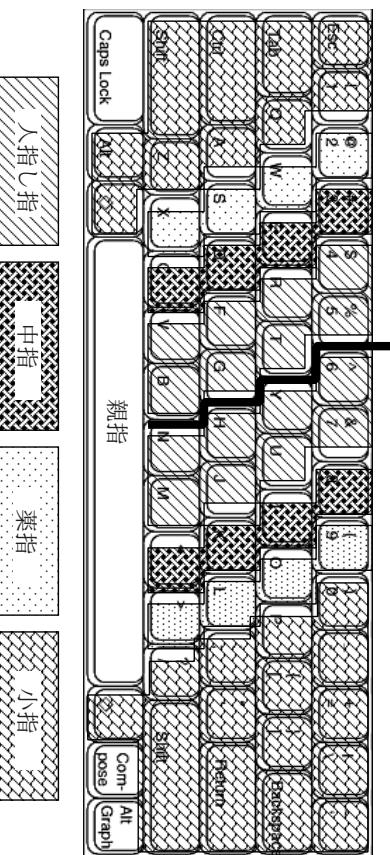


図1.5. 指の割り当て

### 1.4.2 タイピング練習をしてみましょう

タピング練習プログラム typist<sup>5</sup> を試してみましょう。

1. ターミナルの中で typist と入力し、リターンキーを押します。
2. 説明にしたがって練習してみて下さい。
3. 途中で止めた場合には C-c を押して下さい。

### 1.5 端末エミュレータ

コンピュータの利用には GUI を用いたものと CUI を用いたものがあります。CUI の代表的なツールは端末エミュレータです。端末エミュレータは X Window System で動作するものは xterm, gnome-terminal 等があります。Mac OS X 端末の Aqua インターフェースでは Applications フォルダ内の中の Utilities フォルダ内の Terminal が利用できます。端末エミュレタ内でコマンドをタイプし、実行することで様々な操作や情報処理を行うことができます。

1. アプリケーションを起動すること。…例) emacs, firefox
2. ファイルの管理を行うこと。…例) cp, mkdir
3. 情報処理を行うこと。…例) grep, sort, awk

従つて、会津大学の新規ユーザのための配布用標準環境は以下の様にターミナルを利用しそく工夫されています。

- Gnome Desktop System では Gnome Panel (≡ gnome-terminal )がある。
- Mac OS X では ドックに Terminal がある。

### 1.6 フォントサイズ

gnome-terminal の場合は  $\text{Ctrl} + \square$  で拡大、 $\text{Ctrl} + \square$  で縮小します。  
Mac OS X の Terminal では  $\text{Command} + \square$  で拡大、 $\text{Command} + \square$  で縮小します。

#### 1.3.3 練習

演習を円滑に進めるために、キーボードからの入力に慣れる必要があります。

**cat<sup>4</sup>** と入力して (リターンキーを押すのを忘れずに)、アルファベットの大文字小文字や、数字、記号を入力しましょう。終了は、まずリターンキーを押し、そのあと C-d を入力します。

#### 1.4 タイピングの練習

**1.4.1 タイピングの基礎知識**  
キーボードを使ってタイブする際に、タイピングの基本を知っておくと良いです。基本を十分に理解した上でトレーニングを積んでいきましょう。

<sup>4</sup>See 'man cat': concatenate and display files

<sup>5</sup>See 'man typist': typing tutor

```
% This is an example of LaTeX documents.
% This is a comment-line.
\documentclass[a4paper,12pt]{article}
\begin{document}
Name: XXXXXX\\
Birthday: XXXXX\\
Blood type: XX\\
Phone: 0242-XX-XXXX(悪戯電話を避けるためにXのままでよろしいです。)\\
Hobby: XXXXXX\\
Comment: XXXXXX\\
\end{document}
% To compile this, do 'pdflatex introduction.tex'.
% To convert this, do 'dvips introduction.dvi -o introduction.ps'.
% To preview this, do 'gv introduction.ps &'.
```

## 1.7 宿題1

以下のテキストをEmacsでタイプして、ファイル名をピリオドから始まる.planとしてホームディレクトリに保存しなさい。

Hello, I am XXXX.

`finger -m username` を実行するとユーザのホームディレクトリに.plan ファイルがあわせその内容等を表示します。`finger`のサービスが機能しているコンピュータシステムでは..plan の内容をインターネットを利用して参照されます。自宅の電話番号などを書くと悪戯に利用されることもあるので、書かないでおくことが賢明です。<sup>6</sup>

.plan ファイルはホームディレクトリで`ls`コマンドを利用して確認します。

`ls -l ..plan`

オプション1を使用すると情報を多く含んだロングフォーマットで表示します。

## 1.8 注意1

コンピュータを利用する場合にはいろいろな命令やアプリケーションソフトウェアの説明を読む必要が出てきます。UNIXシステムには例えれば日頃良く使用する命令の説明のために`man`という命令があります。その説明文は英語でかかれていることが当たり前なので英語が得意でスラスラと意味が解る人以外は英和辞典を引きながら説明文を読むことになるでしょう。`man`に限らずに、外国で開発されるものは英語で説明されるのが普通のことです。ですから、英語の辞書をいつでも持つてることを勧めます。

環境変数(`LANG`)の内容を`ja_JP.UTF-8`にセットすると日本語のオンラインマニュアルを表示する場合もあります。

<sup>6</sup>See 'man finger': display information about local and remote users

```
env LANG=ja_JP.UTF-8 man man
env LANG=ja_JP.UTF-8 man set
env LANG=ja_JP.UTF-8 man file
env LANG=ja_JP.UTF-8 man sftp
env LANG=ja_JP.UTF-8 man ping
```

Webの利用が一般化してきたために、manではなく、各アプリケーションのホームページ上でドキュメントを公開することが増えてきました。manで表示した結果に「詳しくはWebで」と書いてあり、詳細説明はWebを見るように指示するものがあります。

## 1.9 Unixコマンド(1) ファイル/ディレクトリ

ファイルを適切に管理するためには、以下の操作に慣れる必要があります。他の授業でも要求される操作です。`cd`と`pwd`と`ls`を使っていろいろなディレクトリを散歩して、充分に練習してください。

- `cd` ディレクトリ名... チェンジディレクトリ (Change Directory)<sup>8</sup>: ディレクトリを移動します。引数なしでユーザのホームディレクトリに戻ります。
  - `pwd`<sup>9</sup> ... プリントワーキングディレクトリ (Print Working Directory): 現在のワーキングディレクトリを表示します。
  - `ls`... 現在のワーキングディレクトリ内にあるファイルを表示します。`ls`にオプション(スイッチ)を付けて実行すると異なる表示を得られます。例えば`ls -F`を実行したときにディレクトリであればファイル名の後尾に/を付けて表示します。これはディレクトリであることを示すフラグ(旗)というものです。`ls -1F` を使い、ファイルなのかディレクトリなのかを区別して表示できます。
- ディレクトリを作成/削除するコマンドを練習します。
- `mkdir` ディレクトリ名... メイクディレクトリ (Make Directory)<sup>10</sup>: ディレクトリを新規に作成します。
  - `rmdir` ディレクトリ名... リムーヴディレクトリ (Remove Directory)<sup>11</sup>: ディレクトリを削除します。

### 1.9.1 \$HOME/literacyを作る

リテラシーの課題提出用のディレクトリを作ります。ホームディレクトリで

`mkdir literacy`

を実行します。

<sup>8</sup>See 'man cd': cd, chdir, pushd, popd, dirs - change working directory

<sup>9</sup>See 'man pwd': return working directory name

<sup>10</sup>See 'man mkdir': mkdir - make directories

<sup>11</sup>See 'man rmdir': remove directory entries

## 第2章 パスワード管理, X Window System, Emacs (1)



Target : The University of Aizu

### 2.1 パスワードの変更

入学時に与えられた初期パスワードのままで、まだパスワードを自分で変更していない人は以下の説明をよく読んで、今回の演習から1週間以内にパスワードを自分で考えたものに変更してください。1週間後以降にパスワードを変更していない場合には情報センターから警告があり、それに従わない場合にはアカウントの利用停止の可能性があります。会津大学では定期的にパスワードを更新する規則を2004年度から採用しています。パスワード更新後およそ90日程度を経過すると、パスワード更新の警告メールが送信されます。これに応じない場合は、

正しく利用されていない危険な状態であると判定され、

アカウントの利用停止(=アカウントロック)になります。

#### 2.1.1 パスワードの変更法

パスワードの変更をするには terminal 内で passwd というコマンドを用いるか、Web ブラウザ内で利用できる Unicorn ID Manager サービスを使います。パスワードについての詳細は情報センターの案内 <http://web-int.u-aizu.ac.jp/labs/istc/ipc/topic/passwd/passwd.html> を参考にしてください。

ターミナルで passwd と入力すると、変更前の現在のパスワードと、変更後の新しいパスワードを尋ねきますので、メッセージにしたがって入力します。このとき、入力されたパスワードは、新旧ともに画面に表示されませんので、慎重に入力して下さい。

##### 2.1.1.1 passwd の実行

<CR>は、Return キーを押すことを意味します。

1. passwd: Changing password for xxxxxxxx
  2. Enter existing login password: current password<CR>
  3. New Password: new password<CR>
  4. Re-enter new Password: new password<CR> :確認のため、もう一度新しいパスワードを入力。
  5. passwd: password successfully changed for xxxxxxxx
- 入力されたパスワードは、新旧ともに画面に表示されませんので、慎重に入力して下さい。

#### 2.1.1.2 パスワードに関する一般的事項

1. パスワードは一般に6文字以上の文字列で、いくら長くても構わない。ただし、システムによっては8文字目よりあとは無視される。
  2. 名前など、本人に関する情報から容易に類推できる文字列は不適当。できれば単語としての意味がないものの方がよい。
  3. 英語、日本語に限らず、辞書に出ているものは使わない。
  4. 大文字と小文字を混ぜると不正loginを防ぎやすい。
  5. 数字を混ぜるのもよい。
  6. 時々パスワードを変えるとなおよい。
  7. パスワードは自分で覚えておく、メモなどに書いておかない。
- くれぐれもパスワードを忘れないように

#### 2.1.1.3 演習室以外や学外でのパスワード変更

ネットワーク経由で大学内の演習室環境にリモートログインすることで、地球上のどこにいてもパスワードを変更できます。

## 2.2 マウスの操作

X ウィンドウシステムについて解説する前に、その操作の基本となるマウスの使い方について説明します。マウスとは、キーボードの脇においてある、3つのボタンとケーブルのついた装置です。マウスを動かすことによって、ワークステーションの画面上の位置を指定し、ボタンを押すことによって、割り当てられた作用を実行します。

古くは機械式マウスが利用されました。現在は光学式マウスが主流です。これはレーザ光の反射を利用して動きを読み取るので、マウスの下の表面の状態によっては動作が不安定になる場合があります。マウスの基本操作を以下にあげます。

|         |                                         |
|---------|-----------------------------------------|
| ムーブ     | マウスを動かす。                                |
| ボタンプレス  | マウスのボタンを押すこと。                           |
| ボタンリース  | マウスのボタンを離すこと。                           |
| ドラッグ    | ある場所で、マウスのボタンを押して、押したまま他の場所へ、マウスを動かすこと。 |
| クリック    | ボタンプレスとリースを短時間に続けて行なうこと。                |
| ダブルクリック | 短時間にクリックを二回行なうこと。                       |

なお、マウスにはボタンが3つありますが、今回の演習では、一番左のボタンのみを使います。マウスのホールはWebブラウザ等で画面スクロールのために使用します。

## 2.3 X ウィンドウシステム

UNIX ワークステーションの特徴的な機能の一つに、X ウィンドウシステムがあります。X ウィンドウシステムとは、画面上に複数の窓(ウィンドウ)を表示して、あたかもウィンドウの一つ一つが独立したコンピュータのように動作し、複数の仕事を、同時に並行してこなすことができるシステムです。

### 2.3.1 X ウィンドウシステムの利用

#### 2.3.1.1 Solaris 端末

Solaris 端末で90ページの図(1.1)の様なグラフィカルなログインスクリーンが表示されている場合は X ウィンドウシステムは常に動作しています。

#### 2.3.1.2 Mac OS X 端末

Mac OS X の場合は Aqua インターフェースが動作しています。X Window System 対応アプリケーション(sylpheed 等)を起動すると、同時に X Window System が起動します。

## 2.3.2 ろいろなウィンドウ、マウスの操作

X window system の利用時にはいくつかのウィンドウが表示されます。その中には、以下の特別のウィンドウがあります。

- ルートウィンドウ xterm オプションで起動した場合の背景に当たる部分です。

デスクトップ Gnome Desktop System ではルートウィンドウは表示されません。代りに、デスクトップウィンドウが背景として表示されます。

ログインウィンドウ xterm オプションで起動した場合の xterm です。これを終了するとログアウトできます。Gnome Desktop System には現れません。

アクティブウィンドウ 作業の対象になっている状態のウィンドウのことです。キーボードの操作は、このウィンドウに対しても効果になります。あるウィンドウをアクティブにするには、マウスカーソルをそのウィンドウの上に移動して左ボタンをクリックします。

### 2.3.3 練習

1. 表示されているいろいろなウィンドウを、アクティブにしてみなさい。

2. Terminal と表示されているウィンドウ(gnome-terminal)をアクティブにしなさい。

3. 次の命令を実行してみなさい。新しい窓を開く命令には & を付けます。

- (a) emacs &(テキストエディタ Emacs)
- (b) xeyes &(マウスカーソルを見つめる可愛いツール)

4. 次の命令を実行してみなさい。ターミナル内に直接表示するツールは基本的に & を付けずに実行します。

- (a) cal (カレンダーの表示)
- (b) who (ワークステーションにログインしている人の一覧)
- (c) whoami (あなたのログイン名)
- (d) date (現在の日付と時刻)
- (e) jobs (ターミナルから起動したジョブの表示)
- (f) x11clients 動作している X クライアントのリスト
- (g) ps (動作しているプロセス(プログラム)のリストを表示)
- (h) rlogin ホスト名(他の計算機へのログイン)

動作しているプログラムを終了するには、emacs や firefox の様にアプリケーションに固有の正しい終了の方法があれば必ずその方法で終了します。xeyes などのようにそれが上手に行かない場合には強制終了します。強制終了には以下のやり方があります。

1. ウィンドウの枠に提供されている Close で終了する。
2. kill プロセス番号

3. kill %ジョブ番号  
プロセス番号とは ps コマンド等で表示される PID のことです。ジョブ番号は使用しているシェルが管理する番号で jobs コマンドで表示されます。

```
bash-3.2$ ps
 PID TTY TIME CMD
17240 pts/11 0:00 csh
7010 pts/11 0:00 ps
17341 pts/11 0:00 bash
20648 pts/11 4:29 evince
bash-3.2$ jobs
[1]+ Running evince . ./book.pdf &
bash-3.2$
```

X ウィンドウシステムが異常な状態になって端末が利用できなくなった場合には他計算機から rlogin 等を利用して、リモートアクセスし X ウィンドウシステムのプロセスを kill すると良いでしょう。firefox が原因の場合はリモートアクセスして firefox を強制終了します。この場合は firefox の lock ファイルや parentlock ファイルが残留することがあるので、これらを削除する必要があります。

```
bash-3.2$ ps -elf |grep username |grep firefox
```

で firefox の PID を確認できます。

## 2.4 Emacs(1)

Emacs<sup>1</sup> は UNIX に限らず Windows や Mac OS X 等でも最も良く利用される文書編集プログラムです。また Lisp 言語が動作する環境であり、Emacs 上で様々なプログラムが動作します。Emacs はターミナル内でも使用できますが、日本語を表示するためには日本語表示可能なターミナルが必要です。

### 2.4.1 Emacs の編集モード

Emacs<sup>1</sup> は UNIX に限らず Windows や Mac OS X 等でも最も良く利用される文書編集プログラムです。また Lisp 言語が動作する環境であり、Emacs 上で様々なプログラムが動作します。Emacs はターミナル内でも使用できますが、日本語を表示するためには日本語表示可能なターミナルが必要です。

### 2.4.2 Emacs の基本操作

Emacs の基本操作の一部を述べます。以下では、Ctrl-x のことを C-x のように表記します。

起動

```
emacs /usr/local/bin/emacs
Solaris 端末ではバージョン 24.3.1 です。
Mac OS X 端末ではバージョン 24.3.1 です。
```

起動オプション -nw  
Solaris で iim を使って日本語入力ができます。

複数のファイルを同時に編集する場合に Emacs を実行すると、その都度 Emacs が起動しますが、これはメモリの無駄な使用方法です。**Files** メニューでも確認できますが、窓を分割したり (Split Window=C-x 2)、新しい窓を作ったり (Make New Frame=C-x 5 2) できます。

カーソル移動

|     |                    |
|-----|--------------------|
| C-f | カーソルを一文字分、先に進めます。  |
| C-b | カーソルを一文字分、後ろに戻します。 |
| C-n | カーソルを一行分、下に進めます。   |
| C-p | カーソルを一行分、上に戻します。   |
| C-a | カーソルを行頭に移動します。     |
| C-e | カーソルを行末に移動します。     |

<sup>a</sup>カーソルという言葉はテキストエディタの場合にはキー gord を押した場合に文字が入力される場所のことを意味します。Emacs では四角い一文字分の領域が白黒が反転しています。

削除

|       |                               |
|-------|-------------------------------|
| Del   | カーソルの前の一文字を削除します。             |
| C-d   | カーソルの位置の一文字を削除します。            |
| C-ck  | カーソルの位置から右側を削除します。            |
| C-/   | 直前の削除を無効にします (アンドゥー)。         |
| C-x u | 直前の削除を無効にします (C-/ と同様のアンドゥー)。 |

コピー、カットアンドペースト

GUI のメニューにはこれらの機能がありますが、以下の方法を使うと効率良く編集できます。

|        |                                                                            |
|--------|----------------------------------------------------------------------------|
| C-スペース | 削除する領域の先頭にマークをつけるために用います。カーソル位置にマーク <sup>a</sup> をセットします。C-@でもマークセツトが可能です。 |
| Esc w  | カーソルの位置とマークで囲まれた領域をコピーします (コピー・バッファに書きます)。                                 |
| C-w    | カーソルの位置とマークで囲まれた領域を削除します (コピー・バッファに移動します)。                                 |
| C-y    | C-w で削除した領域をカーソルの位置に復活します (=ペースト)(コピー・バッファの内容をバッファに書きます)。                  |

<sup>a</sup>マークとはもちろん印のことですが目には見えません

<sup>1</sup> See 'man emacs': GNU Emacs

### ~ファイル操作

- C-x C-f ファイル名を指定して、ファイルの内容を読み込み、編集の対象とします(ファイルを開く、ファイルオーブン)、ファイルが存在しない場合は、新しく作れます。
- C-x C-s 編集結果を元のファイルに書き戻します。(セーブ)セーブを行うまで、ファイルの内容は元のままです。
- C-x C-w 編集中のバッファを別名のファイルにセーブします。
- C-x i 編集中のバッファにファイルの内容を挿入します。(Insert File)

### その他

- C-x C-c Emacs を終了します。
- C-g 実行しがけのコマンドを取り消します。

### 2.5 日本語入力 (1)

#### 2.5.1 Emacs を利用した日本語入力

- Emacs は JIS, SJIS, EUC, UTF-8 等の様々なエンコードのファイルに対応しています。
  - Emacs で C-\ をタイプすると日本語入力モードになります。再度、同じキーを押すと英数字入力モードに戻ります。
  - Solaris 端末では quail+kkc と呼ばれるプログラムが動作します。
  - Mac OS X 端末では SKK と呼ばれるプログラムが動作します。

これらはデスクトップ環境が提供する日本語入力機能を利用しないので、リモートアクセス時にも利用可能です。

#### • Solaris 端末の Emacs では `emasc -iim &` で起動した場合や `emasc -nw` で起動しターミナル内で利用する場合は Input Method を利用した日本語入力も可能です。この場合、C-Space が Input Method の切替の機能に割り当たられるので、文書中にマークをするためには C-@ や C-Shift-Space を使います。

- Mac OS X 端末の Emacs は Aqua インターフェース対応なので、ことえり' を利用して日本語入力が可能です。

#### 2.5.2 日本語コードの変換方法

##### 2.5.2.1 確認方法

以下の様に nkf を利用すると文字コードを確認できます。

```
nkf --guess filename 又は nkf -g filename
```

Emacs はファイルを開くと文字コードが下部のステータスバーに表示されます。u は UTF8、E は EUC のことです。

### 2.4.4 課題 2

1. テキストエディタを使用し、myprofile というファイル名でファイルを開きなさい。
2. myprofile の中に、1 行目に学生番号、2 行目に名前、3 行目に literacy1 のホームページの URL、4 行目に netnews のサーバー名を以下の形式で記載せよ。

```
studentid=xxxxxxxxx
name=xxxxxxxxx
litlhomepage=http://web-int.u-aizu.ac.jp/xxxxxxxxxx
newsserver=nhl.u-aizu.ac.jp
```

### 2.5.3 変換方法

Emacs では `C-x RET f` と打つと、ミニバッファに日本語コードの種類を尋ねるプロンプトが出ます。スペースを押すとコードのリストがバッファに表示されます。ミニバッファのプロンプトの後に euc-jp と入力して RET を押すと EUC コード(拡張 UNIX コード)でファイルを保存します。Emacs のステータスバーに E が表示されているはずです。その上で、ファイルを何か編集して再度保存するとファイルの日本語コードが euc-jp になります。

3. myprofile のファイルをセーブしなさい。このファイルは、今後の課題や練習でも使用します。
- UTF-8 に変換する場合
 

```
nkf -w filename1 > filename2
```

- EUC.JP に変換する場合  
nkf -e filename1 > filename2

## 2.5.4 Sylpheed や Firefox, OpenOffice で日本語を入力する方法

Emacs以外のアプリケーションではEmacsに固有の日本語入力支援プログラムは利用できません。Emacs以外のアプリケーションではInput Methodと呼ばれる日本語入力支援ツールを利用します。もちろん、アプリケーション自体が日本語対応の場合に限ります。

### 2.5.4.1 IIM

Solaris 端末の場合はIIMと呼ばれるInput Methodを利用します。IIMが動作しているとGnome PanelにJaかEnのボタンが現れます。もしも動作していない場合は端末内でiiimstart

を実行して動作させます。日本語を選ぶとAtokを利用して日本語入力をします。IIM対応のアプリケーションではC-Spaceを押して日本語入力機能をオン／オフします。IIMを停止する場合にはiiimstopで終了します。

### 2.5.4.2 ことえり

Mac OS X 端末では"Command+Space"を押すと"ことえり"を利用して日本語入力ができます。'ことえり'が日本語入力モードになると画面の上部にあるメニューバーに平仮名のあが表示されます。このボタン上でマウスの左ボタンを押すと"ことえり"の操作メニューを利用できます。'ことえり'はMac OS XのAquaインターフェースのためのツールなので、X window system上で動作するアプリケーションでは利用できません。

### 2.5.4.3 Mac-UIM

Mac OS X 端末でSylpheedなどのX Windowアプリケーションを使用する場合には、ことえりを利用できません。そこで、日本語入力をするためにはMac-UIMを使用します。SylpheedなどでMac-UIMを利用する場合は、Shift+Spaceを押して日本語入力に切り替えます。

### 2.5.4.4 kinput2

Solaris 端末では、いくつかのX Window アプリケーションがiiimの日本語入力を利用できません。そこで日本語入力をするためにはkinput2を使用します。kinput2を利用するためには kinput2 & をターミナルの窓であらかじめ実行しておきます。その後で, tgifなどを

起動し、tgif の場合はCtrl+Spaceを押して kinput2 の日本語入力に入り、Shift+Spaceを押して日本語入力から戻ります。

## 2.6 Unix コマンド(2) アクセス制限管理

ファイルを適切に管理するためには、以下の操作に慣れる必要があります。他の授業でも要求される操作です。充分に練習してください。Unixコマンド(l)の節ではcd, pwd, mkdir, rmdirを説明しました。ここではファイルやディレクトリのアクセス制限管理(chmod)を練習しましょう。アクセス制限管理が適切でない場合には、課題などの採点ができないことや、他の学生による課題ファイルの不正取得が発生します。各自が不利益を被らないように注意しましょう。

- chmod u+r ファイル/ディレクトリ … オーナの読み込みを許可/禁止
- chmod u-w ファイル/ディレクトリ … オーナの書き込み削除を禁止
- chmod u-w ファイル/ディレクトリ … オーナの書き込み/削除を許可
- chmod u+x ファイル… オーナによるファイルの実行を許可
- chmod u-x ファイル… オーナによるファイルの実行を禁止
- chmod u+r ディレクトリ… オーナによるディレクトリへのアクセスを許可
- chmod u-x ディレクトリ… オーナによるディレクトリへのアクセスを禁止

オーナ(u)以外のターゲットはグループ(g), その他(o),すべて(a)です。

ls -l

を利用してアクセス制限の状態を確認する習慣をつけると良いでしょう。

### 2.6.1 \$HOME/literacy のアクセス制限

リテラシーの課題提出用のディレクトリのアクセス制限を設定します。ホームディレクトリで chmod g-rwx literacy を実行しましょう。

chmod g-rwx literacy

## 第3章 ファイル管理, Emacs(2), メール, ニュース

`ls -l`で表示される内容の意味 /bin/ls -l を実行すると、例えば次のような出力が得られます。

```
total 43
-rw-r--r-- 1 s1xxxxxx student 1392 Apr 23 15:51 aaaa
-rw-r----- 1 s1xxxxxx student 15680 Apr 23 15:51 bbbbb
-rwxr-x--x 1 s1xxxxxx student 1675 Apr 19 16:10 ccc
-rw-rw-rw- 1 s1xxxxxx student 9789 Apr 22 16:42 dddd
-rw-r----- 1 s1xxxxxx student 538 Apr 20 16:08 eee
drwxr-x--- 1 s1xxxxxx student 13257 Apr 27 09:08 ff
```

### 3.1 UNIX ファイル(1)

#### 3.1.1 ファイルに関するコマンド(命令)

ここでは、UNIX の基礎的なファイル操作に関するいくつかのコマンドを紹介します。

##### 3.1.1.1 基礎的なファイル操作コマンド

UNIX には、ファイルの操作に関するコマンドとして次のようなものがあります。

|                                |                                                                                                                                                        |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ls</code>                | ファイルの一覧を表示します。ただの <code>ls</code> より詳しい情報を表示します。                                                                                                       |
| <code>ls -1</code>             | ファイルの一覧を表示します。たまたま <code>ls</code> より詳しい情報を表示します。                                                                                                      |
| <code>ls -a</code>             | 名前がピリオドから始まるいわゆる隠しファイルも表示します。 <code>.cshrc</code> , <code>bashrc</code> などを始めとする環境設定ファイルは通常は表示する必要がないのでほとんどこの形式の名前です。                                  |
| <code>cat filename</code>      | ファイルの内容を表示します。                                                                                                                                         |
| <code>cat file1 file2</code>   | <code>file1</code> と <code>file2</code> を接続して表示します。                                                                                                    |
| <code>wc filename</code>       | ファイルに含まれる文字数、単語数、行数を表示します。                                                                                                                             |
| <code>rm filename</code>       | <code>/usr/local/gnu/bin/rm</code> が利用できます。消去したファイルは元に戻すことはできないので、このコマンドを使う時は注意が必要です。 <code>-i</code> という削除の確認( <code>y/n</code> )をするための便利なオプションがあります。 |
| <code>cp file1 file2</code>    | <code>file1</code> を <code>file2</code> に複写します。                                                                                                        |
| <code>cp file directory</code> | <code>file</code> を <code>directory</code> の中に同じファイル名を、 <code>file2</code> に変えます。                                                                      |
| <code>mv file1 file2</code>    | <code>file1</code> を、 <code>directory</code> の中に同じファイル名で移動します。                                                                                         |
| <code>mv file directory</code> |                                                                                                                                                        |

#### 3.1.1.2 コマンドオプション

`ls -l` の、`ls` のあとについた `-l` のように、コマンドの後に付けられる文字(列)ことを、コマンドオプションといいます。`ls` コマンドには、ほかにもたくさんのコマンドオプションがあります。また `ls` 以外でも、大抵のUNIX のコマンドはオプションを持ちます。

- 最後の第9列目が、そのファイルの名前(ファイル名)です。
- それではファイルを取り扱うコマンドを、実際に練習してみましょう。
- 第4列目は、ファイルのグループ名。
- 第5列目は、そのファイルの大きさを表します。
- 第6列目は、そのファイルが更新された、月を表します。
- 第7列目は、そのファイルが更新された、日を表します。
- 第8列目は、そのファイルが更新された、時刻を表します。1年以上前のファイルであれば年を表します。

### 3.1.2 演習

1. `ls`, および `ls -l` を実行し, その結果を観察しなさい。
2. `cat -n myprofile` を実行し, `myprofile` の内容を表示させなさい. `cat` は文書ファイルの内容を表示することができます。しかし本来は文書ファイルを結合する命令です。`man cat` を実行し説明を読んで確認しなさい。

3. `wc myprofile` を実行し, その結果を見ながら, どの数字がなにを意味しているかを考えなさい. `man wc` も参照してみましょう。

4. `cp myprofile prof1` を実行し, `myprofile` を `prof1` に複写しなさい. `cp` はファイルをコピー(複写のことです。印刷ではありません。)します。

5. `mv prof1 prof2` を実行し, `prof1` のファイル名を, `prof2` に変更しなさい. `mv` はファイル名を変更します。

6. `rm prof2` を実行し, `prof2` を消去しなさい. `rm` はファイルを削除します。

7. `myprofile` を, `profile1` に複写しなさい。

8. `profile1` の先頭に, 

This is my profile.

 の一行を挿入しなさい. (Emacs を使う)

9. `profile1` のファイル名を `profile.a` に変更しなさい。

### 3.1.3 GUI を利用した File 管理

GUI を利用した操作の長所は視覚に頼った直感的な操作が可能なことです。

- Gnome Desktop System では File Browser というツールがあり、ファイルの複製、移動、削除、アクセスパームッシュョンの制御等が行えます。また、アプリケーションの起動も行うことが可能です。

- Mac OS X 端末では Finder というツールがあります。ファイルの複製、移動、削除、アクセスパームッシュョンの制御等が行えます。また、アプリケーションの起動も行うことができます。
- Gnome Desktop System と Mac OS X の両方で Desktop を利用しています。これはそれぞれの環境が提供するアイコンに加えてユーザの \$HOME/Desktop ディレクトリにあるファイル等が表示されます。作業効率を下げないためにも Desktop を整理すると良いでしょう。

## 3.2 Emacs(2) – 主な編集機能

### 3.2.1 ポイント, マーク, リージョン (領域)

チューントリアルにもあったように、現在のカーソルのある位置を, ポイントと呼びます。ところで, Emacs には、文章の任意の場所に、印をつけておくことができます。この印をマークと呼びます。マークは実際に目に見えるものではなく、Emacs 内部で記録されていて、新たな場所にマークが付け直されるまで、その場所に保持されます。

**C-(スペースキー), C-shift-(スペースキー), C-@ カーソルのある文字と、その直前の文字との間にマークをセットします。**

ポイントとマークにはさまれた部分を、リージョン(領域)といいます。この場合、ポイントとマークのどちらがテキストの先頭の近くにあるかということは、関係ありません。

### 3.2.2 リージョンに対する編集

Emacs では、リージョンに対する編集機能が用意されています。

**C-w リージョンの内容を削除します。削除した内容は、バッファへ保持されます。**

**C-x r (スペースキー) r カーソルの位置を, r というレジスタにコピーします。**

**C-x r j r , r というレジスタに保存された位置のところにジャンプ(カーソルを移動)しまくる(jump).**

**C-x r s r リージョン(マークとカーソルで挟まれた領域)の内容を, r というレジスタにコピーします(save).**

### 3.2.3 バッファとレジスタ

**C-w** や **C-x r x** で削除、**コピー**された内容は、それぞれバッファ、レジスタという場所に保持されます。バッファは削除された内容を一時的に保持しておく場所です。もう一度削除が実行されると、内容は書き変わります! また、レジスタとは、コピーした内容を保持していくところですが、1 文字の名前が付けられます。例えれば、

と入力すると、リージョンの内容は、a という名前のレジスタにコピーされ、保持されます。同じレジスタに再度コピーをしなければ、レジスタの内容はいつまでも保持され続けます<sup>2</sup>。前小節にてきた 2 機能に関連する機能を、以下に挙げます。

**C-x r s a**

**C-y バッファの内容を、ポイントの位置に貼付けます。(=ペースト、 paste)**

**C-x r i b レジスタ b に保存されている内容を、ポイントの位置に挿入(insert)します。(あらかじめ、C-x r s b にリージョンの内容を保存しておきます)**

上記 2つを使えば、文章の段落の移動や、同じ文章のコピーが簡単にできます。

### 3.2.4 矩形領域の編集

マークとポイントを対角とした矩形領域を削除、ペーストできます。

**C-x r k 矩形領域を削除します。削除した内容は、レジスタへ保持されます。**

<sup>1</sup> 実際は前の内容も保持されていて、あとから利用可能です。詳しくは、GNU Emacs マニュアルを見て下さい。  
<sup>2</sup> レジスタに保持される内容は、通常のリージョンの内容だけではありません。詳しくは、GNU Emacs マニュアルを見て下さい。

### 3.2.5 課題3

1. opendapsearch -x -LLL sendmailMTAkey=s125xxxx の出力を参考にメール受信サーバ名を取得せよ。
2. nslookup www.u-aizu.ac.jp の出力を参考に DNS サーバの IP アドレスを取得せよ。
3. myprofile に以下の行を追加せよ。myprofile の中に、5 行目にメール受信サーバ、6 行目にメール送信サーバ、7 行目に FTP サーバ、8 行目に DNS サーバの IP アドレスを以下の形式で記載せよ。

```
receive_server=xxxxxxxxxx
smtp_server=receive_server
ftp=ftp.u-aizu.ac.jp
name_server=xxxxxxxxxx
```

### 3.3 メールとネットニュース

メールは個人と個人の通信、ネットニュースは発信元を明らかにしたパブリックな情報発信です。使用するツールの上では同じように見える場合もあるでしょうが、インターネットの異なる種類のツールです。適切に使い分けで利用しましょう。

ニュースグループについて 学外のニュースグループは基本的に世界中の人々からアクセス可能だと考えて下さい。記事を投稿する際には

1. ニュースグループを間違えていないか、
2. 詐や誤解を招きやすい表現をしていないか、
3. 公共性がある記事を書いているか、
4. 他の人が同じ記事を書いていないか、

といった、基本的な注意事項を再確認する必要があります。

また、演習室のワークステーションでは問題がないと思われますが、個人所有のパソコンなどを利用して記事を投稿する場合には不良なソフトウェアを利用してしまい、多くのニュース利用者に迷惑をかける危険性があります。

ニュース記事の配信は次の様な仕組みになっています。ニュースのための中核となるホストコンピュータは存在しません。インターネットに接続されているニュースサーバがお互いに記事をやり取りしあって最終的に世界中のニュースサーバに記事が行き渡ります。従って投稿された記事が瞬時にどのサーバでも参照できるということはありません。投稿された記事をキャンセルすることはできますが、キャンセルするという命令自身もすべてのニュースサーバに行き渡るには時間がかかります。従って一度投稿した記事はキャンセルしても実際にキャンセルされる時間差の間に多くの読者に読まれることになります。キャンセルできるからといつてい加減な内容の記事を投稿してはいけません。

同じ記事を複数のニュースグループに投稿するときは、クロスポストを用います。これは、ニュースグループを複数並べて指定して同時に投稿するものです。同じ記事を別々に複数のニュースグループに投稿することをマルチポストといいますが、これはさて下さい。マルチポストをすると同じ内容でも別の記事として扱われるため、記事の参照などが面倒になります。また、ファイルを無駄に作ることになります。なによりも、同じ内容の記事を何度も見るのは読者にとってはとても煩わしいことです。一方、クロスポストされた記事の場合はどこかのニュースグループで一旦読めば、他のニュースグループでも既読として処理されるので、未読記事だと思って開いたら同じ内容だったということは起こりません。

クロスポスト

クロスポストのやり方は、Newsgroups: のところに投稿したニュースグループを comma (=;, コンマ) で区切って並べます。たとえば、u-aizu.course.lit1 と u-aizu.course.prog0 に投稿したいときは、

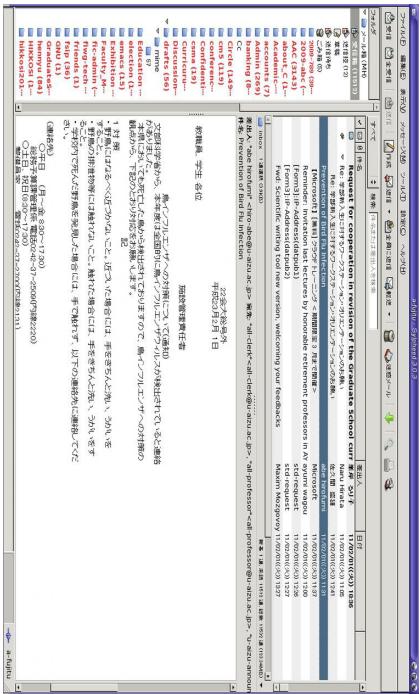
Newsgroups: u-aizu.course.lit1, u-aizu.course.prog0  
の様にして指定します。

#### 3.3.1 メール・ニュースリーダ

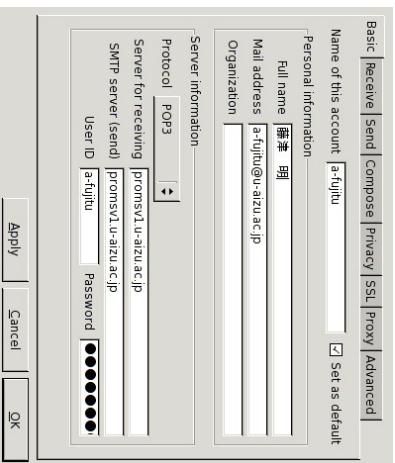
Sylpheed は新入生のために推奨されるメールツールです。予め情報センターでユーザのための設定が施してあります。新規ユーザーに対してはメールアカウントに加えて学内ニュースのアカウントも設定済なので簡単に使えるはずです。是非利用しましょう。

##### 3.3.1.1 Sylpheed の使い方

Sylpheed は新入生のために推奨されるメールツールです。予め情報センターでユーザのための設定が施してあります。新規ユーザーに対してはメールアカウントに加えて学内ニュースのアカウントも設定済なので簡単に使えるはずです。是非利用しましょう。



Sylpheed ではメールアドレスに加えて設定した名前を差出し人の名前として利用できます。Configuration の Preferences for current Account を選び、名前を設定できます。



### 3.3.2 宿題 3

u-aizu.course.list にニュースを投稿すること。投稿する記事は次の条件を満たしている者のみを採点する。

- コンピュータの利用に関する質問であるかその回答であること。
- 他のユーザが u-aizu に投稿したものと内容が異なること。
- Subject: には必ず最初に [shukudai3] と書き、続けて、内容を反映したタイトルを書くこと。

### 3.4 LATEX(0) サンプルを使った練習

自分の文章を作る前に、LATEX<sup>3</sup> を使用した文章の作成手順に慣れるためにサンプルのファイルを使って、コンパイル、レビューという一連の手順を行います。以前演習の 1 回目で作成した introduction.tex をサンプルファイルとします。そのファイルをコンパイルします。ここではコンパイルという言葉はテキスト形式で書かれたファイルを元に、画像データファイルを作ることを意味します。

コンパイル作成の手順

`plateax introduction`

<sup>3</sup>See 'man latex': structured text formatting and typesetting

を確認します。日本語が含まれていない場合は xdvi<sup>4</sup> を使います。(拡張子 .dvi は省略できます。) xdvi は日本語に非対応なので dvi からポストスクリプトや PDF に変換して表示します。

- dvips introduction.dvi -o introduction.ps  
dvi から ps を生成する。
- dvipdfmx introduction.dvi  
dvi から直接 pdf を生成する。
- ps2pdf introduction.ps &  
ps から pdf を生成する。
- gv introduction.ps &  
ps をプレビューする。
- evince introduction.pdf &  
pdf をプレビューする。

evince の代りに xpdf や pdfstudio も利用できます。また、Mac OS X 端末では Acrobat Reader も利用できます。

コンピュータ演習室 1,2,3,4 にはそれぞれ 2 台のネットワークプリンタがあります。それぞれのプリンタにはネットワーク上で識別するための名前が付けてあり、その名前によってどのプリンタに出力するか指定することができます。コンピュータ演習室 1 には std1pr1 と std1pr2, 演習室 2 には std2pr1, std2pr2 のプリンタが置かれています。

#### 注意

コンパイルした時に、ソース (introduction.tex) に間違い (バグ) があると正常にコンパイルが終了しません。エラーについての表示が出力された後に、以下のようない表示がでて?

入力待ちになります。この状態で、x を入力するとこの時点で処理は終了します。処理を続行させるには、RET, または r を押してください。その他、例えは誤ったファイル名を指定した場合等で処理がどまらなくなったら、^D(control D) を入力して下さい。

## 第4章 ディレクトリ, X Window System (2), プロセス管理, Window Manager

てから行ないなさい(演習第2回資料参照(100ページ)). また始めて実行するコマンドの場合は、オンラインマニュアルを見て説明を見なさい。(ただし、始めはマニュアルを全部理解する必要はありません。) 下の演習の指示の余白に各自でわかり易い説明をメモする下さい。

### 4.1 UNIX(2) ディレクトリ

ディレクトリとはファイルを入れておく箱の様なものです。UNIX の扱うファイルシステムは大元(おおもと)にルートディレクトリ(/)があります。ルートとは root(=根っこ)のことです。

1. cd /
2. pwd
3. ls -F  
その中に幾つかのファイルやディレクトリがありますが、home という名前のディレクトリがあります。ディレクトリとは枝別れをイメージすることも可能です。ファイルは葉っぱでしょう。では home の中に入つてみましょう。
4. cd home
5. pwd
6. ls -F  
そこにはさらに幾つかのファイルやディレクトリがあります。student という名前のディレクトリに移動してみましょう。
7. cd student
8. pwd
9. ls -F  
そこにはあなたのユーザ名を名前に持つディレクトリがあります。そこに移動してみましょう。
10. cd ユーザ名
11. pwd
12. ls -F  
さてここまで旨く移動しながら戻つて来られたでしょうか? cd という命令は(カレント)ディレクトリを移動する命令です。ここまで内容が理解できた人は次の練習をしてみましょう。ホームディレクトリに profile.a というファイルがない人は、ホームディレクトリにコピーし

### 4.2 課題 4

1. 以前作った、myprofile, profile.a, introduction.tex の各ファイルを、literacy というディレクトリの下に移動しなさい。(今後リテラシー演習で作成したファイルは .plan を除いてすべてこの中に置くことになります。)
2. myprofile, profile.a, introduction.tex の各ファイルに対して, u と o の全てに読み出しを許可しなさい。
3. literacy というディレクトリのアクセス許可が rwx---r-x に設定されていることを確認しなさい。異なる場合には正しく設定せよ。

**注意** 今後この状態に保ち、変更しない様にしなさい。これとは異なる許可状態の場合にはこのディレクトリの中に置かれるファイルは一切採点されない。

## 4.3 X ウィンドウ (2)

### 4.3.1 X ウィンドウシステムの構成と操作

X ウィンドウシステムの構成と操作法について追加の説明をします。

#### 4.3.1.1 各ウィンドウの構成

各ウィンドウは、図4.1のような構成になっています。この図では、terminal を例にとっていますが、X ウィンドウの各ウィンドウは、同じ構造 (Look and Feel) になっています。ここでは、Gnome Desktop System に基づいた Look and Feel のみを説明します。他の Desktop や Window Manager を使用している場合はここで説明が当てはまらないことがあります。Mac OS X の動作する端末でも、Aqua インターフェース上で X Window System が動作します。

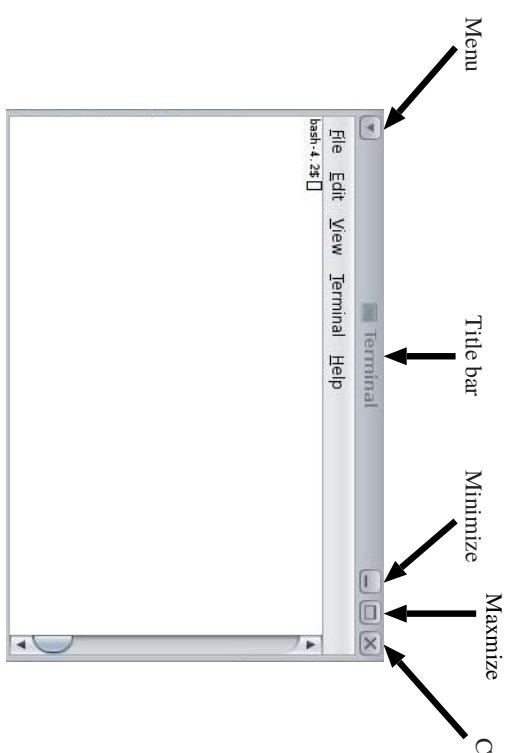
- ・ タイトルバー… そのウィンドウのタイトルが表示されます。マウスカーソルをここに持ってきて、マウスをドラッグすると、そのウィンドウを移動することができます。

- ・ 最小化ボタン… このボタンをマウスでクリックすると、ウィンドウを、アイコン (図4.2) と呼ばれる小さな記号の表示にします。Gnome Desktop System では Gnome Panel にアイコンが常時表示されています。

- ・ 最大化ボタン… このボタンをマウスでクリックすると、ウィンドウを画面一杯に表示します。このボタンはトグルスイッチになっている、もう一度クリックすると、もどります。
- ・ 終了ボタン… このボタンをマウスでクリックすると、ウィンドウメニューから、ウィンドウメニューには、以下のものがあります。

|            |                                                                                                                                       |
|------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Minimize   | 最小化ボタンと同じ動きをします。                                                                                                                      |
| Unminimize | 最小化ボタンの逆の動きをします。                                                                                                                      |
| Maximize   | 最大化ボタンと同じ動きをします。                                                                                                                      |
| On Top     | ウィンドウを手前に表示します。                                                                                                                       |
| Roll Up    | ウィンドウをタイトルバーだけに巻き上げます。                                                                                                                |
| Unroll     | ウィンドウの逆の動きをします。                                                                                                                       |
| Move       | ウィンドウを移動します。                                                                                                                          |
| Resize     | ウィンドウの大きさを変えます。                                                                                                                       |
| Close      | そのウィンドウのアプリケーションを強制終了し、ウィンドウを消します。ただし、使用しているアプリケーションによってはこのボタンで正常に終了させることができない場合があります。Emacs, Firefox などはそれぞれに指定された方法で正常に終了させる必要があります。 |

図4.1. ウィンドウの構成



ウインドウメニューの使い方 ウィンドウメニューから、ウィンドウの操作ができます。メニューには、以下のものがあります。

- ・ ウィンドウメニュー… タイトルバーやフレームの上でマウスの右ボタンをクリックすると、ウインドウメニューがポップアップします。
- ・ 操作…

ウインドウメニューの項目を選択するには、次の二つの方法があります。

- ・ ウィンドウメニューボタンにマウスカーソルをあわせ、ボタンを押し、希望の項目までマウスをドラッグし、リリースする。
- ・ ウィンドウメニューボタンにマウスカーソルをあわせ、マウスをクリックし、希望の項目をクリックする。

ウインドウのリサイズ ウィンドウの大きさを変えるには、次の二つの方法があります。

- ・ ウィンドウの枠にマウスカーソルをあわせ、ドラッグする。
- ・ ウィンドウメニューから、Size を選択する。

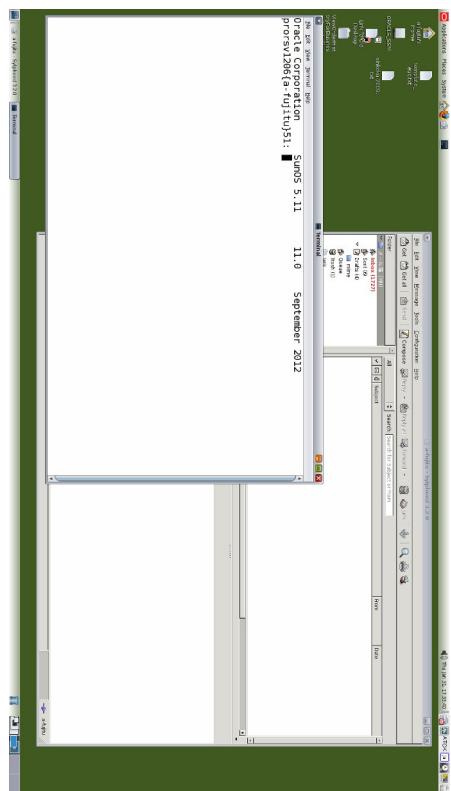


図 4.2 X ウィンドウの画面構成例 (Gnome Desktop)

アイコンをもとに戻すには アイコン化されたウィンドウを、もとに戻すには、次の二つの方法があります。

- ・アイコンをダブルクリックする。
- ・アイコンは、ウインドウメニューボタンになっているので、メニューから Restore を選択する。

タイトルバーが外に出てしまつた場合 ウィンドウメニューから、Move を選択してウインドウを移動する場合、タイトルバーもウインドウメニューボタンも、画面の外に出てしまうことがあります。このときは、ウインドウの枠の部分をマウスの右ボタンでクリックします。するととウインドウメニューがポップアップしますので、Move を選んで、ウインドウを適当な位置まで戻します。

### 4.3.2 いろいろな X ウィンドウプログラム

X ウィンドウの上で動作する様々なプログラムがあります。そのなかの幾つかを、紹介します。詳しい使い方等は、参考書やオンラインマニュアルを参照してください。

gnome-terminal  
UNIX のコマンドを受け付ける、Gnome のターミナルプログラムです。Desktop 上のマウスの右ボタンクリックメニューの Open Terminal で開きます。

Emacs  
xeyes  
xcalc  
xwininfo  
X の窓の大きさや座標などの情報を表示します。

### 4.3.3 練習

X ウィンドウの操作と、X ウィンドウプログラムの練習をしましよう。

1. terminal, Emacs のウィンドウをアイコン化し、また元に戻せ。
2. terminal ウィンドウの大きさを変更せよ。
3. terminal, Emacs のウィンドウを適当な位置へ移動せよ。
4. terminal から、

- (a) xeyes とうつて、xeyes を実行してみなさい。terminal が入力を受け付けないことを確かめよ。
- (b) xeyes を終了させよ。
- (c) xeyes & と入力し、再び実行せよ。
- (d) terminal が、こんどは入力を受け付けることを確かめよ。
- (e) xwininfo を実行し窓の情報を表示してみよ。

バックグラウンド処理 4a のように、terminal からコマンドを入力すると、入出力処理は、そのコマンドに受け継がれ、そのコマンドが終了するまで terminal に処理が返ってきません。これは、いくつもウィンドウを開いて処理をおこなうウンドウシステムには、ふさわしい利用の方法ではありません。4c のように、コマンド名に '&' をつけることにより、入出力はすべてに terminal に返ってきます。次のコマンドと、前のコマンドは、同時に平行して実行されます。terminal からみると、前のコマンドは表に出ないで、裏方で実行されているようになります。これをバックグラウンド処理といいます。

バックグラウンド処理が有効なのは、X ウィンドウシステムで動作するプログラムを起動する場合だけではありません。とくに、時間のかかる処理をバックグラウンドで実行させると、ユーザが処理の終了を待つことなく、また、新しい端末を起動せざとも、同じ端末から次の処理を実行することができるのです、大変効率的です。

```
% xeyes
^C
% xeyes &
% jobs
[1] + Running xeyes
% fg %1
xeyes
^Z
Suspended
% jobs
[1] + Suspended xeyes
% bg %1
[1] xeyes &
% kill %1
% jobs
[1] Terminated xeyes
```

## 4.4 Window Manager

プロセスの動作の状態 `xeyes &`を付けないで動作させると、`terminal`はコマンドを実行できなくなりますが、`C-z`を`terminal`内で押すとコマンドを入力できるようになります。このとき、注意深く`terminal`を見ると、

`^Z`  
Suspended

と表示されています。すなわち、`xeyes`は一時的に停止しています（終了したわけではありません）。

強調しますが、`C-z`ではプログラムは終了しません。

もとの（実行）状態に戻すには`terminal`の中でを実行します。`xeyes &`を実行した状態にするとには`terminal`の中で`bg`を実行します。`fg1`はフォアグラウンドジョブ、`bg`はバックグラウンドジョブという意味です。`jobs`を実行すると`terminal`から実行した命令が番号付きリストされます。それぞれの状態も表示されるので

`fg %ジョブ番号`のようにジョブを指定して状態を変更できます。また、異常なジョブ<sup>2</sup>

は  
`kill %ジョブ番号`の様に`kill`<sup>3</sup> コマンドを用いてプロセスを強制終了することができます。ジョブ番号の代わりに、プロセス番号を使うことも可能です。プロセス番号はpsコマンドを実行して表示されます。プロセス番号はpsファイルの場合と同じように、コンピュータで動作しているプロセスもユーザによって所有されています。ユーザは他のユーザが所有するプロセスを終了させることはできません。

### 4.3.4 stty -a

`stty -a`の出力を確認して、`^C`や`^Z`が何に割り当てられているかを確認しましょう。

注意

UNIXオペレーティングシステムは非常に安定性が高く、X Windowシステム等のユーザアプリケーションプロセスが異常に多くなることはあっても計算機システムの再起動が必要とする状態にならなければ、画面が固まつたり、キーボードやマウス等のデバイスが使用できない状態に陥ってもシステムの再起動を必要とすることは稀です。計算機システムが火や煙を吐き始めるなどの極端な事態を除いてどのような状態になつても管理者以外のユーザが電源コンセントを引き抜いたり、システムを停止したり、再起動してはいけません。

## 4.5 UNIX(3)−パイプ, リダイレクト

パイプと、リダイレクトの練習をしましよう。各操作の意味は、講義第2回資料を参照してください。

### 4.5.1 練習

次の操作を端末（=gnome-terminal）の中で行いなさい。操作は全て、ホームディレクトリの下の`literacy`（=\$HOME/literacy）の中で行いなさい。ただし以下の例の中の`sXXX`の部分は自分の学籍番号を参照して適切なものに変更しなさい。

<sup>1</sup> See 'man fg': jobs, fg, bg, stop, notify - control process execution

<sup>2</sup> 黒常なプロセス、何かの理由によって正常に終了せずに残るプロセス。`"run"`の状態であり続けねばシステムパフォー  
マンスが著しく低下するので好ましくない。

<sup>3</sup> See 'man kill': terminate or signal processes

#### 4.7 GUI とターミナルモード

各種アプリケーションがGUIで提供されることが多くなった現代でも、ターミナル内で動作するプログラムは重宝されます。ターミナルを利用したコンピュータの利用は伝統的な利用方法です。ですから、"古い"とか"不要だ"とか思うかも知れません。しかしながら、現在も、WindowsやMac OS Xであってもターミナルは捨てきれない様です。昔からのソフトウェア資産の利用を継続するためという理由はすぐ思いつますが、ターミナルを捨てられない理由は他にもあるでしょう。以下にターミナル内の動作や操作の幾つかの例を紹します。

1. ls > file.list
2. cat file.list
3. cat file.list | more
4. ls -l | more
5. ls | wc
6. ls -l /usr/bin | more
7. getent passwd | grep sXXX
8. getent passwd | grep sXXX | sort
9. getent passwd | grep sXXX | sort > new.list

- リダイレクト(>)によってコマンドの出力をファイルに記録が出来たことを確認しましょう。

- パイプ(|)によってコマンドの出力を次のコマンドのデータ入力として処理できたことを確認しましょう。

**getent**について getent passwd というコマンドは, LDAP<sup>4</sup>で管理されているユーザのログイン名やホームディレクトリなどのデータを表示させます。

7で, sXXX を s110 とした場合には getent passwd が 出力するデータから s110 の文字列データを含む行、つまり、10期生のデータを抜き出しています。この様に、情報を取り出す操作を「フィルタリング」と呼びます。(コーヒー豆とお湯の混ざったものから、コーヒーを抽出するのに"コーヒーフィルタ"を使います。そんな感じです。) 9では、それを new.list という名前のファイルに書き出し(> new.list)しています。

#### 4.6 宿題4(タピングの練習)

配布された資料には、「タピングの練習」用にC言語の簡単なプログラムを集めてみましたが(付録C.189ページ)。プログラミングは始めて!と言う人も特に緊張する必要もあります。タピング練習用C言語プログラムのすべてをそれぞれ自身でタイピして指定されたファイル名(ソースプログラム名)に保存しなさい。Emacsを使ってタイピして下さい。

プログラムの楽しみ方 楽しみたい人はコンパイルして実行してみて下さい。コンパイラはgccを使って下さい。(see 'man gcc') コンパイルという言葉の意味は変換するということです。ただし、ここではテキスト形式で書かれたプログラムを元に、実行可能な命令を作ることを意味します。

- GUIを表示しないでアプリケーションを利用する事が快適な場合もあります。
  - ImageMagickの画像加工処理はGUI表示のあるdisplayとGUI非表示のconvertがあるので、適切に使用する。
  - pdf2psを利用してpdfをpsに変換する場合はGUI非表示で実行できる。
  - rsh stdrv1 last | more stdrv1のログイン状況を表示します。
  - rsh stdrv1 top stdrv1の動作プロセスの様子を表示します。

## 第5章 印刷

### 5.1 印刷の方法

各演習室にはネットワークプリンタが設置されています。ファイルの印刷は以下 の方法が利用できます。必ず、最初に利用するプリンタのプリンタ名と機種名を確認して下さい。

| 機種名              | タイプ            | 配置されている演習室等                        |
|------------------|----------------|------------------------------------|
| OKI B841         | ポストスクリプト対応プリンタ | 演習室 1,2, CALL1,2 ハードウェア実験室 1,2,3,4 |
| EPSON LP-S4200PS | ポストスクリプト対応プリンタ | 演習室 3,4,5,6, iLAB1,2               |

印刷を行なうために lpr コマンドを利用します。プリンタの指定や、ファイルの形式の指定、出力の形式を指定するためにオプションスイッチを付ける必要があります。オプションスイッチは機種により異なることがあります。

[ポストスクリプトファイルの印刷] ポストスクリプト対応プリンタでポストスクリプト(PostScript)と呼ばれる言語で記述された画像ファイル(一行目が%!PS-Adobe で始まるテキスト形式のファイル)を印刷する方法は

```
lpr -Pプリンタ名 ポストスクリプトファイル名
```

です。

印刷の前に必ず印刷するページ数を確認しましょう。印刷を実行する前に ps 形式のファイルのイメージを確認する習慣を身につけておくことは、無駄な印刷を減少させるためにとっても良いことです。gv<sup>1</sup>を利用すれば確認できます。

```
gv ポストスクリプトファイル名 &
```

#### 注意事項

印刷出力は必ず印刷した本人が回収すること、無駄な印刷をしてはいけません。出力数は情報センターが監視し、異常な量を印刷する者に対しては相当の処分を行ないます。

[印刷の待ち状態の確認]

```
lpq -Pプリンタ名
```

で指定したプリンタの印刷状態を表示できます。印刷処理それぞれにジョブ番号がついているのを確認しましょう。

[印刷の取消] 印刷の待ち行列に並んでいるジョブや印刷途中のジョブは取り消すことができます。  
1prm -Pプリンタ名 ジョブ番号

を実行します。ただし、最近のプリンタのメモリは大容量になり、場合によっては印刷のイメージデータを全て、プリンタのメモリに瞬時に取り込んで印刷することができます。そのため印刷が始まってしまっては、lprm で印刷を取り消すことができない場合があります。

#### 応急処置

大量の紙を無駄に消費しているような緊急の場合にはプリンタの電源スイッチを一旦切り、メモリをクリアすることも必要です。電源をスイッチを印刷中に切った場合には紙詰まりが発生することもあるので取り除く必要が生じます。

何より無駄な印刷をしないように心がけましょう。

#### 5.1.1 テキスト形式ファイルをポストスクリプト形式のファイルに変換する

テキスト形式のファイルをそのまま印刷することはできません。必ずポストスクリプト形式のファイルを生成します。テキストファイルを印刷するため postscript を生成するツールとして、k2ps があります。これは、EUC コードの日本語に対応しています。UTF-8 に対応している a2ps も利用できます。

```
nkf -e textfile | k2ps > textfile.ps
または
a2ps textfile > textfile.ps
```

(textfile には出力したいファイル名を入れます)。得られたポストスクリプトファイルを gv で確認し、問題が無ければ lpr を用いて印刷します。k2ps -help で k2ps の利用方法を表示できます。k2ps には、上記のものとは別に gnu が提供するものがあります。混乱をさけるために本学では、a2ps\_gnu という名前にしてあります。

#### 5.1.2 DVI ファイルをポストスクリプト形式のファイルに変換する

T<sub>E</sub>X(L<sub>A</sub>T<sub>E</sub>X) 組版システムによって生成される DVI ファイルをそのまま印刷することはできません。必ずポストスクリプト形式のファイルに変換(=Convert) します。DVI ファイル foo.dvi を PS 形式のファイル bar.ps に変換する場合には dvips foo.dvi -o bar.ps を実行します。得られたポストスクリプトファイルを gv で確認し、問題が無ければ lpr を用いて印刷します。

<sup>1</sup> See "man gv": a PostScript and PDF previewer

注意事項: ポストスクリプトファイルはテキスト形式で書いてあり、PS 対応プリンタがその文章に書いてあるポストスクリプト命令を解釈して画像を出力します。実際の内容は more

コマンドで確認することができます。ポストスクリプトファイルを画像としてではなくテキスト形式として印刷すると大量のページになってしまい、森林資源を膨大に消費することになります。(Save the earth!!) ちゃんと印刷できるかどうか不安を感じる場合には `gv` 等などのPreviewerを利用して印刷出力のイメージを確認することが必要です。

### 5.1.3 PDF 形式のファイルの印刷

- Acrobat Reader

Adobe 社の PDF (Portable Document Format) 形式のファイルを印刷するためには Mac OS X 端末では Adobe 社の Acrobat Reader を利用できます。印刷のためには `File` メニューから **Print** を選び、図 5.1 を表示させて、印刷するプリントを指定したり、他のオプションを適切に設定します。

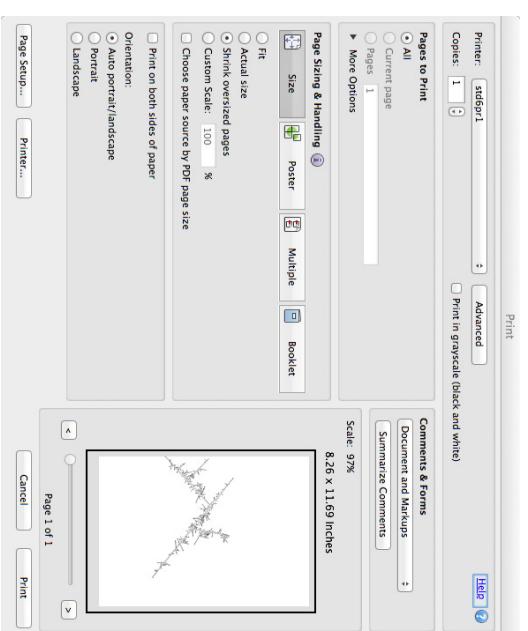


図 5.1. Acrobat Reader の印刷オプション設定

### 5.1.4 便利なオプション

印刷の命令 `lpr` に両面印刷のオプションを利用して両面印刷が可能です。機種によって異なるので使い分けてください。CUPS のオプションか LPRng のオプションが利用できます。Mac や Solaris11 の `/usr/bin/lpr` では (標準の環境) CUPS のオプションを使用します。

|               |                          |
|---------------|--------------------------|
| 両面印刷(ポートレート)  | -o Duplex=DuplexNoTumble |
| 両面印刷(ランドスケープ) | -o Duplex=DuplexTumble   |

Solaris11 の `/usr/local/LPRng/bin/lpr` は以下のオプションです。

|               |                 |
|---------------|-----------------|
| 両面印刷(ポートレート)  | -Cduplex        |
| 両面印刷(ランドスケープ) | -Cduplex,tumble |

会津大学の計算機システムでは毎年一部のシステムを更新しています。プリンタの機種の変更により使用方法が変更される場合があるので機種更新について毎年確認する必要があります。詳しくは情報センターのホームページを参照してください。  
<http://web-int.u-aizu.ac.jp/Labs/istc/ipc/release/release.html>

### 5.1.5 便利なツール

#### 5.1.5.1 psutils

ポストスクリプトファイルをアレンジする様々なツールがあります。1枚に複数のページをレイアウトするものや、一部のページを抽出するものなど大変役立つことでしょう。

**psutils** Collection of useful utilities for manipulating PostScript documents

**psbook** Rearranges pages into signatures

- `psbook in.ps out.ps`

**psselect** Selects pages and page ranges

- `psselect -p2-4 in.ps out.ps`

**pstop** Performs general page rearrangement and selection

- `pstop '2:@L@.7(21cm,0)+11@.7(21cm,14.85cm)' in.ps out.ps`

**psnup** Put multiple pages per physical sheet of paper

- `psnup -2 in.ps out.ps`

**psresize** Alter document paper size

- `psresize -PA4 -pletter in.ps out.ps`

**epsfit** Fits an EPSF file to a given bounding box

- `epsffit 11x11y urx ury in.ps out.ps`

**psmerge** filter to merge several PostScript files into one.

- `psmerge -oout.ps in1.ps in2.ps ... inN.ps`

**beamer0.ps** が生成されます。他にも、PDF ファイルからポストスクリプト形式のファイルを生成するプログラムがあります。オンラインマニュアル(`man -k pdf`)を利用して検索してみましょう。ポストスクリプトファイルは比較的ファイルのサイズが大きいので印刷が終了したら削除しましょう。

### 5.1.5.2 pdfjam

PDF に対して psutil と同等の処理をするツールです。texlive にパッケージされています。

**pdfjam** pdfjam is a shell-script front end to the LaTeX `pdftools` package

**pdfbook** Rearranges pages into signatures

- **pdfbook** in.pdf --outfile out.pdf

**pdfjam** Selects pages and page ranges

- **pdfjam** in.pdf '10,{},3-6,11-' --outfile out.pdf

**pdfnup** Put two pages per physical sheet of paper.

- **pdfnup** in.pdf --outfile out.pdf

**pdfjoin** filter to join several PDF files into one.

- **pdfjoin** in1.pdf in2.pdf ... inN.pdf --outfile out.pdf

### 5.1.6 ハンドアウトの印刷

もしもハンドアウトを忘れてしまったら、演習の時には WS の画面で

evince /home/course/lit1/pub/lit1/2017/sum/book.pdf  
とすれば参照できるので、森林資源を保全するためになるべくハンドアウトの印刷を行わずに画面表示を利用して演習に参加しましょう。リテラシー 1(2017 年度) のホームページは次の場所です。  
<http://web-int.u-aizu.ac.jp/course/lit1/>

### 5.1.7 Web Page の印刷 (Mozilla Firefox)

#### 5.1.7.1 Page Setup

Firefox のウインドウの右上部にある **メニュー** をマウスで押して、フルダウンメニューを出します。メニューの中にある **Print** を押すと印刷のイメージが表示されます。このときの窓の上部にある **Page Setup** を押すと印刷イメージのページレイアウトを設定する機能を利用できます。窓(図 5.2)の中でポートレート(縦書き)からandscape(横書き)か、用紙の大きさを設定できます。 **Print...** メニュー内の Options タブ内で印刷イメージのヘッダー やフッターの情報を設定できます。

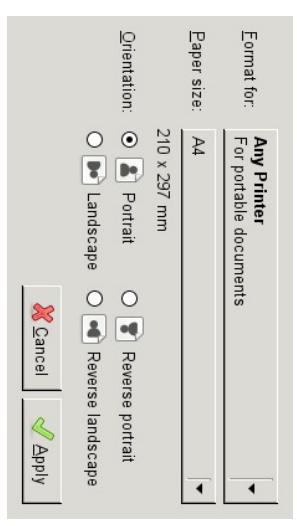


図 5.2. Mozilla Firefox のページ設定

#### 5.1.7.2 Print

**Print...** メニュー内の General タブでは印刷が実行できます。これを選ぶと印刷のための窓(図 5.3)が現れます。演習室のプリンターがリストされれば、それを選択して印刷できます。

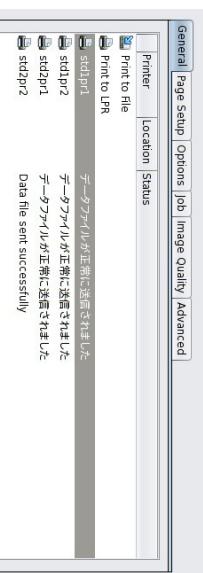


図 5.3. Mozilla Firefox の印刷窓

もしも、ここで **Print to LPR** を選択した場合は、手書きで Command Line の中に印刷の命令をプリント名を指定します。Page Setup のタブメニューで両面印刷等のオプションも設定できます。ここに書いた命令は保持されるので、演習室を移動した場合等はプリンタを適切に変更します。



図 5.4. Mozilla Firefox のページ設定

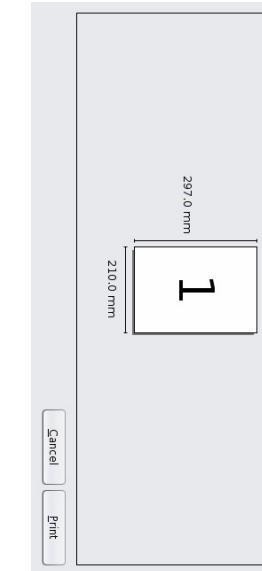


図 5.5. Mozilla Firefox のファイルへ印刷

### 5.1.7.3 プレビュー

印刷を実行する前に印刷のイメージやページのレイアウト等を確認することをプレビューといいます。Page Setup では印刷するページを絞り込むことや段組印刷も設定できます。

### 5.1.7.4 Print to File

印刷のイメージを File に保存すると、他のツールでイメージの加工をしたり、無駄な印刷を少なくするために利用できます。また、記録としても利用できます。Print... メニュー内の General タブで Print to File を選択すると、印刷イメージをファイルに保存できます。これを選ぶとファイルに出力するための窓(図 5.5)が現れ、ファイル名を設定したり、保存形式を設定できます。Solans 端末では PDF か PostScript(=ポストスクリプト)形式を選択できます。PDF で保存されたファイルは pdftudio, xpdf, evince, Acrobat Reader あるいは gv で確認できます。PostScript 形式のファイルは gv 等で確認できます。また、psutils などを利用して紙の消費を少くできます。以下は psnup を利用して 1 つの用紙に 2 ページ分をまとめる方法です。

```
% psnup -2 mozilla.ps a.ps
% gv a.ps &
```

を試してみましょう。man psnup で詳しく使用法を確認できます。

```
pstopos '2@L@.7(19cm,0)+1@.7(19cm,15cm)' mozilla.ps a.ps
```

も試してみましょう。

Mac ⇒ Windows の Firefox  
Mac 版の Firefox ⇒ Windows 版の Firefox では postscript ファイルは生成しません。



### 5.1.8 課題 5

今まで作成した課題のテキストファイルのどれかを印刷して学籍番号を明記して提出して下さい。

1. k2ps を用いて以下の条件を満す ps ファイルを作成する。

- ポートレイト
- フォントサイズは 12 ポイント

2. 印刷ジョブの待ち状態を観察して下さい。(lpq)
3. 印刷ジョブのキャンセルを実行してみて下さい。(lprm)

### 5.2 宿題 5(タイピングの練習)

配布された資料には、タイピングの練習、用に L<sup>A</sup>T<sub>E</sub>X の簡単なサンプルを集めてみました。(付録 D.191 ページ)。L<sup>A</sup>T<sub>E</sub>X は始めて！と言う人も特に緊張する必要もありません。Emacs を使ってタイプして下さい。タイピング練習のすべてをそれぞれ指定されたファイル名(ソースプログラム名)に保存しないでください。それらを各自のホームディレクトリの下のディレクトリ literacy に移動しておいて下さい。

## 第6章 UNIXの復習

### 6.1.4 ジョブの管理

この節の内容はこれまでに紹介された命令の復習です。UNIXの命令はターミナル(端末, xterm, gnome-terminal等)の窓の中でキーボードからタイプして実行できます。以下に簡略な説明をしますが、決して完全な説明ではありません。詳しくは、オンラインマニュアル(man)等を参照して下さい。もしもわからないことがあれば今回の演習中に解決するため先生等に質問して下さい。

### 6.1.1 ディレクトリの内容を出力する命令

`ls [option] path-list`

`/usr/local-gnu/bin/ls` には主に以下のようなオプションがある。

- F ファイル名の後に、ディレクトリ(/), リンク(@), 実行可能ファイル(\*)を示すフラグを表示する。
- l 長い形式で表示する。ユーザ、グループ情報とともに表示。
- a ドット「.」で始まるファイルおよびディレクトリを表示する。
- g 長い形式で表示する。グループは表示。ユーザは非表示。

`/usr/bin/ls` と `/usr/local-gnu/bin/ls` のオプションに対する動作は多少違います。

### 6.1.2 ファイルの内容を出力する命令

`cat file1 file2 ... fileN` リストにあるファイルを全て連結して出力する。

`more file` `file` の内容を terminal の 1 画面分出力する。

`less file` `file` に逆スクロールの機能を追加したもの。

### 6.1.3 ネットワークやホストの管理情報

|                                             |                            |
|---------------------------------------------|----------------------------|
| <code>getent</code>                         | Network 管理情報を標準出力に出力する命令   |
| <code>getent passwd   grep a-fuji tu</code> | ユーザーのログイン名を知る。             |
| <code>whoami</code>                         | ホストにログインしているユーザーを知る。       |
| <code>who</code>                            | ホストにログインされるユーザーの情報を知る。     |
| <code>finger -m name</code>                 | name で指定されるユーザーの情報を知る。     |
| <code>finger @hostname</code>               | hostname にログインしているユーザーを知る。 |

### 6.1.5 ディレクトリの操作

|                    |                       |
|--------------------|-----------------------|
| <code>cd</code>    | ディレクトリを移動する。          |
| <code>mkdir</code> | 新しくディレクトリを作る。         |
| <code>rmdir</code> | ディレクトリを削除する。          |
| <code>pwd</code>   | カレントワーキングディレクトリを表示する。 |

### 6.1.6 ファイルの操作

|                                     |                         |
|-------------------------------------|-------------------------|
| <code>cp file1 file2</code>         | ファイルをコピーする。             |
| <code>cp file-list directory</code> | ファイルをディレクトリに同じ名前でコピーする。 |
| <code>mv file1 file2</code>         | ファイルの名前を変更する。           |
| <code>mv file-list directory</code> | ファイルをディレクトリに移す。         |
| <code>rm file-list</code>           | ファイルを削除する。              |

### 6.1.7 フィルタ

フィルタとはご存知の通り、コーヒーフィルタの様に、漉す、プログラムのことです。そもそも良く使われるのが grep です。

|                                    |                                |
|------------------------------------|--------------------------------|
| <code>grep majoren</code>          | 文字列のある行だけを出力する。                |
| <code>sort file</code>             | 入力をアルファベット順に並べ替える。             |
| <code>head -n file</code>          | 入力から最初の n 行を表示する。              |
| <code>tail -n file</code>          | 入力からお尻の n 行を表示する。              |
| <code>sed 's/foo/bar/' file</code> | file 中の文字列 foo を文字列 bar に置換する。 |

## 6.1.8 ネットワークを利用したログイン

ネットワークで接続されていて、しかもログインの許可を有しているなら他のホストにネットワーク越しでログインできます。

|        |                         |
|--------|-------------------------|
| rlogin | リモートログイン rlogin stdrsv1 |
| rsh    | リモートシェル rsh stdrsv1     |

リモートログインではログインの認証のパスワード等の文字が暗号化されずにPlain-Textのままネットワークを流れています。従って、インターネット経由では暗号化を利用した、より安全な(セキュアな)ツールを利用します。  
会津大学では学外からのrsh, rlogin, telnet, ftpを利用したアクセスは制限しています。学外から会津大学のネットワークへアクセスするためには一旦 ssh サーバにアクセスし、そこを経由して演習室計算機等にログインします。

```
ssh username@sshgate.u-aizu.ac.jp
```

```
process25@a-fujitsu58: ssh a-fujitua@sshgate.u-aizu.ac.jp
```

```
a-fujitu@sshgate.u-aizu.ac.jp's password:
```

```
Warning: Remote host denied X11 forwarding, perhaps xauth program could not be
```

```
un on the server side.
```

```
Last login: Thu Feb 1 16:50:35 2007 from sshgate
```

```
No mail.
```

```
sshsv3(a-fujitsu51: rsh std2dc3
```

```
Password:
```

```
Last login: Thu Feb 1 16:52:11 from sshsv2
```

```
Sun Microsystems Inc. SunOS 5.10 generic January 2005
```

```
std2dc3(a-fujitsu51: std2dc3@fujitsu51:■
```

命令を実行することもできます。

```
rsh rsh stdrsv1 who などとやってみると stdrsv1 で who コマンドを実行する。
```

## 6.1.9 その他の命令

端末プログラム(gnome-terminal等)の中で動作するもの。

|                 |                                |
|-----------------|--------------------------------|
| exit            | 終了する。                          |
| logout          | ログアウトする。                       |
| wc              | 行末(行の終り), ワード, 文字の数をカウントする。    |
| cal             | カレンダーを表示する。                    |
| man             | オンラインマニュアルを表示する。               |
| lpr, lprq, lprm | 印刷関係の命令                        |
| pdflatex        | TeXを処理する(file.tex -> file.dvi) |
| which command   | コマンドがどこにあるかを表示する。              |
| emacs -nw       | ターミナル内で編集する。                   |
| vi, vim         | UNIX標準テキストエディタ。                |

ターミナルの外に新しい窓を作り動作するもの。

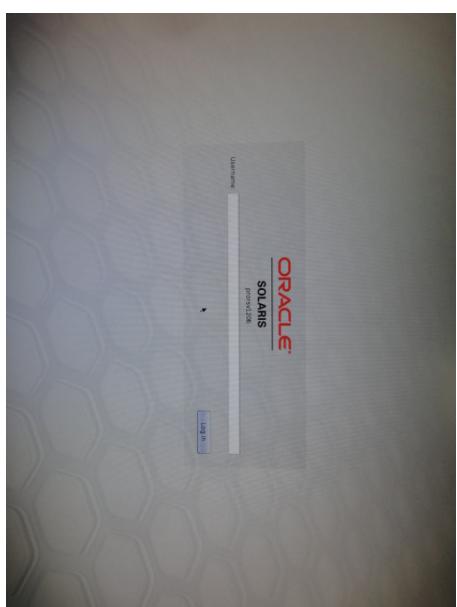


図 6.1. gdm login

## 6.1.10 ログインとX Window System

### 6.1.10.1 GUIベースのログインの場合

Solaris11ではGUIインターフェースを利用してユーザーがログインする機能をGnome Desktop Managerが提供します。あらかじめ、管理者所有のX Window Systemプロセスが動作します。

一般的な名称としてはxdm(X Display Manager)です。演習室のSolaris端末では図(6.1)の様なログインスクリーンが表示されています。

図(6.1)はxeyes & の様にアンパーソンド(&)をつけてバックグラウンドジョブにしておきます。

|                  |                                  |
|------------------|----------------------------------|
| gnome-terminal & | Gnome Terminal を起動する。            |
| xeyes &          | 目玉電卓                             |
| xcalc &          | テキストエディタ                         |
| emacs &          | テキストエディタ file.dvi を画面に表示する。      |
| gedit &          | ポストスクリプトファイル & PDF ファイルを画面に表示する。 |
| xdvi file &      | PDF ファイルを画面に表示する。                |
| gv               | ポストスクリプトファイル & PDF ファイルを画面に表示する。 |
| pdfstudio PDF    | PDF ファイル & PDF ファイルを画面に表示する。     |
| evince PDF       | PDF ファイル & PDF ファイルを画面に表示する。     |
| xpdf PDF         | PDF ファイル & 様々な種類の画像ファイルを画面に表示する。 |
| display 画像ファイル & | 画像ファイルを画面に表示する。                  |

gdm のログインでは次の図(6.2)にあるように Gnome Desktop と xterm の 2 種類から選択できます。

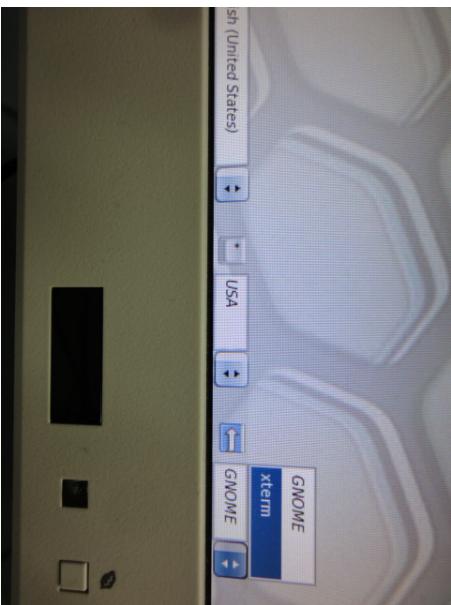


図 6.2. Select Desktop

Gnome Desktop System の場合にはログイン時に \$HOME/.gnome2/session-manual に書かれてあるプログラムが実行されます。このファイルはデスクトップの System メニューから Preferences → Startup Applications で編集可能です。

Solaris 11 に付属の Gnome Desktop System は GNOME 2.30.2 ベースのデスクトップ環境です。GNOME の最新版は 2012/09 現在は 3.6 であって Linux の各種ディストリビューションや FreeBSD などで利用できます。

確かめてみよう

演習室のログインスクリーンを表示しているプログラムを ps コマンドで確認してみましょう。ps 命令に適切なオプションを付けて利用すると全てのプロセスが表示されます。

オーナー(ユーザ:u), グループ(g), その他(o), そして各々に対して読みだし(r), 書き込み(w), 実行(x)のアクセス許可が設定できる。属性を確認する方法を知らない人は /usr/bin/ls -l を実行して、その出力をよく観察してみましょう。

### 6.1.12 chmod の練習

例に習って説明を加えなさい。

|                     |                             |
|---------------------|-----------------------------|
| chmod a-w file      | 全てのユーザ(ugo)に対して書き込み許可がなくなる。 |
| chmod u+w file      |                             |
| chmod o-r directory |                             |

## 6.2 標準入出力とパイプ, リダイレクト

標準入出力とは、UNIX の命令が使用する標準の入力や出力を行なう対象の装置のことであります。UNIX の命令は標準的な入力装置としてキーボードを扱い標準的な出力装置として(ディスプレー上の)ターミナルを使用します。

リダイレクションとは re-direction ("再"-->"向き付け"), すなわち UNIX では標準入出力の向き付けを変更することを意味します。パイプとは水管をイメージすると良いでしょう。パイプには二つの口があり一つを入り口もう一つを出口とします。一つの命令(lsなどのコマンド)を一つのパイプと見立てればその命令から出力される情報を次の命令の入力として次の命令を実行することが二つの命令をつなぐことをイメージさせます。そこでそのような仕組みや、それを実現する処方をパイプと呼びます。

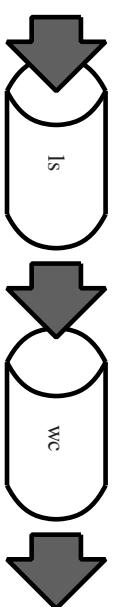


図 6.3. Pipe を繋ぐ

ファイルやディレクトリには次のような属性がある。

|   | 読みだし | 書き込み | 実行 |
|---|------|------|----|
| r | w    | x    |    |

UNIX には cat や、ls, more, sort, grep を始め多くの小道具(小さな命令)が用意されています。これをパイプで繋いで処理することによって多様な情報処理を実現することができます。以上の説明を理解するためにも以下が同じ処理を実行することを確かめることは有意義です。

- cat *file* | grep *mojiretu*

1. grep *mojiretu* < *file*
2. grep *mojiretu* *file*

なぜ同じ結果を得るかを考えて下さい。cat コマンドをターミナルで実行すると cat はキー入力からの入力を待ちます。C-d をタイプすると入力ファイルの終了(EOF=EndOfFile)と同じことになり cat は動作を終了します。標準では cat コマンドの出力はディスプレーに出力されますが、これをリダイレクトを利用して置き換えることが可能です。

*cat > filename*

これはコマンドの出力先を *filename* という名前のファイルに置き換える方法です。したがってこれは新しいファイルを作成したいときにも利用できます。以下に示すのは、キーボードから打ち込んだ文字を *file1* というファイルに格納する方法です。cat > *file1* で、キーボードから打ち込んだ文字を *file1* というファイルに格納するできます。ファイルに格納したい文字をすべて打ち込んだら、C-d を打ちます。以下に示す例では、C-d は省略しています。

1. \$ HOME/literacy/`whoami` > stdlist
2. whoami を実行し、その出力結果と stdlist に記録せよ。
3. getent passwd |grep XXXX | wc >> stdlist
4. を実行し、自分と同じ年度に入学した学生の数を stdlist に追加記録せよ。ただし XXXX の部分には自分自身の学籍番号を参照し、適切な 4 文字の文字列に置き換えること。

1. seq -100 -2 3000 > file
2. cat -n file

でテキストファイル *file* の各行頭に行番号を付けて出力される。

1. 文字列 100 を含む行を検索する処理 grep 100 file を実行し *file* の 100 行目の出力を求めよ。
2. grep 100 file | od -a を実行し各行の先頭の文字のパターンを観察せよ。
3. grep の -e オプションあるいは、-f オプションを用いて *file* の 100 行目のみを表示する命令を作り、cat100 に書きなさい。

### 6.3 UNIX のファイルシステム

UNIX のファイルやディレクトリを絶対パス名で記述すると

/home/course/lit1/pub/literacy/2017/sum/book.dvi  
の様になっています。UNIX の命令 ls では(ディレクトリや、ファイル名を指定するため)引数を省略した場合に仮想的に自分が居るディレクトリ(current working directory)直下にあるファイルやディレクトリのみを表示します。そのおかげでディレクトリを用いたファイルの整理が比較的容易になります。

UNIX ではユーザーが設定している PATH という環境変数のディレクトリストを利用しています。従って、命令の絶対パス名を入力しなくても PATH に登録されているリストから自動的に見つけて実行する仕組みになっています。

PATH の内容を確認してみましょう。

% echo \$PATH

ls コマンドのフルパスを確認してみましょう。

% which ls

### 6.4 課題 6

1. \$ HOME/literacy/`whoami` > stdlist
2. whoami を実行し、その出力結果と stdlist に記録せよ。

3. getent passwd |grep XXXX | wc >> stdlist
4. を実行し、自分と同じ年度に入学した学生の数を stdlist に追加記録せよ。ただし XXXX の部分には自分自身の学籍番号を参照し、適切な 4 文字の文字列に置き換えること。

### 6.5 宿題 6

## 第7章 文字列検索と文字列置換

**7.1.1 バイナリファイルを文字列検索する**

`strings` というツールを使用するヒファイル中に含まれる印字可能な文字列を出力します。<sup>1</sup> 例えは、`/bin/csh` 中の `too` という文字列を含む部分を出力するには

```
strings /bin/csh | grep "[Tt]oo"
```

とします。[Tt] は T または t のいずれかの一文字を表す正規表現です。

文書に書かれている情報を利用する場合に特定の文字列が文書に含まれているかどうかを調べると役に立つ場合があります。その様な場合には文字列検索の機能を使用します。また、文書中の特定の文字列を置換し、新規文書として再利用することもあるでしょう。

ワードやエクセルなどのオフィスツールでも検索や置換の機能があり、これらの機能を使い、こなすことで作業効率を高めることができます。

また、数万行の情報から特定の文字列を含む部分を抽出したり、数千の文書ファイルから、特定の文字列を含むファイルを特定することが、情報処理として必要な場面もあるでしょう。

ここでは文字列検索と置換の基本的な操作を学びます。

### 7.1 文字列検索

`/home/course/lit1/pub/literacy/ConstitutionOfJapan.txt` に日本国憲法の英語版が置いてあります。このファイルを利用するので、`$HOME/literacy` にコピーしましょう。

```
cd ~/literacy
```

```
cp /home/course/lit1/pub/literacy/ConstitutionOfJapan.txt .
```

通常は、特定の文字列を含む行を表示する場合は `grep` コマンドを利用します。

```
% grep ^CHAPTER ConstitutionOfJapan.txt
```

```
CHAPTER I. THE EMPEROR
```

```
CHAPTER II. RENUNCIATION OF WAR
```

```
CHAPTER III. RIGHTS AND DUTIES OF THE PEOPLE
```

```
CHAPTER IV. THE DIET
```

```
CHAPTER V. THE CABINET
```

```
CHAPTER VI. JUDICIARY
```

```
CHAPTER VII. FINANCE
```

```
CHAPTER VIII. LOCAL SELF-GOVERNMENT
```

```
CHAPTER IX. AMENDMENTS
```

```
CHAPTER X. SUPREME LAW
```

```
CHAPTER XI. SUPPLEMENTARY PROVISIONS
```

この実行例は `ConstitutionOfJapan.txt` の中の行頭に文字列 `CHAPTER` がある行を出力しています。`grep` 以外のツールで同様のことが実現できます。

```
sed -n '/^CHAPTER/p' ConstitutionOfJapan.txt
awk '/^CHAPTER/ {print}' ConstitutionOfJapan.txt
perl -ne 'm/^CHAPTER/ && print' ConstitutionOfJapan.txt
```

### 7.2 文字列置換

文字列を置換するツールとして代表的なものは `sed` です。`sed` ユーティリティは、テキストファイルを読み込み、編集コマンドのスクリプトに従って内容を編集し、結果を標準出力に書き出すストリームエディタです。今回の演習では文字列置換の機能に的に絞って `sed` を利用します。`sed` の文字列置換の機能は `s` 命令を利用します。

```
s/regexp/replacement	flags
```

`regexp` は正規表現です。この命令は文書の各行に対して `regexp` のマッチを試みます。行中にマッチした部分があればその部分を `replacement` に置換します。`replacement` は特殊文字である `&` を含むことができ、これはマッチした部分を参照します。また `\1` から `\9` までの特殊エスケープを含むこともでき、これは `regexp` の副表現 (sub-expression) にマッチした部分をそれぞれ参照します。`ConstitutionOfJapan.txt` には `JAPAN` という文字列があります。`JAPAN` をすべて `AIZU` に置換するには `Japan` をすべて `Aizu` に、`Japanese` をすべて `Aizu` に置換するには

```
sed -e 's/JAPAN/AIZU/g;s/Japan /Aizu /g;s/Japanese/Aizu/g' ConstitutionOfJapan.txt
```

を実行します。この出力を `ConstitutionOfAizu.txt` に保存して、元の `ConstitutionOfJapan.txt` と `diff` ツールを利用して比較をしてみましょう。

```
diff ConstitutionOfAizu.txt ConstitutionOfJapan.txt
```

`diff` は 2 つのファイルの違いを出力します。正確に置換したことを探認できるでしょう。

#### 7.2.1 Emacs で文字列検索、文字列置換する

`emacs` で文字列を検索する方法をおさらいしておきましょう。前方検索 (`C-s`)、後方検索 (`C-r`) があります。`emacs` で `ConstitutionOfJapan.txt` を開き、`peace` を検索してみましょう。`emacs` で文字列を置換する方法をおさらいしておきましょう。

`replace-string(M-x replace-string)`, `replace-regexp(M-x replace-regexp)` が利用できます。`emacs` で `ConstitutionOfAizu.txt` を開き、`Aizu` を `Japan` に置換してみましょう。

<sup>1</sup>`strings` は GNU の binutils バッケージで配布されています。

## 7.2.2 単語リストを作る

英語の単語は空白で区切られています。従って空白文字の連なりを改行文字に置換すると各行に一単語ずつがOutputされます。

space2newline

```
s/_.**/\n/g
```

この様に2行に書かれた sed のスクリプトファイルを（ここでは、ファイルの名前を space2newline とします）利用して

```
sed -f space2newline ConstitutionOfJapan.txt
```

を実行すると単語に分解することができます。スクリプトファイルは、emacs などで作成します。1行目のその後には何もいれません。また、2行目は必ず改行しておきます。 sed では改行を replacement 文字列中に一行で表示できないのでこの様に2行になります。さらにこの結果を、sort を利用して並び替えると重複した行は1行にまとめられ、アルファベット順に並んだ単語のリストが出来上がりです。

```
sort -u <space2newline_ConstitutionOfJapan.txt | sort -u
```

sort -u を利用して得られたリストは出現頻度の情報が欠落します。

## 7.2.3 tr を用いて置換する

空白を改行に置換するのとは少し意味が違いますが、/usr/bin/tr を使用すると単語を簡単に抽出できます。

/usr/bin/tr -c "[alpha:]" "[\n\*]" <ConstitutionOfJapan.txt

/usr/bin/tr を使用すると Upper case を Lower case に容易に置換できます。

/usr/bin/tr -[upper:] -[lower:] "

これは単語を Upper case と Lower case で区別したくない場合には便利です。こちらは、-c オプションはつけません。

## 7.2.4 他のツールで文字列置換する

sed 以外にも文字列置換の機能を持っているツールがあります。perl で空白文字の連なりを改行に置換する例です。

```
perl -ne 's/_.**/\n/g; print; ' <ConstitutionOfJapan.txt
```

perl では置換コマンド s の replacement 文字列中に \n を利用できます。

perl では lc 関数を用いて大文字を小文字にすることも簡単にできます。

perl -ne 'print lc(\$\_);' <ConstitutionOfJapan.txt

## 7.3 課題 7

ConstitutionOfJapan.txt の単語出現頻度リストを作りなさい。単語出現頻度リストを作る際に行った操作を全てファイル \$HOME/literacy/words\_of\_ConstitutionOfJapan.txt にレポートせよ。以下はヒントです。

- 文書を1行1単語のデータに変換する。（=文書から単語を抽出する。）
  - A-Z,a-z 以外の数字や記号などの文字を削除する。
    - (sed や perl を利用して s/[^\w]/ /g を実行するか、あるいは tr を利用する。)
  - Upper case を Lower case に置換する。
    - (sed は L を使う。perl は lc。tr でも良い。)
  - アルファベット順に並び替える。
  - 出現頻度を数える。
    - 単語リストファイルに対して以下の awk のプログラム wordcount.awk を利用するか、あるいはソートされた単語リストファイルに対して以下の awk のプログラム wordcount2.awk を利用して

awk -f wordcount.awk 単語リストファイル

を実行すると良い。

wordcount.awk

```
{
 count[$1]++;
}
END {
 for(i in count){
 print i, count[i];
 }
}
```

wordcount2.awk

```
BEGIN {
 word="";
}
{
 if($1==word) {
 print word, count;
 word=$1;
 count=1;
 } else {
 count++;
 }
}
END {
 print word, count;
}
```

## 第8章 表計算

8.1.2 Awk を使った簡単家計簿  
家計簿を作りましょう。Emacs などを用いて家計簿のファイル kakeibo を書きます。

```
利用方法を kakeibo ファイルの中にコメントとして書いておくと便利ですね。
grep -v # kakeibo | awk 'BEGIN{FS=""}{sum+=$3;print $1":"$2":"$3"\t:$sum}'
4/15:銀行から :1000000
やつと生活ができる。
:学食 :-490
:教科書 :-1200
:コーヒー :-450
#:コーヒーは美味しいですね。
4/16:学食 :-440
:ノート :-200
:コーヒー :-450
:家賃 :-500000
#:高い!
```

表計算とは表に書かれた数値を縦横に集計したり、平均値や分散を求めたりする計算です。

### 8.1.1 Awk を使った簡単な足し算

Awk<sup>1</sup> (=オーケーク。GNU-Project 製のGawk<sup>2</sup>も利用できます。Gawk はゴーケークとは呼びません。) は表計算を比較的容易に行なうことのできる道具(ツール)であり、プログラム言語です。

```
cat | awk 'BEGIN{sum1=0;sum2=0}{sum1+=$1;sum2+=$1^2}END{print sum1,sum2}'
```

をターミナルで実行して

```
1
2
3 4
5
C-d
```

をタイプしてみましょう。入力したデータの1列目の数値の合計(sum1)と数値の2乗の合計(sum2)を表示します。Awk のプログラムは3つの部分から構成されます。

```
前処理 — BEGIN{sum1=0;sum2=0}
各行に対する処理 — {sum1+=$1;sum2+=$1^2}
最後に後処理 — END{print sum1,sum2}
```

です。前処理では変数 sum1, sum2 の初期値を 0 にセットしています。(sum1=0;sum2=0) 各行に対する処理では一行ずつデータを読み込んで第一番目のフィールドにある値を sum1 に追加しています。また同時に sum2 に数値の2乗を追加。(sum1+=\$1;sum2+=\$1^2) 後処理で sum1 と sum2 の値を出力しています。(print sum1,sum2)

Awk は整形されたテキストファイルのデータの表計算を非常に楽に行なうことができます。Awk では予約変数が幾つかあります。各行のフィールド数を表す NR=Number of Fields や処理中の行数 NR=(Number of Records) は比較的利用する機会が多いでしょう。

この例の様にファイル kakeibo の中に書いてあるとしましょう。集計の計算をするためには grep -v # kakeibo | awk 'BEGIN{FS=""}{sum+=\$3;print \$1":"\$2":"\$3"\t:\$sum}' を実行すると良いでしょう。この kakeibo のファイルの様にデータの他に処理の方法を記入しておくると操作の方法を忘れて困ることができます。

この awk プログラムでは前処理の部分で FS=":" を実行しデータの区切り文字(Field Separator)=FS を文字: にセットしています。FS がセットされていなければデータは空白やタブで区切られます。

この様に awk を利用すればテキストデータについて表計算処理を実行することができます。表計算とは表に記入された数値を処理する計算の呼び名です。awk を利用することができます。あれば他の高価な表計算ソフトウェア(例えば MS 社の Excel 等)は必要ないでしょう。カンマ(,) を区切り文字としたテキスト形式のファイルを CSV 形式のファイルと呼びます。CSV とは Comma Separated Values の略です。CSV 形式は伝統的な形式であって、Excel や OpenOffice でも対応しています。

```
grep -v # kakeibo
```

は kakeibo 中の # を含む行を落とす操作です。Awk とは直接関係ありません。このやり方以外に、特定の文字列を含む行に対して処理を行うために、Awk では

```
$N~/pattern/{第 N フィールドの文字列が pattern にマッチした場合に実行する処理}
$N!~/pattern/{第 N フィールドの文字列が pattern にマッチしなかった場合に実行する処理}

の様な便利な機能があります。ただし、NR の値はあくまでも入力されるデータの行数なので、grep 等を使用してあらかじめ不要な行を落とした場合とでは NR の値が異なります。
```

```
grep -v # kakeibo| awk -F: '{print NR}'
awk -F: '$1!~/#/ {print NR}', kakeibo
```

の出力を比較してみましょう。

<sup>1</sup>See 'man awk': pattern scanning and processing language

<sup>2</sup>Also see 'man gawk': pattern scanning and processing language

### 8.1.3 Awk を使った簡単成績処理

成績ファイルを作りましょう。Emacs 等を用いて成績のファイル score を書きます。

```
利用方法を score ファイルの中にコメントとして書いておくと便利ですね。
grep -v # score | awk -F: '{print $1":"$2":"$3":"$4"\t"$2+$3+$4}'
#
名 前:英語:数学:国語
会津太郎:100:70:80
鶴賀丈次:60:40:50
猪苗代子:60:80:100
東山温子:90:20:70
芦野牧男:50:30:80
```

のようにファイル score の中にデータが書いてあるとしましょう。それぞれの学生の総合点を計算したい人は

```
grep -v # score|awk -F: '{print $1":'$2':'$3':'$4':\$2+\$3+\$4}'
```

を実行すると良いでしょう。

この awk プログラムでは awk の起動オプション -F: によってデータの区切り (Field Separator)=FS を文字 : にセットしています。

### 8.2 課題8

成績ファイル score 中の英語の平均点を計算する awk スクリプトを作りなさい。  
kadai.awk というファイルに1行目にユーザ名、2行目に名前、3行目に awk スクリプト、4行目に得られた結果のみを記入すること。

- `$(NF-1)` で右から一つ前のフィールドの値を参照できる。

8.3 表の連結

paste コマンドや join コマンドを用いると、表の連結することができます。

- paste
- 指定された複数のファイルの各行を TAB で区切りながら連結して一つの行にします。標準出力に出力されますので、リダイレクトを用いてファイルに格納したり、パイプを用いてさらに他のコマンドなどで別の処理をします。区切り文字は、-d オプションで指定して変更できます。利用可能なオプションの詳細は `paste -help` で確認できます。
- join
- 指定された2つのファイルの各行から、指定した共通のjoinフィールド(デフォルトは先頭)を持つ行を連結して、標準出力に書き出します。2つのファイルはjoinフィールドについてあらかじめソートされなければならない。利用可能なオプションの詳細は `join -help` で確認できます。

paste の例

```
% cat 123.txt
1 2 3
4 5 6
7 8 9
% paste 123.txt 123.txt
1 2 3 1 2 3
4 5 6 4 5 6
7 8 9 7 8 9
```

join の例

```
% cat abc.txt
a b c d
e f g h
i j k l
% cat abc2.txt
a 1 2 d
b 3 4 h
e 5 6 1
i 7 8 k
% join abc.txt abc2.txt
a b c d 1 2 d
e f g h 5 6 1
i j k 1 7 8 k
```

8.4 宿題8

/home/course/lit1/pub/literacy/ に用意してあるスコアファイル math.csv, english.csv, physics.csv を awk を用いて集計して以下を求めよ。

- 各学生に対する3科目の合計点リスト。

それぞれを求める手順と標本平均と標本分散についての結果をファイル名 shukudai.awk にレポートせよ。各レコードに複数のスコアがあるものは、追試の結果である。追試の結果を利用するように工夫せよ。以下のヒントを活用しても良い。

- 合計点の標本平均と標本分散。
- `$(NF-1)` で右から一つ前のフィールドの値を参照できる。
- paste コマンドを利用する複数のファイルの各行を組み合わせることができる。

$$\text{偏差} = Y \text{ の不偏分散 } V(Y) = \frac{1}{M-1} \left( \sum_{i=1}^M (y_i - \bar{Y})^2 \right)^{1/2}$$

## 第9章 情報検索

### 9.1 情報の入手

会津大学では、ほんどの情報を計算機を利用して取得できます。以下に項目リストを紹介します。

- NetNews  
ネットニュースを見る習慣をつけて、良く見るようにしましょう。さもなくば重要な情報を知ること無く、損をする可能性があります。他のメディアでは連絡が無い場合もあるので、毎週必ず、確認しましょう。以下の3つは特に重要です。
  - 情報センターからのアナウンス(ニュース u-aizu.istc)
  - 学生部からのアナウンス(ニュース u-aizu.announce.gakuseibu)
  - その他のアナウンス(ニュース u-aizu.announce)
- World Wide Web
  - 図書検索  
<http://libeopsv.u-aizu.ac.jp/top/index.do?method=change&langMode=JPN>
  - 授業の履修(学務システム)  
<http://www.u-aizu.ac.jp/current/administration-system/>  
学生課からの連絡などが学務システムのフォーラムで参照できます。
  - 情報センター  
<http://web-int.u-aizu.ac.jp/labs/istc/ipc/>
  - 情報センター:よく聞かれる質問とその答(FAQ)  
<http://web-int.u-aizu.ac.jp/labs/istc/ ipc/faq/faq.html>
  - 授業時間割り  
<http://www.u-aizu.ac.jp/current/classschedule/>
  - サークル活動のページ  
<http://www.u-aizu.ac.jp/campus/circles/>
  - 卒業論文について  
<http://www.u-aizu.ac.jp/curriculum/thesis/>  
卒業論文の書き方、研究室配属の情報があります。

以上が主な情報ページです。その他の情報が大学のホームページ(<http://www.u-aizu.ac.jp/>)に登録されているので利用して下さい。

- 教員が提供する情報、教員が授業でいろいろな手段で情報を与えることがあります。

### 9.1.1 情報検索

• WWW を使って情報検索をすると役に立つ場合があります。ここでは Google を例に紹介します。



検索では検索オプションを適切に使うことで情報の SN 比 (Signal Noise) を高くすることができます。  
基本的な検索のヘルプ  
<http://support.google.com/websearch/bin/answer.py?hl=ja&answer=134479>  
演算子と他の検索ヘルプ  
<http://support.google.com/websearch/bin/answer.py?hl=ja&answer=136861>  
などを参考にしましょう。

- すべてのキーワードを含む(基本検索、単語の順により検索結果が異なる)  
"I love you so much"
- フレーズを含む(フレーズ検索)。ダブルクォーテーションで囲む。  
"I love you so much."
- キーワードを含めない(マイナス検索)  
I -love you so much

フレーズ検索を利用すれば、「ヒット数が多い場合は表現が標準的であり無難である。ヒット数が極端に少い場合は表現が稀であり通用しない。」と判断できます。

### 9.2 文字列検索

文字数の多い文書から必要な情報を取得するために、文字列検索をすると便利な場合があります。もちろん、文書全体を読むことが大切なのは当然ですが、時間が限られている場

- e-mail
- ニュース(u-aizu.course.XXX)
- ファイルの公開  
</home/course/lit1/pub/literacy/>にはコンピュータリテラシーの授業のプリント等の資料があり、公開されています。

合などはキーワードを絞って部分を読む必要がで出来ます。このような場合に文字列検索が役に立ちます。

Grepは強力な文字列検索機能を提供しますが、シェル環境内でテキスト情報のみに適用できます。他のアプリケーションでは文字列を検索するために、Grepそのものは利用できません。

情報を閲覧するアプリケーションでは其々において文字列検索の機能が提供されています。

- Firefox の場合は **Edit** メニューの **Find** で利用できます。
- Acrobat Reader の場合は Toolbar 内に word 検索欄があります。また、**Edit** メニュー内には **Find** に加えて **Search** もあります。
- OpenOffice (soffice で起動) には **Edit** メニュー内に **Find & Replace ...** があります。
- Mac OS X 端末で利用できる MS-Office では **Edit** メニュー内に **Find** や **Replace** があります。

もしも LaTeX の dvi ファイルだけが渡されて文字列を検索する必要がある場合にはどうすれば良いでしょうか？dvi ファイルの Viewer では検索機能が無いかも知れません。こんな場合は pdf に変換すれば良いでしょう。文字端末での dvi Viewer である dvit2tty というツールを使用すれば dvi ファイルに含まれている文字情報をテキストに変換できます。

```
dvi2tty sample.dvi > sample.txt
```

こうして、文字情報をテキストファイルに記録すれば、Grep を用いて検索できます。

### 9.2.1 課題 9

以下を調べて word-search に保存せよ。

1. "I give you it" と "I give it to you" を検索しヒット数を比較せよ。
2. "make me up" "let me up" "get me up" "take me up" "wake me up" "raise me up" それぞれの意味とヒット数を調べよ。他に "xxx me up" を 2つ考えて意味とヒット数を調べよ。
3. "higgs particle" を検索しヒット数を調査し、ヒット数が少い理由を述べよ。

## 第10章 ネットワークのエチケットや倫理

### 10.1 Network Etiquette = Netiquette

ネットワークはとても便利です。海外への電子メールは数分の単位で届きます。これは大変な事実です。人類は沿途の量と速度で情報を共有できる時代に突入しようとしています。これは今までの制度や国境を越えて、初めて地球レベルでの文化の共有が実現されるということを意味します。

ネットワークを利用する場合に守るべきことがらがあります。それは、法律や条例で定められているものもあれば、明文化されないまでも、社会の常識としてある程度の幅を持ちながら理解されているものもあります。

以下に列挙する事例について正しいと判断するものには○、間違っていると判断するものには×を付けなさい。間違いである場合にはその理由と訂正を記入すること。

- [ ] 11. まだインターネットの利用に関しては法の整備が遅れているので何をやっても許される。
- [ ] 12. インターネットの利用によって誰かに迷惑をかけたとしても見知らぬ他人のことだから気にしないでも良い。
- [ ] 13. ネットワークは実生活とは関係がなく仮想的な空間なので、普段と違って、自分勝手に好き放題をしても良い。
- [ ] 14. 会津大学では、多くの計算機にある様々なファイルを共有することによって知的情報をシェアしあっている。
- [ ] 15. 会津大学では、多くのユーザが登録されて利用しており、各ホームディレクトリにはプライベートな情報や大学の管理内容に関係する機密性の高い情報もある。
- [ ] 16. どの様なコンピュータでも一般的にファイルの内容はユーザ、グループ、その他のそれぞれに対するアクセス(読みだし、書き込み、実行)の許可(=permission)、不許可によって制御できる。
- [ ] 17. UNIX や UNIX をベースしたコンピュータでは、一般的にファイルの内容はユーザ、グループ、その他のそれぞれに対するアクセス(読みだし、書き込み、実行)の許可(=permission)、不許可によって制御できる。
- [ ] 18. 自分の情報を適正に管理するにはアクセスパニックションに注意する必要がある。
- [ ] 19. 他人のファイルのパニックションがアクセス可能になっている場合は無断で内容を取得したり、利用する事が可能である。

[ ] 10. 他人のファイルのパニックションがアクセス可能になっている場合は無断で内容を取得したり、利用することが倫理的に許されることを意味している。

[ ] 11. 会津大学では学生は目の前にあるワークステーションにログインすれば、そこからの演習室のどのホストにもリモートでログインでき、計算機資源を利用できる。

[ ] 12. パスワードが見破られ易いとの警告を管理者から受けたが新しいパスワードを覚えるのが面倒なので変更しない。

[ ] 13. Emacs で電子メールを読めば問題がないので一行に何文字でもだらだらと文章を続けるのも良い。

[ ] 14. Emacs で電子メールを読めば問題なく読むことができるが、適切に改行を入力するなどのメールの受信者に対する配慮をすることが望ましい。

[ ] 15. サブジェクトに漢字コードを書くと日本人同士は理解し易いので良いと思われるが、メールやニュースのヘッダ部分に2バイトコード(漢字や半角カタカナ)が含まれているとネットワーク上のトラブルの原因になるので避けるべきである。

[ ] 16. サブジェクトに漢字コードを書くと日本人同士は理解し易いので良いと思われるが、メールやニュースのヘッダ部分に2バイトコード(漢字や半角カタカナ)が含まれているとネットワーク上のトラブルの原因になるので避けるべきである。

[ ] 17. メールを受け取ったが変な文字ばかりであった。いやがらせに違いない。

[ ] 18. とても複雑で美しいシグネチャを受け取った人は皆喜ぶ。

[ ] 19. 大事なメールを送る時には、保存、記録、確認の目的で自分宛に Carbon Copy をしておくと役に立つ場合がある。

[ ] 20. メールアドレスがはつきりしなかつたけれども郵政公社の郵便局と同じで相手の管理者が適当に配慮してくれるだろう。

[ ] 21. 海外の計算機(FTP サイトや World Wide Web サイト)にアクセスしたが転送に時間がかかる。回線が細いが混みあっているに違いない。近くに同様の情報を保持している強力なサイト(ミラーサイト)があるかもしれないで探してみる。

[ ] 22. anonymous(匿名)FTP の利用の際に尋ねられるパスワードには適当なニックネームを入力する。

[ ] 23. anonymous FTP は匿名利用なので利用者 id として ftp や anonymous を使用することができます。パスワードを尋ねられた場合には利用者のメールアドレスを入力する。

[ ] 24. anonymous FTP でパスワードを尋ねられて正直にユーザアカウントのパスワードを入力してしまった。

[ ] 25. ネットワークは仮想的な空間なので匿名でやりとりをするべきだ。

[ ] 26. 他人の振りをしたメールや投稿をしてはいけない。

[ ] 27. 通常の手紙と同様に、礼儀正しく書く。

[ ] 28. e-mail やネットニュースでは自動的に From: がつくので、文章の中では自分の名前を名乗らなくても良い。

- [ ] 29. e-mail やネットニュースでは自動的にFrom: 行が追加される場合が一般的である。それだけでは発信元を十分に明かにしたとは言えないのに本文の中やシグネチャで自分の名前を名乗ることが望ましい。
- [ ] 30. From: 行を書き換えれば、誰だから分からず、どんな記事(いたずら、中傷)をメールやニュースに書いてもばれてしまうことは無い。
- [ ] 31. インターネットは無料で使用できるのだから不特定多数宛へのダイレクトメールをしても良い。
- [ ] 32. チェーンメールはしない。
- [ ] 33. 核実験反対の意見メールを受け取った人は、次に5人の人に同様の内容でメールを送れば、世の中が平和になるし、おまけに誰の迷惑にもならない。
- [ ] 34. 友人から受け取ったメールの内容を無断でニュースに投稿した。
- [ ] 35. e-mail で受け取った文章を相手の了解を得ないで公開してもよい。
- [ ] 36. プライベートなやりとりのみのニュースへの投稿は控える。
- [ ] 37. ニュースやメールで誤のわからぬことを言う連中には乱暴な言葉使いで対応すれば十分だ。
- [ ] 38. ニュースやメールに対し興奮した状態で書いた文章は1日程度頭を冷やしても一度再読してから投稿すると良い。
- [ ] 39. ニュースやメールで自分自身の失言を認めたら素直に謝罪する。
- [ ] 40. また聞きの話を利用しない。
- [ ] 41. 引用をする場合は引用元を明らかにする。
- [ ] 42. ニュースグループの名前から判断したりリスト全部に適当に同じ内容の投稿をする。
- [ ] 43. 悪者はニュースでとっちめるべき。
- [ ] 44. 悪者は許してはいけない、ネットニュースでその悪事を追求することが自分の使命である。
- [ ] 45. 無闇やたらに挙げ足を取るような態度は控える。
- [ ] 46. 挙げ足を取ることは言葉遊びの要素もあるので、知的私にはちょっぴり楽しい、だけど無闇やたらに挙げ足を取るような態度を続けていると人格を疑われるし、最近友達も減ったようだ、反省して今後は挙げ足を取ることは控えよう。
- [ ] 47. ニュースでは議論している相手だけが対象ではなく、多くの良識のある読者がいるものだと割り切るなど、気持ちの整理をする。
- [ ] 48. 質問に対する返事のお礼をする場合には、簡潔に「ありがとうございました」の一言をニュースに投稿すれば良い。
- [ ] 49. 質問に対する返事のお礼をする場合には、簡潔に「ありがとうございました」の一行のみをニュースに投稿すれば良い。
- [ ] 50. アカウントを持っていないサイトに無闇にtelnet やrloginなどを利用してアクセスしてはいけない。
- [ ] 51. 端末から長時間離れる時はlogout する。
- [ ] 52. 端末から長時間離れる時はxlock をかけておかば良いだろう。
- [ ] 53. 演習室の端末から1時間程度離れる時はxlock をかけておかば良いだろう。
- [ ] 54. 他人の端末をいたずらしたり、覗きこんだりしても良い。
- [ ] 55. 管理者から指定されたディスクの使用量(1GByte)を守るように気を配る。
- [ ] 56. 尊敬する両親にはパスワードを報告している。
- [ ] 57. アカウントを共有すると便利である。
- [ ] 58. サークルでユーザーアカウントを共有すると便利なのでそうするべきだ。
- [ ] 59. パスワードをタイプしている手元や口元、モニタを覗くと時々わかつてしまがある。
- [ ] 60. パスワードをタイプしている手元や口元、モニタを覗くと時々わかつてしまことがあるが、その様な態度はエチケットに反する。
- [ ] 61. 会津大学のパスワードの有効期限は90日だが、有効期限後もパスワードを変更しない。
- [ ] 62. 長時間重いプロセスを実行する場合にはnice をかけて、cpu の使用優先度を押えて、他のユーザに迷惑をかけない様に気を配る。
- [ ] 63. 秘密にしたいファイルにはアクセス許可を出さない。
- [ ] 64. アクセス許可があるファイルでも共有を前提としない他のファイルは覗かない。見たい場合は了承を得る。
- [ ] 65. 画像データなどのファイルはサイズが大きいのでホームディレクトリには置かずUSB メモリやremovable HDD 等のメディアに移すと良い。
- [ ] 66. 個人で作成した貴重なファイルは個人で保管するためにUSB メモリやremovable HDD 等のメディアに保存すると良い。
- [ ] 67. データのバックアップは基本的に個人の責任で行うべきである。
- [ ] 68. せっかく、会津大学の演習室は24時間利用できるのに、夜間や日曜日などに建物の中に入るためにカード読み取り器が設置されているドアに制限されるので不便このうえない。ドアを工夫して解放しておけば便利だと思ったので、そうしておいたら、どうやらどこかで盗難騒ぎが発生したようだ。ドアを開放していたのは全く悪気が無かったし、泥棒が悪いのだから自分には少しも責任がない。

[ 169. 会津大学は会津若松市にあって、計算機を盗む泥棒などいのだから、講義棟や研究棟のドアは全て解放しておけば良い。

[ 170. 公共物を利用する場合に公共物の保全や他の利用者の迷惑を避けるために、遡けることが困難な不便さが存在する場合がある。その様な不便さは管理者が解決するべき問題なので利用者が公共物の保全をするために協力的な態度をとる必要は全く無い。

[ 171. ネットワークやコンピュータのシステム管理者はユーザのパスワードを知っている。

[ 172. ネットワークやコンピュータのシステム管理者と名乗る者に自分のパスワードを教えてはならない。

[ 173. 情報センターはフリーメールアドレスからの問い合わせに対応しない。

[ 174. リテラシーについての問い合わせは正規のメールアドレス(sXXXXXXX@u-aizu.ac.jp)からのものについてのみ対応する。

## 10.2 課題 10

Network Etiquette について以下の問い合わせの中から選んでファイル netiquette 内に回答せよ。

1. Network Etiquette に関して質問を述べよ。
2. ハンドアウトに列挙した"Network Etiquette に関する問い合わせ"に更に追加すべき問い合わせを述べよ。

## 10.3 宿題 10

WWW はとても良くできたシステムで容易にいろいろな情報にアクセスできます。会津大学の学生も、いろいろな目的で個人のホームページを公開しています。

- 会津大学の教員のホームページ
  - 自分以外の会津大学の学生のホームページ
  - 他の大学の学生のホームページ
  - その他のホームページ
  - NetNews への投稿記事
- 記事の Message-ID を忘れずに記載してください。記載洩れの場合は減点します。

のいづれかを觀察し、著作権や、マナー、アピール度やオリジナリティーについて感想をレポートしなさい。以下に用意するテンプレートに従って、net-report.tex に保存しなさい。

```
\documentclass[a4paper]{jarticle}
\title{A report of a Student's Web page}
\author{sXXXXXX}
\date{\today}
\begin{document}
\maketitle
\begin{itemize}
\item \textbf{掲載タイトル}\\
% この上の行に(ページの URL とタイトル)あるいは

% (投稿のニュースグループと Message-ID, サブジェクト)を挿入して下さい。
\item \textbf{掲載内容の要約}\\
% この上の行にページ/投稿の内容を簡単に紹介した文章を挿入して下さい。
\end{itemize}
\end{document}
```

## 第 11 章 $\text{\LaTeX}$ の基礎

### 11.2.1 英文のみの文書

英文のみの文書の場合は book, article, report, letter 等のクラスを使用します。またコンパイルは `pdflatex` ではなく `latex` コマンドを実行します。

### 11.1 $\text{\LaTeX}$ の文章の最小構成

$\text{\LaTeX}$  の文章は次のように構成になっています。

```
\documentclass[a4paper]{article}
\begin{document}
本文
\end{document}
```

`a4paper` にある部分には各種スタイルを指定できます。`jarticle` にある部分には各種クラスを指定できます。クラスは他に `jbook`, `tbook`, `tarticle`, `jreport`, `treport` 等があります。

### 11.2 $\text{\LaTeX}$ のコンパイル

$\text{\LaTeX}$  の場合、`tex` 形式で書いたテキストファイル==ソースファイルを元にして `dvi` 形式と呼ばれる画像ファイルに変換することをコンパイルすると呼ぶことがあります。

`pdflatex` のソースファイル

の様に端末プログラム (gnome-terminal 等) の中で実行します。`tex` 形式の間違いがあるとコンパイルは停止し、その後の指示を待ちます。

```
% pdflatex test
This is pTeX, Version 3.1415926-p3.1.11 (euc) (TeX Live 2009)
restricted \write18 enabled.
```

```
ltxmp/test.tex
platex2e <2006/11/10>+0 (based on Latex2e <2009/09/24> patch level 0)
Broken Pipe
Document Class: jarticle 2006/06/27 v1.6 Standard plTeX class
Broken Pipe
(/usr/local/texlive-2009/p2009/texmf/ptex/platex/base/jarticle.cls
-
```

```
! LaTeX Error: Environment document undefined.

See the LaTeX manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
...
1.2 \begin{document}
```

### 11.3 $\text{\LaTeX}$ (1)

#### 11.3.1 $\text{\LaTeX}$ の注意

ascii 文字の {と、尖角の }とを混同している人が見られました。これらのいくつかの特殊文字と、英数字は、ascii 文字と JIS コード文字の 2 種類が存在します。この 2 つはみかけは似ていますが、コンピュータ内部では全く違う文字として処理されます。これらの文字を入力する時は、原則として ascii 文字を使用するようにしてください。

#### 11.3.2 特殊文字

文章には英数字だけでなく、様々な文字が使われますが、., ; ! ? () [ ] ‘ ’ - + \* / @ はそのまま入力できます。

表 11.1. 特殊文字 1

| 特殊文字 | 役割          | 印字法                                                     |
|------|-------------|---------------------------------------------------------|
| {    | グループの始まり    | \{                                                      |
| }    | グループの終り     | \}                                                      |
| \$   | 数式モードの始めと終り | \\$                                                     |
| &    | タブ文字        | \&                                                      |
| #    | マクロの引数      | \#                                                      |
| %    | コメント行の作成    | \%                                                      |
| _    | コマント行の作成    | \%                                                      |
| ~    | 文字空白        | \verb+~+                                                |
| ^    | 上付き添え字の記号   | \textasciitilde (L <sup>E</sup> T <sub>E</sub> X2e で有効) |
| `    | 下付き添え字の記号   | \verb+`+                                                |
| -    | コマンドの先頭文字   | \verb+`+                                                |
| /    | コマンドの先頭文字   | \textbackslash (L <sup>E</sup> T <sub>E</sub> X2e で有効)  |

表 11.2. 特殊文字 2

| 特殊文字    | 印字法   | 特殊文字 | 印字法        | 特殊文字    | 印字法  |
|---------|-------|------|------------|---------|------|
| \$      | \\$   | ©    | \P         | †       | \dag |
| ¥       | \ddag | ®    | \copyright | \pounds | \OE  |
| ¥\llap= | \ae   | Œ    | \oe        | \aa     | \AA  |
| æ       | \AA   | œ    | \AE        | \o      | \o   |
| À       | \L    | ø    | \O         | \ø      | \O   |
| à       | \l    | ł    | \l         | \i      | \i   |
| ł       | \j    | ł    | \L         | ł       | \i   |
| ł       | ł     | ł    | ł          | ł       | \i   |
| ì       | ì     | ì    | ì          | ì       | \i   |
| ì       | ì     | ì    | ì          | ì       | \i   |
| ì       | ì     | ì    | ì          | ì       | \i   |
| ì       | ì     | ì    | ì          | ì       | \i   |
| J       | ì     | ì    | ì          | ì       | \i   |
| ì       | ì     | ì    | ì          | ì       | \i   |
| ì       | ì     | ì    | ì          | ì       | \i   |
| -       | -     | -    | -          | *       | *    |
| TEX     | \TeX  | TEX  | \LaTeX     | --      | --   |

しかしながら、以下に示す文字は特殊な用途に使われているため、その文字がタイプライタのキーに存在していても、その文字を入力しただけでは印刷できません。

{ } \$ & # % ~ ^ \_ \

それらの記号を出力するには、表 11.1 に示した印字法を用いて指定しなければなりません。表 11.1 に示した以外の特殊文字を表 11.2 に示します。

### 11.3.2 アクセント記号

LATEX ではドイツ語、フランス語、スペイン語、ロシア語などの文字を簡単に出力できます。表 11.3 に示したアクセント記号を使うことができます。

表 11.3. アクセント記号

| 名前     | 記号 | 印字法   | 名前        | 記号 | 印字法     |
|--------|----|-------|-----------|----|---------|
| grave  | à  | \`a   | check     | š  | \v{s}   |
| acute  | é  | \^e   | breve     | í  | \u{u}   |
| hat    | ô  | \^o   | long      | ѓ  | \H{j}   |
| umlaut | ü  | \u{u} | tie-after | ѝ  | \t{i} u |
| tilde  | ñ  | \~n   | cedilla   | ç  | \c{c}   |
| bar    | ÿ  | \=y   | bar-under | ќ  | \b{k}   |
| dot    | þ  | \.p   | dot-under | ѝ  | \d{h}   |

### 11.3.3 字体 (フォント = font) と文字の大きさ

文章中で字体 (文字の種類、文字の大きさ) を変えるには付録に出ている *latex2.tex* (D.2, 191 ページ) を参照して下さい。下線や囲みで文字列を強調する方法も *latex2.tex* を参照して下さい。

### 11.3.4 ソースをそのまま出力する方法

\begin{verbatim} と \end{verbatim} の二つの行で囲んだ部分は、どの文字でもソースがそのまま出力されます。例えば

\begin{verbatim}\begin{verbatim}特殊文字 \$ もそのまま出力できます.\end{verbatim}\end{verbatim}

と指定すると

出力

特殊文字 \$ もそのまま出力できます。

となります。  
 \begin{verbatim} と \end{verbatim} のように、ペアになった命令のことを環境といいます。たとえば

\begin{verbatim}\begin{verbatim} ... \end{verbatim}\end{verbatim}

\begin{verbatim} と "verbatim" 環境といいます。

### 11.3.5 左寄せ・右寄せ・センタリング

ソース

```
\begin{flushleft}左寄せ\end{flushleft}\begin{flushright}右寄せ\end{flushright}\begin{center}真中寄せ\end{center}
```

この様になります。

出力

左寄せ

真中寄せ

右寄せ

### 11.4 数式モード, 参照, 図の挿入

以下に続く小節は LATEX の数式モード, 参照, 図の挿入の機能の入力サンプルとそれに対応する出力サンプルです。



## 11.4.1 数式を書く場合の注意

数式は数式モードに入ると手軽に書くことができます。数式モードに入る方法は、主に  $x^2$  の様に \$と\$で囲む方法、

$$y = x^2 + 1$$

の様に \$\$と\$\$\$で囲む方法、

$$y = x^2 + 1$$

の様に equation 環境を使う方法、そして

$$y = x^2 + 1 \quad (11.2)$$

$$= 2^2 + 1 \quad (x = 2 を代入)$$

$$= 5 \quad (11.3)$$

の様に eqnarray 環境を使う方法（これは (11.1) 式に  $x = 2$  を代入した結果です）等があります。

## 11.4.2 Examples

Index

$$a_{ij} b^{jk} = c_i^k$$

Fractional number

$$\frac{x^2 - y^2}{x + y} = x - y$$

Vector

$$\vec{v} = (1, 0), \quad \vec{w} = (0, 1), \quad \vec{v} \cdot \vec{w} = 0 \quad \overrightarrow{AB} + \overrightarrow{BC} = \overrightarrow{AC}$$

Sum, Integral and limit

$$\lim_{N \rightarrow \infty} \sum_{i=1}^N f(x_i)(x_{i+1} - x_i) = \int_{x_0}^{x_\infty} f(x) dx$$

Greek

$$\lim_{\epsilon \rightarrow 0} \frac{f(x+\epsilon) - f(x)}{\epsilon} = \frac{d}{dx} f(x)$$

放射線にはその実体として、原子核、電子、高エネルギー光子があり、それぞれ  $\alpha$  線、 $\beta$  線、 $\gamma$  線と呼ばれている。

Square root

$$\sqrt{x^2 + 2x + 1} = |x + 1|$$

Integration

$$\left( \int_{-\infty}^{\infty} dx e^{-x^2} \right)^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} dx dy e^{-(x^2+y^2)} = \int_0^{2\pi} \int_0^{\infty} d\theta r dr e^{-r^2} = \pi \quad (11.5)$$

Matrix

$$\left( \begin{array}{c|c} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \\ \hline & & c \end{array} \right)$$

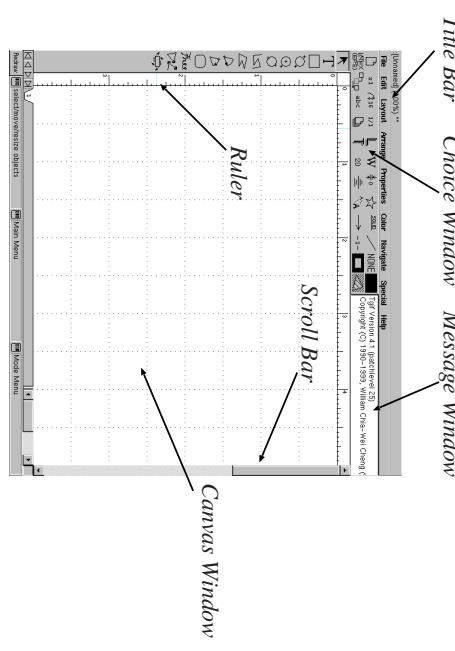


図 11.1. サブウインドウの名前

画像の一部を切り出して表示することもできます。



図 11.2. サブウインドウの名前

図 11.1 は TeX の説明です。文献 [1] や [2] は LATEX2e の前身である LATEX についての参考書です。他にも LATEX や TeX の参考文献は図書館にありますので、利用してみましょう。

## 関連図書

- [1] LaTeX 自由自在, 磯崎秀樹, サイエンス社
- [2] 文書処理システム L<sup>A</sup>T<sub>E</sub>X Leslie Lamport, アスキー出版局
- [3] 日本語 L<sup>A</sup>T<sub>E</sub>X2e ブック, 中野賢アスキー出版局
- [4] L<sup>A</sup>T<sub>E</sub>X2e 美文書作成入門 改訂第3版, 奥村晴彦, 技術評論社
- [5] The TeXbook, Donald E. Knuth, Addison-Wesley

## 11.5 課題 11

サンプルを出力と照らし合わせながら確認しなさい。特に以下の表の項目について確認し、メモを完成させなさい。メモは latex-memo というテキストファイルに記録せよ。

| 項目              | 使用目的          |
|-----------------|---------------|
| figure          | 環境<br>絵を貼り込む。 |
| includegraphics |               |
| equation        | 環境            |
| eqnarray        | 環境            |
| array           | 環境            |
| center          | 環境            |
| label           |               |
| ref             |               |
| thebibliography | 環境            |
| cite            |               |
| verbatim        | 環境            |

## 第 12 章 L<sup>A</sup>T<sub>E</sub>X の練習

話を地球上に持つてみると、 $M$  を半径  $R$  の大きさの地球とし  $m$  を地表から高さ  $h$  の  
宇宙にあるりんごとしよう。そうすると  $r = h + R$  が成り立つので、

$$\frac{d^2}{dt^2}(h + R) = -G \frac{M}{(h + R)^2}$$

となる。 $R$  は変化しない量なので、

$$\text{左辺} = \frac{d^2}{dt^2}h$$

と書け、 $R$  に対して  $h$  が小さい場合のみを考えると

$$\begin{aligned}\text{右辺} &= -G \frac{M}{(h + R)^2} = -G \frac{M}{R^2} \times \left(1 + \frac{h}{R}\right)^{-2} \\ &= -G \frac{M}{R^2} \left(1 - 2\frac{h}{R} + \dots\right)\end{aligned}\quad (12.2)$$

物体に力が作用すると物体にはその力に比例した

加速度  $\alpha$  が生じる。

その比例係数は質量の逆数であり、

```
\begin{equation}\alpha = \frac{F}{m} \quad \text{\label{motion}}\end{equation}
```

となる。式 (\ref{motion}) を積分すれば運動の軌跡を求めることができる。

これを参考にして次の文章を L<sup>A</sup>T<sub>E</sub>X で書いてみましょう。

### 落体の運動

落体がなぜ落ちるかといえば、それは重力が作用しているからに他ならない。質量  $M$  の  
物体と質量  $m$  の物体の間に作用する万有引力の大きさは

$$F = G \frac{Mm}{r^2}$$

であり、向きは 2 物体を結ぶ直線の内側である。ただし、 $G$  は重力定数、 $r$  は 2 物体間の距離で  
ある。物体に力が作用すると物体にはその力に比例した加速度  $\alpha$  が生じる。その比例係数は  
質量の逆数であり、

$$\alpha = \frac{F}{m} = G \frac{M}{r^2} \quad (12.1)$$

となる。ここで加速度は  $m$  に寄らないことがわかる。ガリレオが実験して結論したことによ  
りである。

次に、簡単の為に 2 物体を結ぶ軸だけの世界を考え、物体  $M$  を原点とし物体  $m$  の座標を  
 $r$  としよう。加速度は速度  $v$  の単位時間当たりの変化量なので  $\alpha = \frac{dv}{dt}$ 、また速度は座標の单  
位時間当たりの変化量なので結局、 $\alpha = \frac{d^2v}{dt^2} = \frac{d^2r}{dt^2}$  と書ける。従って式 (12.1) は  $r$  について  
の微分方程式

$$\frac{d^2}{dt^2}r = -G \frac{M}{r^2}$$

である。

### 雨水の運動

落体の速度は時間変数  $t$  の 1 次関数であって、どんどん加速されていくはずです。雨水は  
随分高いところから落ちてきていて、たくさんの時間がかかるのはずなのに、手元ではお  
り投げるボールとそれほど速度には大差ない、ように思えます。これはどうしたことでしょうか。  
重力による加速度は物体の質量には寄らないはずです。種明しはこうです。空気の摩擦によ  
る抵抗力が速度とは逆向きに作用するからです。

$$\frac{d^2}{dt^2}h = -g + \frac{k'}{m}v \quad (1.1)$$

と書くと良いでしょう。 $k'$  は速度と抵抗力の比例係数です。 $\frac{k'}{m}$  を改めて  $k$  と書くことにします。

$$\frac{d}{dt}v = -g + kv \quad (1.1)$$

のようになります。時間が十分経った時には  $v(t = \infty) = v_\infty = \text{一定} \text{ となるとすれば, 式(1.1)} \text{ に } v_\infty \text{ を代入して}$

$$0 = -g + kv_\infty$$

を解いて、 $v_\infty = g/k$  が得られます。逆に最終到達速度を測定することによって、定数  $k$  を測定できます。

## 12.2 課題12(itemize)

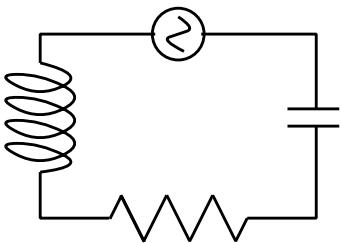
(付録、D の D.4.192 ページ)。を使って奇麗に書き換えて `platax` でコンパイルしたものを作成して下さい。

## 12.3 宿題12(circuit.tex)

次の文章を LATEX で書いて `platax` でコンパイルし、  
`circuit.dvi` を作成して下さい。

注意

- ギリシャ文字やその他の記号は LATEX の命令で記述すること。
- ドキュメントクラスは `jarticle` とし、式番号は表記の通りにならなくてもよい。



が見つかります。そこで

$$Q = Q_0 e^{iwt} + C_+ e^{-\lambda_+ t} + C_- e^{-\lambda_- t}$$

と置き、初期条件  $Q(t = 0) = 0, \frac{dQ}{dt}|_{t=0} = 0$  の元で  $C_+, C_-$  を求める。

$$Q = Q_0 \left( e^{iwt} + \frac{iw + \lambda_-}{\lambda_+ - \lambda_-} e^{-\lambda_+ t} + \frac{iw + \lambda_+}{\lambda_- - \lambda_+} e^{-\lambda_- t} \right) \quad (12.7)$$

が条件を満たす解として見つかります。

## 電気回路の微分方程式

電気回路の基本法則は電荷の保存と、閉じた回路では電圧降下の総和が0となることに尽きます。電荷の保存は回路の各分岐点における電流の保存（キルヒホフの定理）として表現されますが、ここでは分岐のない単純な閉じたループ状の回路を考えてみましょう。回路上にコンデンサ（C）、抵抗（R）、コイル（L）、起電力（E）があるとしましょう。

$$E = V_C + V_R + V_L \quad (12.5)$$

が基本的な方程式です。 $V_C, V_R, V_L$  は C, R, L それぞれにおける電圧降下です。それぞれにおける電圧降下は電荷、電流、電流の変化率で  $V_C = \frac{Q}{C}, V_R = IR, V_L = L \frac{dI}{dt}$  と表されます。また電荷の保存を考慮すれば  $I = \frac{dQ}{dt}$  ですから、

$$E = \frac{Q}{C} + R \frac{dQ}{dt} + L \frac{d^2Q}{dt^2} \quad (12.6)$$

となり、Q についての線形微分方程式となります。これを  $E = E_0 e^{iwt}$  の強制振動として、  
 $Q = Q_0 e^{iwt}$  とおいて定常解を求める、

$$Q_0 = \frac{E_0}{\frac{1}{C} - Lw^2 + iRw}$$

という振幅が得られます。 $Q = Q_0 e^{iwt} + q$  として、式(12.6)に代入すると、過渡現象を記述する部分  $q$  に対する同次方程式、

## 第13章 作図

### 13.2.3 作図

作図の補足として、マニュアルの一部を付録に添付します。(付録E,195 ページ)  
試してみよう  
tgif以外のツールを使って、それぞれの特徴や違いを確認しましょう。

### 13.1 大学で利用できる主なグラフィックエディタ

会津大学で現在利用できるツールをリストしました。(絶対パス名で表示しています。)

| コマンド                       | バージョン    | 捕捉説明(日本語対応など)                           |
|----------------------------|----------|-----------------------------------------|
| Solaris 端末と Mac OS X 端末に導入 |          |                                         |
| /usr/local/bin/tgif        | 4.2 p5   | C-SP または C-` でオン、Shift-SP でオフ。(kinput2) |
| gimp                       | 2.6.11   | C-SP でオン、オフ。(iim), ことえり                 |
| OpenOffice draw            | 3.3.0    | iim, ことえり                               |
| Solaris 端末に導入              |          |                                         |
| /usr/local/bin/xfig        | 3.2.p5a5 | Shift-SP でオン、オフ。(kinput2)               |

日本語入力に対応しているツールで日本語を利用する場合には日本語 Input Method を動作させる必要があります。(標準環境では動作しています。) Solaris 端末では iim が動作していますが、tgif と xfig では kinput2 が必要です。Mac OS X 端末では Mac-UIM と「ことえり」が動作しますが、tgif では日本語入力はできません。

### 13.2 tgif の使い方 : 補足

#### 13.2.1 印刷

講義資料では、tgif<sup>1</sup> で作った図を L<sup>A</sup>T<sub>E</sub>X ファイルへ取り込んで印刷する方法を説明しました。ここでは、L<sup>A</sup>T<sub>E</sub>X ファイルに取り込まずに印刷する方法を説明します。  
印刷は、まず、「.ps ファイル」を生成し、それをプリントアウトします。「.ps ファイル」は、「.eps ファイル」を生成したのと同じように以下の手順で作れます。

1. 「Choice ウィンドウ」の「プリント先」アイコンを「左ボタン」で「ps アイコン」になるまでクリックして下さい。
2. 「File メニュー」から「Print」を選択して下さい。
3. カレントディレクトリに、「fileName.ps」というファイルが生成されます。fileName.obj は tgif のデータ保存のためのファイル名です。

#### 13.2.2 表示

<sup>1</sup>See 'man tgif': Xlib based interactive 2-D drawing facility under X11.

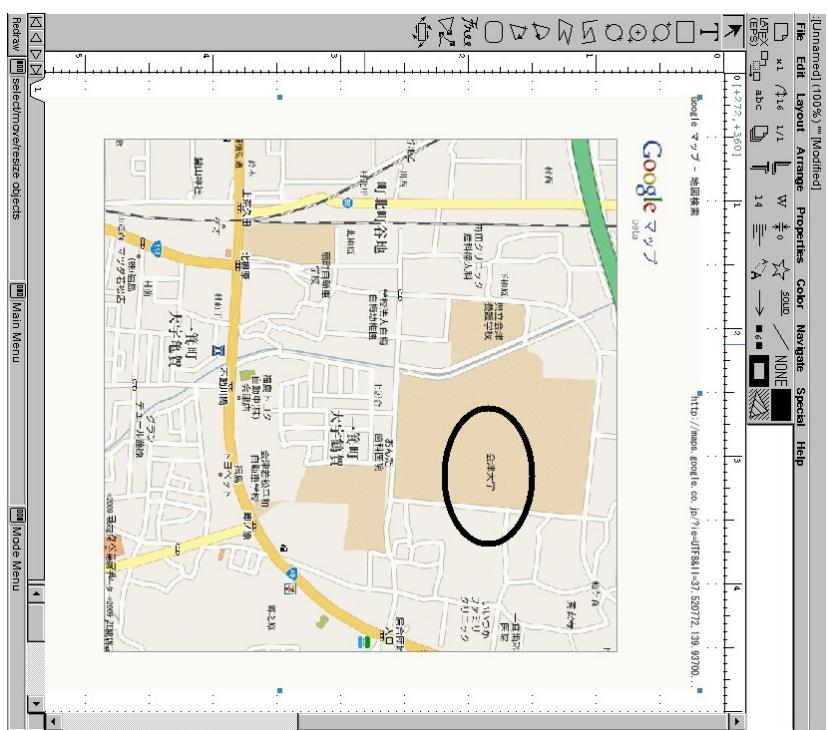
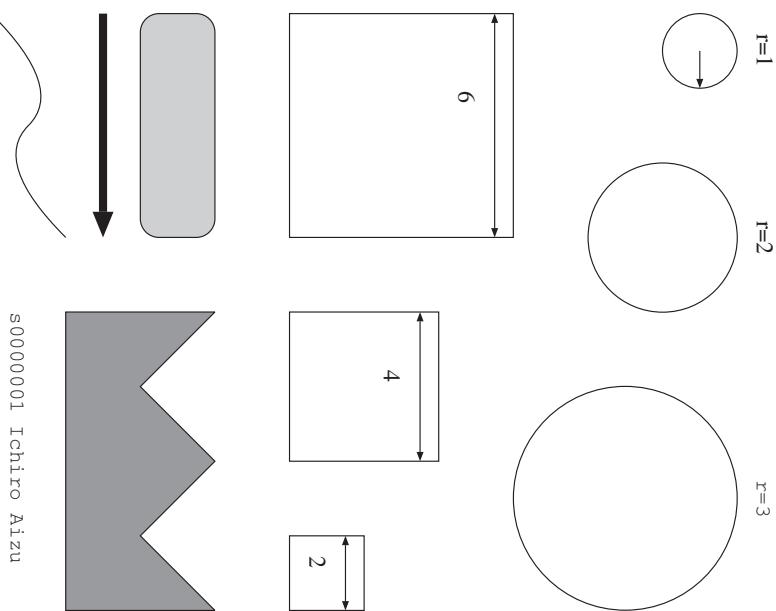


図 13.1. tgif import

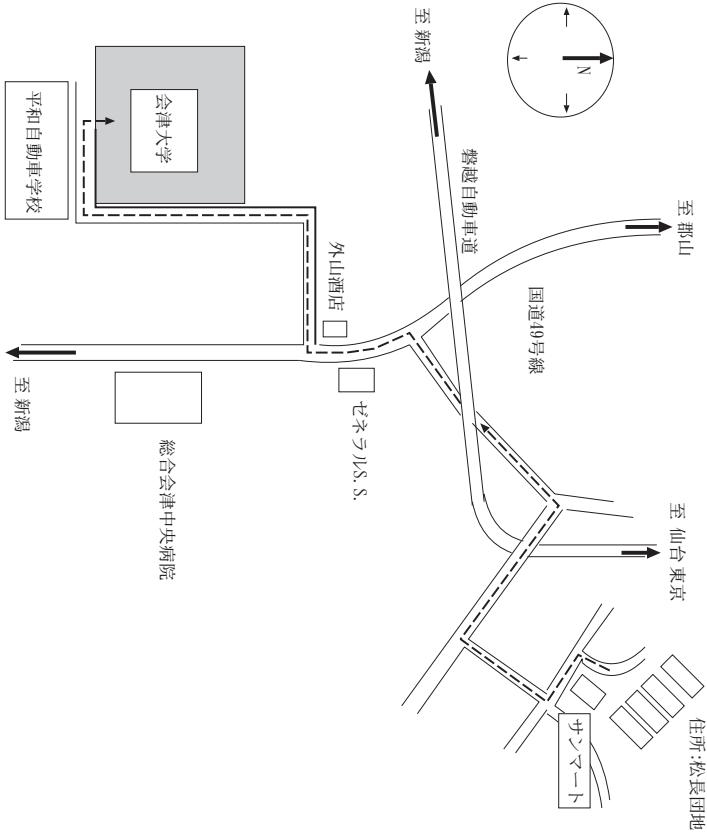
## 13.4 課題 13

1. `tgif` を使用して、以下の図を参考に絵を描きなさい。(細かい模様の違いなどは気にしないこと) 同程度の作図技術を利用していると判断できれば、自由なテーマの作図でもよいでしょう。
2. 絵のデータをファイル `kadai13.obj` として保存しなさい。ディレクトリは `$HOME/literacy` とすること。
3. 同じ絵の「.ps」ファイルを生成し、`gv` でプレビューしなさい。
4. プレビューし、確認の後に印刷して提出しなさい。学生番号、名前を忘れずに。



### 13.5 宿題 13

夏休みに友達があなたのところ遊びに来て、「会津大学に行ってみたい。」といつても、(たとえ、あなたが忙しくても)困らないように、自宅(下宿、アパート等)から大学までの道順を, *tgif* を用いて描いた図を *L<sup>A</sup>T<sub>E</sub>X* ファイル *maptex* に取り込むとともに、文章で説明しなさい。図中、説明の参考となる建物を3つ以上書きなさい(もちろん、自宅、大学は除く). *map.dvi* と *map.tex* を採点対象ファイルとします。



ただし、電車通学の場合は自宅から最寄りの駅までの地図を書いてください。プライバシー保護のために自宅を描きたくない場合には、会津の観光地から会津大学までの地図を書くと良いでしょう。

### 第14章 UNIX の試験

今回は UNIX の利用方法についての試験を行ないます。試験用紙は演習の時間に配布されます。答案の作成ができたら提出し、余った時間は宿題をするなどして下さい。

## 第15章 L<sup>A</sup>T<sub>E</sub>Xの応用

### 15.1 L<sup>A</sup>T<sub>E</sub>X(Tabular)

#### 15.1.1 複雑な表を書く

複雑な表を書くには\clineを利用します。minipage環境を用いれば、表と文章を並べるといった変則的なレイアウトを行なうことができます。まず、例を見てください。

サンプル

ソース

```
\begin{tabular}[t]{|l|c|c|} \hline
\cline{1-1} 専門基礎科目一覧 \\ \hline
確率統計学 & 2 & 3 年前期 \\
幾何学 I & 2 & 2 年後期 \\
幾何学 II & 2 & 3 年前期 \\
幾何学 III & 2 & 3 年後期 \\
量子力学 & 2 & 2 年後期 \\
統計力学 & 2 & 2 年後期 \\
\hline 単位合計 & 12 & \\ \hline

```

```
\begin{minipage}[t]{14zw}
選択科目から 2 科目 4 単位以上
(コンピュータソフトウェア学科
では、幾何学 I, II を必修とする。)
\begin{table}[t]
\begin{array}{|l|c|c|} \hline
\end{array}
\end{table}
以上のように、コンピュータソフトウェア学科では幾何学 I と II を受講すれば専門基礎
科目の要求は満たされる。

```

| 専門基礎科目一覧 | 2  | 3 年前期 |
|----------|----|-------|
| 確率統計学    | 2  | 2 年後期 |
| 幾何学 I    | 2  | 3 年前期 |
| 幾何学 II   | 2  | 3 年後期 |
| 幾何学 III  | 2  | 2 年後期 |
| 量子力学     | 2  | 2 年後期 |
| 統計力学     | 2  | 2 年後期 |
| 単位合計     | 12 |       |

| 選択科目から 2 科目 4 单位以上 (コンピュータソフトウェア学科 では、幾何学 I, II を必修とする。) |
|----------------------------------------------------------|
|                                                          |

以上のように、コンピュータソフトウェア学科では幾何学 I と II を受講すれば専門基礎科目の要求は満たされる。

tabular 環境で作られる表も minipage 環境の中で書かれたテキストの塊も、組み版の作業中はそれぞれ大きな「1 文字」として扱われています。したがって、これらが一行の中に組こまれたとることができます。上の例では、minipage の横の長さを「8cm」と指定しています。長さは適当に選びますがこの指定を省くことはできません。  
[t] の指定は "top の略" で大きい 1 文字のてっぺんを基準線に合わせよ" ということです。これをともに [b] (bottom の略) と指定すると底がきれいにそろいます。

#### 表整形コマンドの説明

- 水平あるいは垂直方向に、強制的にスペースを入れるには、\hspace と\vspace を使います。
- \cline は \hline と似ていますが、何番目から何番目の欄まで罫線を通すかコメントでれます。
- c は中央 l は左寄せ、r は右寄せです。他に p{3cm} を使用すると 3cm の幅で自動的に折り返します。p{10zw} は全角 10 文字で折り返しです。アルファベットの M の標準フォントの横幅の長さを基準として p{10em} は M の 10 文字長で折り返しです。
- 列を連結したい場合には multicolumn(連結する列の数){表示形式}{文字列}を利用します。

#### 15.1.2 拡張版表組環境 (tabularx)

tabular 環境を拡張したものに tabularx 環境があります。この機能はパッケージ tabularx で提供されるので、

```
\usepackage{tabularx}
```

をプリアンブル部に書く必要があります。Tabularx は横幅を均等割りできます。X を使うと均等割りができます。ただし、X は左寄せのみです。

```

ソース
\begin{tabularx}{392zw}{| c || x | x | x |}
\hline
番号 & 料理 & 材料 (4皿分) & 一皿当たりのカロリー (Kcal) \\
\hline
1 & カレー & 肉 200g、玉ねぎ2、人参2、じゃがいも2、市販のルーなど & 900 \\
2 & シチュー & 肉 200g、玉ねぎ2、人参2、じゃがいも2、市販のルーなど & 900 \\
\hline
3 & 肉じゃが & 肉 200g、玉ねぎ2、人参2、じゃがいも2、糸こんにゃく200g、醤油、砂糖 & 900 \\
\hline
\end{tabularx}

```

| 出力 |      |                                         |                      |
|----|------|-----------------------------------------|----------------------|
| 番号 | 料理   | 材料 (4皿分)                                | 一皿当たりのカロリー<br>(Kcal) |
| 1  | カレー  | 肉 200g、玉ねぎ2、人参2、じゃがいも2、市販のルーなど          | 900                  |
| 2  | シチュー | 肉 200g、玉ねぎ2、人参2、じゃがいも2、市販のルーなど          | 900                  |
| 3  | 肉じゃが | 肉 200g、玉ねぎ2、人参2、じゃがいも2、糸こんにゃく200g、醤油、砂糖 | 900                  |

## 15.2 Tabular の練習

以下の例を参考にして自分が修得した技のリストをつくって下さい。保存するファイル名は icando.tex とし ディレクトリ literacy に置いて下さい。

## 15.3 LATEX(Macro)

### 15.3.1 マクロ

繰り返し利用する文やコマンドを短い名前で定義して利用することは、よく行なうことです。定義の基本形は、`\newcommand{[コマンド名]}{[引数の数]}{[置き換えられるコマンドやテキスト]}` のようになります。次の例をみてください。この例では見出し `\rbox` と `\midashi` を定義して利用しています。

```

ソース
\newcommand{\rbox}[1]{\rule{#1}{0pt}}
\newcommand{\midashi}[1]{\vspace{0.5cm}\par\noindent%
\rule[2pt]{textwidth}{1pt}\large\gt\bf{#1}\%%
\rule[8pt]{textwidth}{2pt}\par\vspace{0.5cm}%
\midashi{\rbox}{について}%
\midashi{$backslash$par の機能}%
\verb+\+par は段落の閉じの意味になります。

```

このボックスは、水平垂直線を引く `\verb+\+rule` コマンドを使って作る。コマンドの一般形は

```

\verb+\+rule[深さ]{幅}{高さ}%
で、
\midashi{$backslash$par の機能}%
\verb+\+par は段落の閉じの意味になります。

```

出力

このボックスは、水平垂直線を引く \rule コマンドを使って作る。コマンドの一般形は  
\rule[深さ]{幅}{高さ}

です。

\par の機能

\par は段落の閉じの意味になります。

上のサンプルソースの数値 2pt を 6pt に変更してみましょう。どこがどの様に変化するか、観察すると深さと高さがどの部分かを理解できるでしょう。

### 注意

\midasi マクロの定義で、行末に % が付いていることに注意してください。マクロの定義が長いときは途中で改行してかまいませんが、空白が入ってほしくないところでは行末にかならず % を付けます。%がないと、行末に入っている見えない **RTN** (改行コード) が半角空白の意味を持ったり、不要な空白が出来たりと命令としてきちんと定義されないとがあります。

\newcommand を使ってマクロを定義しソースファイルの適当な位置に書いておきます。このような定義は、ソースファイルの中でその命令を使いたいところより前ならどこに書いてかまいません。特に決まりはありませんが、コマンドの定義はプリアンブル部、つまり

```
\begin{document}
```

の前にまとめるのが通常です。

### 15.4 課題 15

英語の論文では正しい綴り(スペル)で記述する必要があります。(もちろん、日本語の綴りも正しく書く必要があります。) 英文で書かれた LATEX のソースファイル sample.tex のスペルチェックをする為には以下の方法があります。

- aspell -c sample.tex
- Emacs で sample.tex を編集中に ispell-buffer を実行する。

ソースファイルが無い場合など、他人から貰った dvi ファイルの場合は dvitidy を利用して一旦テキストデータに変換したものをスペルチェックします。  
/home/course/lit1/pub/lit1/bad-spell.dvi をスペルチェックし、訂正したテキストファイルを correct-spell.txt に保存せよ。  
bad-spell.dvi に含まれる単語 rescripts は正しい単語です。検索して意味を確かめてみましょう。

### 15.5 宿題 15

以下の中から一つを選んで下さい。ファイル名はそれぞれ指定されています。言うまでもないことが他人が作成したものとの類似性が高いものは採点しません。

1. 市販の履歴書よりも便利で美しい履歴書を書いてみましょう。  
優れた作品を学内で利用していきたいと思います。プライバシー保護のため、履歴書には実在する人間の情報を記入してはいけません。rireki.sho.tex
2. その他にあると便利そうなものを作つてみましょう。  
優れた作品を学内で利用していきたいと思います。nice-templates.tex

コラム 3.

LATEX には不思議な命令があります。 \phantom です。

- 演習室 \underline{\phantom{1,2,3,4}} には Sun の \fbox{\phantom{英語}} キーボードがあります。 \\\ A の左横に Ctrl キーがあるのが特徴です。

- phantom 有り
- 演習室 \_\_\_\_\_ には Sun の        キーボードがあります。  
A の左横に Ctrl キーがあるのが特徴です。

- phantom 無し
- 演習室 1,2,3,4 には Sun の **英語** キーボードがあります。  
A の左横に Ctrl キーがあるのが特徴です。

上の例のように同じ大きさの空白を作ります。穴空きの試験問題を作るときなどに重宝します。

## 第16章 L<sup>A</sup>T<sub>E</sub>Xスライド作成

### 16.1 Beamer

研究発表や商品説明、企画説明などの様々な場面で、プレゼンテーションをする機会があります。その場合に詳細な文書をそのまま使用するよりは要点を大きな見出しにし、理解し易いレイアウトにして発表すると便利なことがあります。つまりスライドです。今回の演習ではL<sup>A</sup>T<sub>E</sub>Xを利用してスライドを作る方法を紹介します。ドキュメントクラスbeamerを利用するとスライドを作ることができます。beamerの他にもスライド作成のためのクラスは複数あります。調べてみましょう。

#### 16.1.1 サンプル1 基本構造

beamerのスライドのテーマは幾つか用意されています。

/usr/local/texlive/2013/texmf-dist/tex/latex/beamer/themes/theme/内にあるbeamerthemedefault.styを使用する場合には次の例のように、defaultを指定します。

```
\documentclass[beamer]{beamer}
\usepackage{color}
\begin{document}
\begin{frame}[Normal Text 1]
普通のテキストを書く。1
\begin{frame}{Normal Text 2}
普通のテキストを書く。2
\end{frame}
\end{document}
```

#### 16.1.2 サンプル2 タイトルページ タイトルページの例。

```
\documentclass[beamer]{beamer}
\usepackage{color}
\begin{frame}
\titlepage
\end{document}
```

#### 16.1.3 サンプル3 箇条書き

箇条書きの例。

```
\documentclass[beamer]{beamer}
\usepackage{color}
\begin{document}
\begin{frame}[frametitle{Itemize}]
\begin{itemize}
\item 箇条書き
\item \itemize や enumerateを使う
\begin{itemize}
\item 入れ子も可能
\item 字下げ（インデント）される
\end{itemize}
\end{itemize}
\end{frame}
\begin{frame}{\rightarrow $ \rightarrow $ \rightarrow $ \rightarrow $ pdf}
\begin{enumerate}
\item (tex \rightarrow $ \rightarrow $) : platex slide.tex
\item ($ \rightarrow $ $ \rightarrow $ $ \rightarrow $ ps) : dvips -o slide.ps slide.dvi
\item (ps \rightarrow $ \rightarrow $ pdf) : ps2pdf -SAPERSIZE=a4 slide.ps slide.pdf
\end{enumerate}
\end{frame}
```

### 16.1.3.1 PDF ファイルの生成

上のサンプルにも紹介してあるように beamer 等を使用してスライドを作った場合には PDF 形式のファイルに変換してプレゼンテーションに使用します。以下の手順を参考にしましょう。

`tex → dvi → ps → pdf`

1. `platex slide.tex`
2. `dvips -o slide.ps slide.dvi`
3. `ps2pdf -sPAPERSIZE=a4 slide.ps slide.pdf`

`ps2pdf` はスライドに限らず、一般に PS 形式のファイルを元に PDF 形式のファイルを生成するのに利用できます。

### 16.1.4 サンプル 4 数式

数式の例。LATEX なので美しい数式を書くことが簡単にできます。

```
\documentclass{beamer}
\usepackage{color} % color
\begin{document}

\begin{frame}{Equation}
\begin{itemize}
\item 式もいれることができる
\begin{equation} \pi = 4\arctan(1) \end{equation}
\item 背景の色も設定できる\\
\colorbox{yellow}{\begin{minipage}{7cm}
\begin{equation} \pi = 4\arctan(1) \end{equation}
\end{minipage}}
\end{itemize}
\begin{itemize}
\item \textcolor{yellow}{色も変えることができる}
\end{itemize}
\end{frame}
\end{document}
```

使用する色の設定方法は `color.sty` を参照してください。

```
/usr/local/texlive/2013/texmf-dist/tex/latex/graphics/color.sty
dvips を使用する場合に利用可能なカラーの名前は以下を参照してください。
/usr/local/texlive/2013/texmf-dist/tex/latex/graphics/dvipsnam.def
```

### 16.1.5 PDF を利用したプレゼンテーション

PDF ファイルを利用したプレゼンテーションをすることができます。

<http://latex-beamer.sourceforge.net/beamerexample1.pdf>

を web ブラウザでアクセスし保存します。

- Adobe Reader の場合は C-1 で Full-Screen 表示にします。ESC キーで終了します。
- evince の場合はメニューの View->Fullscreen か View->Presentation で全画面表示でできます。
- xpd の場合はマウスの右ボタンで表れるメニューのFullscreen で全画面表示にできます。

### 16.1.6 Hyperref

hyperref パッケージは LATEX の文書 (DVI, PDF 等) に以下の機能を加えます。

- HTML のハイパーテリリンク
- \ref や \cite のリンク
- \section を PDF の葉にする

これを利用することにより、紙媒体の文書の限界を超えて、インターネットでの利用に適した文書を作成できます。beamer は hyperref を標準で組み込んでいます。

### 16.2 課題 16

自分のこれまでの人生において所属したことのある団体や組織を一つ選び、それを紹介するスライドの PDF ファイルを beamer を利用して作りましょう。ただし会津大学は除きます。保存するファイルは \$HOME/literacy/beamer.pdf です。

## 第 17 章 L<sup>A</sup>T<sub>E</sub>X PStricks、アニメーション

### 17.1 PStricks

PStricks は TeX  $\Rightarrow$  L<sup>A</sup>T<sub>E</sub>X 内で直接 PostScript による描画を表示できるマクロです。

#### 17.1.1 サンプル 1 図形

線分や直線、円等を描くことができます。

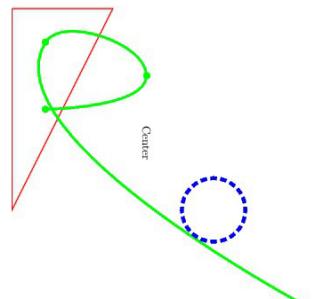
```
\documentclass[]{jarticle}
\usepackage{pstricks}
\begin{document}
\thinspacepagestyle{empty}
\begin{pspicture}(0,0)(10,10)
% Triangle in red;
\psline[linecolor=red](1,1)(7,1)(1,4)(1,1)
% Curve in Green;
\begin{pspicture}[linecolor=green, linewidth=2pt,
showpoints=true](10,10)(2,2)(3,5)(4,2)
\pscircle[linecolor=blue, linewidth=3pt, linestyle=dashed](7,7){1}
% Circle in blue;
\end{pspicture}
\end{document}
```

この例の様に `pspicture` 環境を用いると最初の `xy` 座標が矩形の左下、2 番目の `xy` 座標が矩形の右上となるような領域を確保し、その中の座標を用いて描画をすることができます。文字も指定した座標に表示できます。

#### 17.1.2 サンプル 2 グラフ

PStricks は基本的な命令や機能しか提供していませんが、これを元に拡張機能を加えたマクロが提供されています。関数のグラフを描く場合には拡張マクロ `ps-plot` を利用できます。サンプルでは `pstricks-add` を利用していますが、これは `ps-plot` 以外に幾つかのマクロを含んでいます。

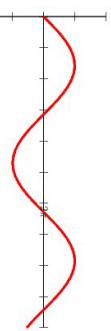
図 17.1. `pstricks0`



#### 17.1.3 サンプル 3 動く図形

beamer と組合せて円を幾つか書きます。

```
\documentclass{beamer}
\usepackage{pstricks}
\usepackage{pstricks-add}
\begin{document}
\thinspacepagestyle{empty}
\begin{pspicture}[algebraic=true, linecolor=red, linewidth=2pt]{-3}{3}{x^2}
\rput(0,0){0}
\psaxes[labels=none](0,0)(-4,-2)(4,10)
% plots the x^2
\psplot[algebraic=true, linecolor=red, linewidth=2pt]{-3}{3}{x^2}
\end{pspicture}
\end{document}
```



convert pstricks1.dvi pstricks1.gif  
を実行すると gif アニメーションが出来上がります。Firefox で表示してみましょう。

## 17.14 サンプル 4 動く文字

同様にして文字を移動させることもできます。

pstricks2.tex

```
\documentclass{beamer}
\usepackage{pstricks}
\newcommand{\hello}[2]{\rput(#1,#2){Hello}}
```

```
\begin{document}
```

```
\begin{frame}{}
```

Mathematica 7.0 for Sun Solaris x86 (64-bit)  
Copyright 1988-2009 Wolfram Research, Inc.

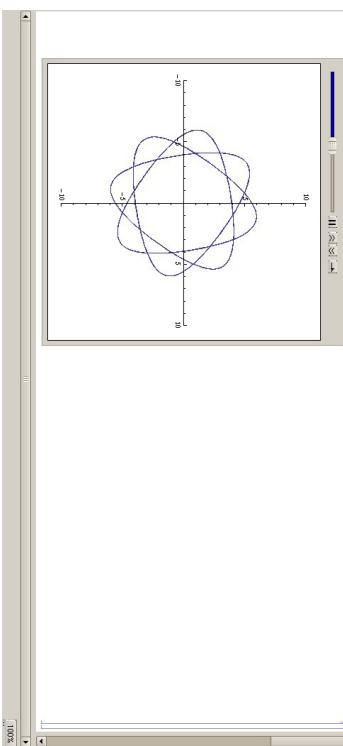
```
In[1]:= anim = Table[Plot[Sin[x + t], {x, 0, 2Pi}], {t, 0, 2Pi, Pi/5}]
In[2]:= Export["sin.gif", anim]
In[3]:= Quit
```

以上で GIF アニメーションのファイル sin.gif ができます。これを Web ブラウザで表示すると動いて見えます。

firefox sin.gif &

gif 形式以外には、avi, flv, swf 形式に出力できます。  
Mathematica のノートブックでは次のよう ListAnimate 関数を利用してアニメーションを手軽に作成し表示することができます。

```
R = 8; t = 3; a = 1.1;
ListAnimate[
Table[x, y = Sin[x]; m = {{x, -y}, {y, x}};
ParametricPlot[
m, {{Sin[t], Sin[t R/x - t]}, {Cos[t], -Cos[t R/x - t]}},
{R - x, a}, {t, 0, 2 Pi * 24},
PlotRange -> 10], {x, 0, 2 Pi, 0.1}]]
```



### 17.2.3 パラパラアニメを作る

gif 形式の画像ファイルを数個用意します。それを f1.gif, f2.gif, f3.gif とします。ImageMagick の convert を利用して以下の様にすると動画ができます。delay オプションで画像を切り替える時間を 0.01 秒単位で設定します。以下の例では 0.2 秒間隔で画像を切り替えてあります。

```
convert -loop 0 -delay 20 f1.gif f2.gif f3.gif f-anim.gif
```

### 17.3 課題 17

円が振り子運動をするアニメーションを PSTRicks と beamer と ImageMagick を使用して作成しましょう。作成したソースは furiko.tex として保存せよ。作成したアニメーションは furiko.gif として保存せよ。

#### 17.2.2 Loop

Web ブラウザでの表示を繰り返しにするためには ImageMagick の convert を利用します。

```
convert -loop 0 sin.gif sin-loop.gif
```

## 第18章 著作権およびコンピュータやネットワーク利用に関連する法律, Octave

- [ ] 15. 個人のホームページに歌手の写真を掲載、公開しても、特に宣伝などをしなければよい。

- [ ] 16. 個人のホームページに歌手の写真を加工処理してパンチの効いたラップができたので WWW のページに掲載し、公開した。

- [ ] 17. 音楽や、放送などの音源を加工処理してパンチの効いたラップができたので WWW のページに掲載し、公開した。

### 18.1 Copyright 著作権などの法律

以下に列举する事例について正しいと判断するものには○、間違いであると判断するものには×を付けなさい。間違いである場合にはその理由と訂正を記入すること。

#### 18.2 著作権に関する事項

- [ ] 11. 著作物(文章、絵、写真、デザイン、音楽、画像、プログラム)は著作者に対しては著作権があり、これを侵してはならない。
- [ ] 12. 著作物(文章、絵、写真、デザイン、音楽、画像、プログラム)は著作者と出版者の契約に基づき、出版者に版権がある場合がある、従って、これを侵してはならない。
- [ ] 13. 書籍の一部を複写し、教材として利用するために使用した。
- [ ] 14. 書籍の全部を複写し、教材として利用するために使用した。
- [ ] 15. 讀度許可のないソフトウェアを無断で譲渡した。
- [ ] 16. フリーソフトウェアには著作権がない。
- [ ] 17. フリーソフトウェアには配布条件が指示してあることがある。これを守らずに譲渡することは著作権違反になる。
- [ ] 18. ワープロのソフトくらいならコピーして複数のパソコンで使用しても構わない。
- [ ] 19. 雑誌に掲載されていた写真をスキャナーでコンピュータに取り込み Web のページに掲載し、公開した。
- [ ] 20. 誰かのホームページに掲載されていった写真を自分のファイルに取り込み Web のページに掲載し、公開した。
- [ ] 21. 新聞や通信社が発信する記事、写真などの情報をインターネット上などで無断利用することは、営利を目的としない場合は許される。
- [ ] 22. ファイル交換ソフトの winny 等を利用して、映画の複製物を取得した。
- [ ] 23. ファイル交換ソフトの winny 等は、著作物や違法ボルノなどのファイルの受信のみならず、送信のための中継サーバとしての機能が動作することから、利用者が意識していないくとも、サーバとして動作した時点で法律を犯す危険性が高い。
- [ ] 24. ファイル交換ソフトの winny 等を利用して、映画の複製物を取得するのは大学ではしてはいけないが、家のパソコンではやっても良い。

#### 18.3 肖像権に関する事項

- [ ] 25. アイドルの生写真を購入し、スキャナーでコンピュータに取り込み Web のページに掲載し、公開した。
- [ ] 26. 街角で見つけた魅力的な人物を画像に取り込み、無断で Web のページに掲載し、公開した。
- [ ] 27. 街角で見つけた魅力的な人物を画像に取り込み、uuencode でテキストに変換して無断でニュースに投稿した。

#### 18.4 民事訴訟(名誉毀損など)に関する事項

- [ ] 12. 転載不許可と明記してあるプログラムを Web のページに掲載し、公開した。
- [ ] 13. 個人のホームページにアニメのポスターを掲載し、公開した。
- [ ] 14. 個人のホームページに歌手の写真を掲載し、公開した。

## 18.5 猥褻物陳列に関する事項

- [ ] 30. 猥褻な画像を手に入れることができたので uuencode や mime-encode などの暗号化プログラムを利用してテキストに変換してニュースに投稿した。見た目には ASCII 文字列に変換されているので猥褻物陳列にはならない。
- [ ] 31. ネットワークで流れていた猥褻な画像を手に入れることができた。これは元々、流した人が悪いのであって、利用した人が罪を問われることはない。

## 18.6 商行為に関する事項

- [ ] 32. TV 番組のビデオをとり損ねたのでニュースでお願いして貰してもらった。
- [ ] 33. fleamarket を装って通信販売をしてはいけない。
- [ ] 34. 教育機関や、公共施設のインターネットサーバを利用している利用者がホームページに商用ネットのページをリンクして公開した。
- [ ] 35. 商業インターネットプロバイダを利用している利用者が、プロバイダの利用約款に触れない範囲で、ホームページに商用ネットのページをリンクして公開した。
- [ ] 36. 会津のお菓子を調べて世界に発信したかったのだが、広告活動になるので商売につながると思われ自歎した。
- [ ] 37. 会津の郷土料理を調べるために、ある料理店で取材をした。そのお礼に、店の紹介もホームページでしてしまった。
- [ ] 38. ねずみ講と呼ばれる商法は法律で厳しく禁止されており、参加するだけでも刑事罰に問われる。
- [ ] 39. 会津大学は、学術情報センターが所有する学情ネットの下流に接続されており商行為は制限されている。
- [ ] 40. 個人で家庭教師を始めたので大学の自分のホームページに広告を載せた。

## 18.7 個人情報の保護に関する事項

- [ ] 41. 他人の個人情報をインターネットで公開することはプライバシーの侵害と呼ばれ、罪を問われる。
- [ ] 42. システム管理者だけが個人情報を保護する責任を負う。

## 18.8 その他

- [ ] 43. コンピュータウイルスを作り、配布した。
- [ ] 44. コンピュータウイルスの利用法のホームページを見て、説明の通りにやってみた。
- [ ] 45. インターネットとはすなわちホームページのことである。
- [ ] 46. 匿名で画像を upload できる掲示板を提供したが、画像に人には見せたくないデータを埋め込む技術(=ステガノグラフィー, Steganography)を利用してテロリストたちが秘密の情報交換をしてしまった。

## 18.9 u-aizu.course.lit1 に質問しましょう

著作権およびコンピュータやネットワークの利用に関する法律について、もしもどうしたら良いかわからないこと  
に遭遇した場合には u-aizu.course.lit1 で相談、あるいは質問をしてみましょう。

## 18.10 課題 18

著作権およびコンピュータやネットワークの利用に関する法律について以下の問いの中から選んでファイル law 内に回答せよ。

1. 著作権およびコンピュータやネットワークの利用に関する法律について質問を述べよ。
2. ハンドアウトに列挙した"著作権およびコンピュータやネットワークの利用に関する法律に関する問い合わせ"に更に追加すべき問い合わせを述べよ。

## 18.11 Octave の利用

Octave (<https://www.gnu.org/software/octave/>) は Matlab と良く似ていますが、Octave はソースコードが公開されているフリーソフトウェアです。もちろん無料で利用できます。(Octave はソースコード以外にも Linux 用のバイナリや、Windows 用のバイナリや Mac OS X 用のバイナリも配布されています。) 同様に無料で利用できるものとしては Scilab(<http://www.scilab.org>) があります。Matlab は、非常に強力なソフトウェアで、数値計算、通信システムや制御系の設計、信号処理、画像処理などに用いられます。

オープソースの利点

MATLAB-like で、ソースコードが公開されているソフトウェアの教育・学習・研究上の大きな利点の一つは、数値計算、信号処理、画像処理ソフトのソースコードをみて、中身を理解できる点です。

ここでは Octave の簡単な利用法を解説します。Solaris 端末では Octave 3.8.2 が、Mac OS X 端末では 3.8.0 が利用できます。

## 18.11.1 行列計算

Octave をターミナル内で実行すると Octave の入力画面が現れます。Octave のプロンプトは octave:<実行番号>です。

```
octave:1> A= [1 1; 1 -1]
octave:2> B = [1 -1; 1 1]
octave:3> C = A*B
```

の様にすると行列のかけ算をすることができます。

```
octave:4> A= [1 -2 3; 2 -3 1; 3 -1 2]
octave:5> inv(A)
octave:6> det(A)
```

で逆行列や行列式を求めることができます。

```
octave:7> A= [1 -2 3; 2 -3 1; 3 -1 2]
octave:8> B= [4 5 6]
octave:9> X = inv(A)*B
```

で連立一次方程式を行列を用いて解くこともできます。B の'をつけたのを忘れないようにしてください。行列の転置の操作を表します。'をつけると、縦ベクトルになります。

## 18.11.2 プロット

関数のグラフを表示できます。

```
octave:10> t = 0: 0.1 : 2*pi;
octave:11> plot (t, cos(t), "-;cos(t);", t, sin(t), "-k;sin(t);");
```

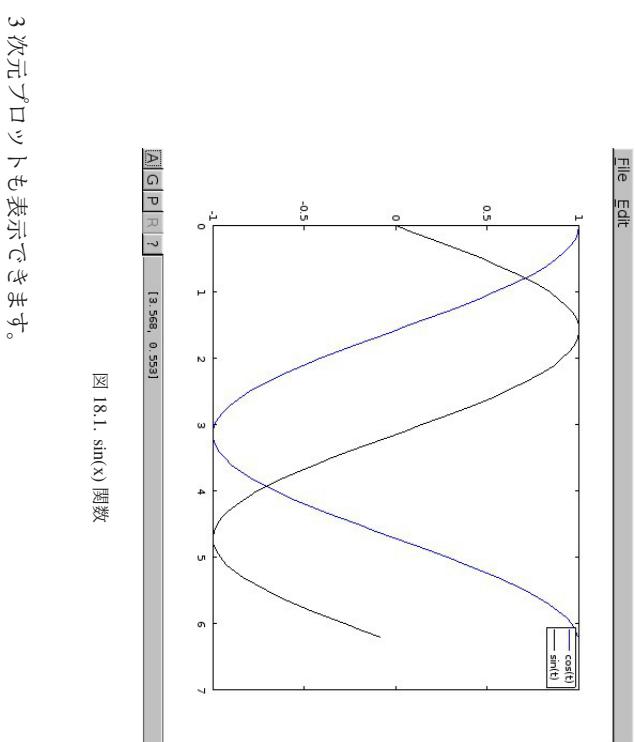
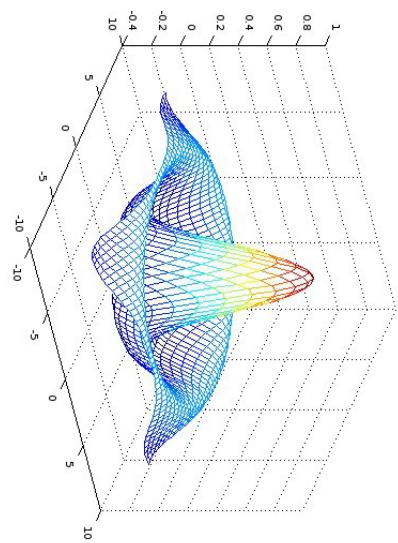


図 18.1.  $\sin(x)$  関数

3 次元プロットも表示できます。

```
octave:12> tx = ty = linspace (-8, 8, 41)';
octave:13> [xx, yy] = meshgrid (tx, ty);
octave:14> r = sqrt (xx .^ 2 + yy .^ 2) + eps;
octave:15> tz = sin (r) ./ r;
octave:16> mesh (tx, ty, tz);
```

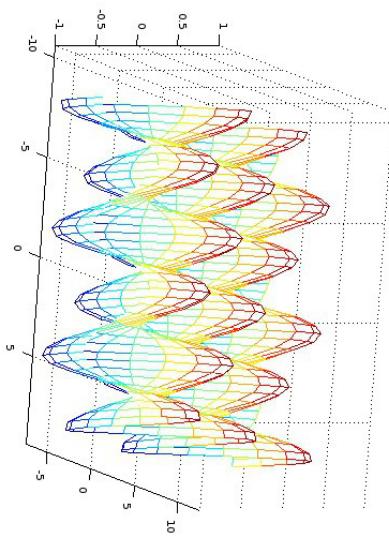


```
[A|G|P|R|?] [0.53, 7.435]
```

図18.2.  $z=\sin(r)/r$ ,  $r=\sqrt{x^2+y^2}$  の 3 次元プロット

```
octave:17> tz = sin (xx) .* cos(yy);
```

```
octave:18> mesh (tx, ty, tz);
```



```
[A|G|P|R|?] [2.078, 0.9935]
```

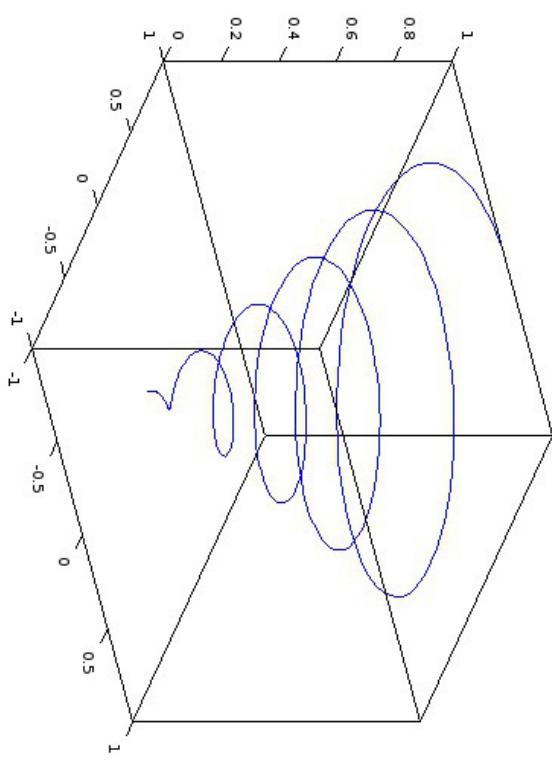
図18.3.  $z=\sin(x)*\cos(y)$  の 3 次元プロット

```
octave:19> t = 0:0.1:10*pi;
```

```
octave:20> r = linspace (0, 1, numel (t));
```

```
octave:21> z = linspace (0, 1, numel (t));
```

```
octave:22> plot3 (r.*sin(t), r.*cos(t), z);
```



```
[A|G|P|R|?] [-0.7935, 0.3011]
```

図18.4.  $(x, y, z) = (t * \sin(t), t * \cos(t), t)$  の 3 次元プロット

## 第19章 グラフ, 画像

で現在の視点を確かめることができます。show view で得られた数値を用いて、視点を設定できます。

```
gnuplot> show view
gnuplot> set view 57,21,1,1
gnuplot> splot [-2:2] [-2:2] sin(x)*sin(y)
```

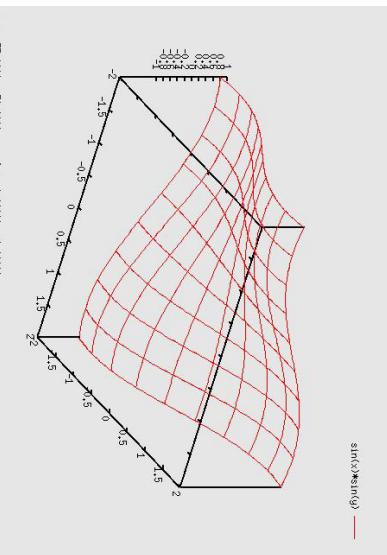
### 19.1 グラフを書くツール

図を作成するツール以外のツール、例えばグラフを書くツールを用いて作成したグラフを L<sup>A</sup>T<sub>E</sub>X に取り込むと説得力のある論文を作成できます。gnuplot<sup>1</sup>, xmgrace 等が利用できます。今回は gnuplot を紹介します。また、mathematica<sup>2</sup>を利用すれば手軽に複雑な関数の形状をグラフに表示できるので紹介します。

### 19.2 Gnuplot でグラフを書く

Gnuplot でグラフを書くことはとても便利です。Windows でも Mac OS X でも利用できます。Windows や Solaris、Linux ではグラフの出力先 (=terminal) は標準では wxt の窓です。Mac OS X ではグラフの出力先は標準では aqua の窓です。まずは手動でグラフを書いてみましょう。

```
% gnuplot
gnuplot> splot [-2:2] [-2:2] sin(x)*sin(y)
```



```
gnuplot> show view
gnuplot> quit
```

### 19.2.1 gnuplot のスクリプトファイル

gnuplot のプロンプトに命令を入力する代りに、命令を書いたファイル=スクリプトファイルを実行することができます。

```
% gnuplot file
```

### 19.2.2 グラフをファイルに出力する

```
gnuplot> set terminal jpeg
gnuplot> set output "hogehoge.jpg"
gnuplot> plot [-18:18] sin(x)/x
```

でカレントワーキングディレクトリに画像ファイル hogehoge.jpg が生成されます。画像のタイプは他に、eps, png, pdf 等があります。

### 19.2.3 データのグラフをファイルに出力する

Gnuplot の場合は以下の手順でデータのグラフを書きます。

#### 19.2.3.1 2次元プロット

```
seq 100 | awk '{x=$1/30;print x*cos(x^2),sin(x^2),sin(2*x)}' > sin.dat
```

新しい窓でグラフが描かれたたら、terminal が wxt や x11 の場合は、それをマウスでドラッグすると視点(=view)を動かすことができます。terminal が aqua の場合は動かすことができないようです。

<sup>1</sup>See 'man gnuplot': an interactive plotting program

<sup>2</sup>Mathematica, A system for doing mathematics by computer, Stephen Wolfram

```
plot "sin.dat" using 1:3 w lp
```

でデータファイルの各行の第1フィールドの数値をx座標、第3フィールドの数値をy座標とした点を順に連結した折線グラフが出来ます。`'using 1:3'`を省略すると`'using 1:2'`の動作をします。

区切り文字がカンマ(,)の場合は区切り文字を設定する。

```
set datafile separator ","
plot "sin.dat" w lp
```

### 19.2.3.2 3次元プロット

```
splot "sin.dat" using 1:3:2 w lp
```

これは3次元座標内の折線を描きます。

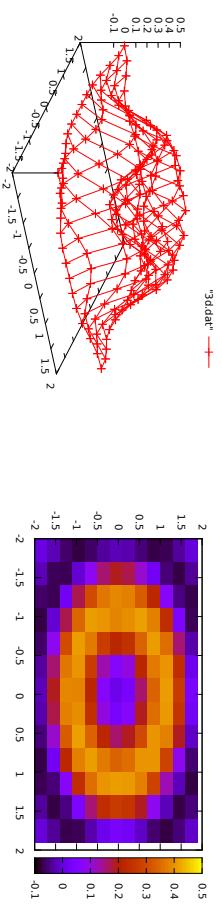
2次元格子上のデータを使うヒサーフェスプロットが得られます。例えば、以下をgrid.bashに保存し

```
x='seq -2 0.3 2'
y='seq -2 0.3 2'
for i in $x; do
 for j in $y; do
 echo $i $j | \
awk '{x=$1;y=$2;r2=x^2+y^2;z=sqrt(r2)*sin(r2)*2^(-r2);print x,y,z}'
done
done
```

によって得られるデータファイル3d.datは

```
splot "3d.dat" using 1:2:3 w lp
splot "3d.dat" using 1:2:3 w pm3d
```

とします。



グリッド型のデータが揃っていない場合や、概形を知りたい場合にはdgrid3dを有効にしてプロットします。

```
set dgrid3d 10,10
splot "3d.dat" using 1:2:3 w lp
```

こうして得られるサーフェスプロットは、大まかなもので正確性は低いです。グリッドサイズを20,20と大きくしても正しいプロットは再現できません。比較して確認しましょう。dgrid3dを無効にするには

```
unset dgrid3d
```

とします。

### 19.2.3.3 Heatmap

2次元グリッドデータを色で表現するヒートマップは

```
set view map
set pm3d corners2color c4 # グリッドの4番目の頂点のzの値を使う。
splot "3d.dat" using 1:2:3 w pm3d
set pm3d interpolate 5,5 # グリッドを5x5で分割し、なめらかにする。
splot "3d.dat" using 1:2:3 w pm3d
```

で得られます。

### 19.2.3.4 Contour map

2次元グリッドデータの等高線を表現するのは

```
set view map
set contour
unset surface
splot "3d.dat" using 1:2:3 w 1
```

で得られます。

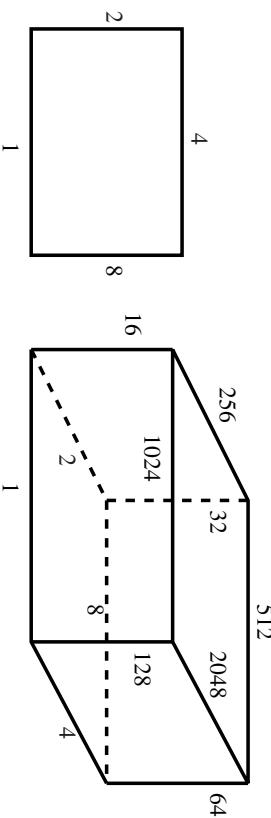
## 19.2.4 グラフを飾る

```
plot sin(x) title "sin" lt 1 lw 4 w lp, cos(x) title "cos" lt 2 lw 4
```

1. Line type — lt
2. Line width — lw
3. Line and Points — w lp

4. Line title — title
5. Border

```
set border lw 10
set border lt 3
set border (1+2+...+2048),
```



## 6. Graph title

```
set title "GRAPH TITLE"
set title font ",20"
```

## 7. Graph Key

```
set key font ",20"
```

## 8. Axis label

```
set xlabel "x"
set xlabel font ",20"
```

## 9. tics

```
set xtics 5
set mxxtics 10
set xtics font ",20"
set tics font ",20"
```

## 19.2.5 横軸を時間にする

```
set xdata time
set timefmt "%Y-%m-%d"
set xr ['2008-09-25' : '2010-03-25']
set xtics 80000000
set title "Influenza 2008-2010"
plot "influenza.dat" using 1:3 w lp
```

## 19.2.6 対数スケール

```
set logscale y
plot exp(x)
```

## 19.3 講題 19

/home/course/lit1/pub/lit1/influenza.dat にはプロジェクト<sup>3</sup>に協力した医師たちから報告された1週間のインフルエンザ診断件数が記録されている。第一フィールドから順に、開始日、終了日、総報告数、重症数、医者一人当たりの報告数が記録されている。

1. このデータから終了日と総数を awk を使って抜き出し、インフルエンザの流行の様子をグラフにせよ。その結果を eps 形式のファイル influenza.eps に保存せよ。ただし、X 軸の表示形式は yy-mm-dd とせよ。
2. このデータから終了日と重症数を抜き出し、インフルエンザによる重症化の傾向の変化の様子をグラフにせよ。その結果を eps 形式のファイル serious-case.eps に保存せよ。

## 19.4 Xmgrace の利用

Xmgrace は数値データを元にグラフを描くツールです。各行が xy 型のデータで、空白で区切られていれば、ターミナルの中で xmgrace データファイル& とタイプするとグラフを表示します。

```
seq 100 | awk '{x=$1/30;print x*cos(x^2),sin(x^2),sin(2*x)}' > sin.dat
```

で生成されたデータファイルをグラフにすると、図 19.1 が表示されます。

<sup>3</sup><http://ml-flu.children.jp/> ML インフルエンザ流行最前線情報 DB

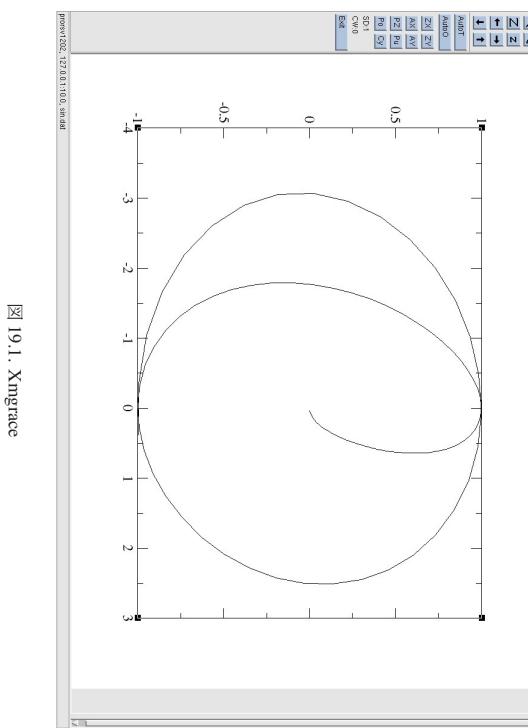


図 19.1. Xmgrace

## 19.5 Mathematica の利用

mathematica は非常にポピュラーな、数学をするための道具、です。Mathematica を利用する最も最適で標準的な方法は、ノートブックインターフェースを用することです。特に初心者には、ノートブックインターフェースにある Help Browser が手助けになります。

**mathematica & Welcome Screen ...** を選択し、Getting Started の説明を開き Mathematica の様々な機能を体験できます。Solaris 端末ではターミナルの中で math とタイプするとターミナル内で利用することができます。

```
Mathematica 7.0 for Sun Solaris x86 (64-bit)
Copyright 1988-2009 Wolfram Research, Inc.
```

```
In[1]:=
```

```
In[1]:= Quit
```

と書いてリターンキーを押すと終了します。ターミナル内ではカーソルの移動ができないのでタイプミスの修正が厄介です。その場合は Emacs 等のテキストエディタで命令を編集し、マウスでコピー＆ペーストして実行すると作業効率が上ります。

### 19.5.1 グラフを書く

グラフを描く方法の一つはを書いてリターンキーを押します。次は3次元空間にグラフを書いてみましょう。

```
In[1]:= Plot[Sin[x], {x, -2 Pi, 2 Pi}]
In[2]:= Plot3D[Sqrt[x^2 + y^2] Sin[x^2 + y^2], {x, -2, 2}, {y, -2, 2}]
```

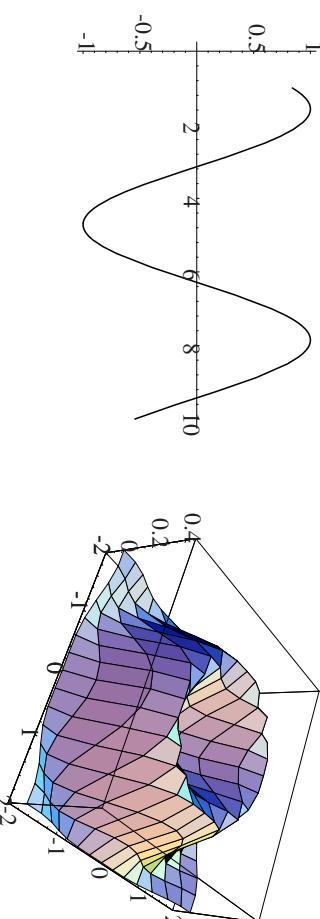


図 19.2. sin(x) 関数

図 19.3. 3D プロット

### 19.5.2 描いたグラフを画像ファイルに保存する

描いたグラフを .eps ファイルに保存するためには

```
In[3]:= Export["graph.eps", Out[1]]
```

を実行します。ファイル graph.eps が作成されたことを確認しましょう。gif や jpeg などの format でも出力できます。

## 19.6 画像

### 19.6.1 表示

ImageMagick と呼ばれるソフトウェアの display を利用すると様々な画像を表示できます。

```
display gazu.jpg
```

```
In[1]:= Quit
```

## 19.6.2 取込(Grab)

Gnome Desktop System の場合は `gnome-screenshot` を利用して Desktop の様子を画像として保存できます。(`/usr/bin/gnome-screenshot`) キーボードの Print Screen を押しても `gnome-screenshot` が起動します。Mac OS X 端末の場合は `/Application/utility` 内の Grab を利用します。Windows では Snipping Tools が利用できます。

### 19.6.2.1 ImageMagick

`ImageMagick`<sup>4</sup> を利用して X ウィンドウの画面を取り込んで様々な画像フォーマットの画像ファイルに保存できます。`display` を実行して現れるメインウィンドウをクリックして現れるコントロールパネルで `File` → `Open` を選択し、そこでファイルを選択せずに `Grab` を選択すると

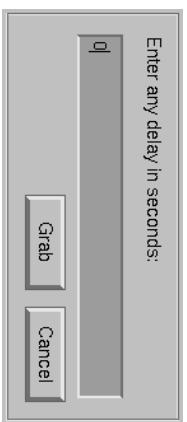


図 19.4. Grab

が表示されます。この数字で何秒後に絵を取り込むかを指定できます。この窓で `Grab` ボタンを押すと指定された時間の後にマウスカーソルが十字形に変形します。その状態でマウスをクリックするとカーソルで指定した窓の画像を取り込み、`display` のメインウィンドウに表示します。

`File` → `Save` を選択するとお好みの形式で画像データを保存できます。ホームページに貼り付けるためには jpeg 形式で保存します。いちいち画像を表示しなくても `convert`<sup>5</sup> コマンドを利用すれば変換作業ができます。

`ImageMagick` では eps 形式でも保存ができますが、`ImageMagick` の生成する eps ファイルは質が悪く、他のソフトウェアで再利用できない場合があります。そのようなときには、`ImageMagick` では一旦 jpeg などのフォーマットで保存し、その後で `bmeps` を利用して適切な eps ファイルを生成します。

```
bmeps gazou.jpg gzaou.eps
```

### 19.6.3 画像を Desktop に貼り付ける

Gnome Desktop System の場合は Desktop の System メニューから Preference → Appearance → Background で画像を追加して変更できます。

<sup>4</sup>See 'man ImageMagick': ImageMagick is an X11 package for display and interactive manipulation of images.

<sup>5</sup>See 'man convert': converts an input file using one image format to an output file with the same or differing image format.

## 19.7 レポート作成

皆さんは近い将来に卒業論文を書くことになるでしょう。良い卒業論文や研究論文、あるいは雑誌の記事を書くためにはたくさんの良い論文や記事を自分が実際に書くということを意識しながら読むことが必要になってきます。また、そうして論文の善し悪しが判るようになることに加えて、実際にレポートを作成してみることはとても良い練習となります。LaTeX, Tiff, Mathematica, ImageMagick, Xmgrace やその他のツールを利用してレポートを作成しましょう。以下に宿題と参考文献を列挙しました。

### レポート作成上の注意

講義で説明されたように、以下の点に注意して下さい。

- アブストラクト(概要)を書くこと。

- レポートの目的を明らかにすること。
- 論理構造を明確にすること,
- 参考文献があれば忘れずに参照すること。

卒業論文等は学内で公開されています。参考にしましょう。

- `/home/committee/aac/MThesisXXXX 卒業論文`
- `/home/committee/aac/DThesisXXXX 博士論文`

### 19.7.1 宿題 19

学術レポート、技術レポートとして示したタイトルの中から 1 つの項目を選択しその項目についてレポートを作成し、提出しなさい。ファイル名は `academic-report.tex` とし、`academic-report.tex` `academic-report.dvi` `academic-report.pdf` の 4 種類のファイルをディレクトリ `literacy` に保存すること。ただし、以下の条件を満たすこと。

- ソースファイルの文字コードは UTF-8 とする。

- [http://web-int.u-aizu.ac.jp/official/students/sad/stsa14\\_j.html](http://web-int.u-aizu.ac.jp/official/students/sad/stsa14_j.html) の卒論ファイル作成の指針のうち LaTeX format を利用すること。

- 4 ページ以上 6 ページ以下。

- 参考文献を 2 つ以上示し、本文内で引用箇所を示すこと。ただし、Wikipedia は参考文献としてカウントしない。

<sup>4</sup>See 'man ImageMagick': ImageMagick is an X11 package for display and interactive manipulation of images.

<sup>5</sup>See 'man convert': converts an input file using one image format to an output file with the same or differing image format.

[学術レポート]

1. 以下にリストされたタイトルに見劣りしない自由課題。

2. 我々の生活と地球環境  
毎日使用している紙や洗剤など、以外と身近な消費材と地球環境は密接に関連します。A4の紙500枚を作るには何年ものの木材を何本消費するかなど具体的な数値を調べることによって初めて地球環境を考える材料となります。

(a) 洗剤、酸化防止剤、殺虫剤、農薬、除草剤などによる化学物質による汚染。

(b) 森林伐採の目的と環境に対する影響。

(c) ペットボトルの価格。

(d) 電気自動車の燃費とガソリン自動車の燃費。

(e) 燃料電池。

(f) 環境ホルモン。

(g) 原子力発電所の寿命と放射性廃棄物の処理。

(h) その他。

3. コンピュータと我々の生活  
コンピュータの長く人間の生活は変化していきます。バラ色の未来があるのか、破滅があるのかわかりませんが、現実としてどの様な生活をもたらしたのか、あるいは近い将来もたらそうとしているかを考察すると面白いでしょう。

4. コンピュータ的魅力  
コンピュータを床の間に飾って魅力を感じる人や、性能が素晴らしいと言うだけで魅力を感じる人は別として、コンピュータは実利的な魅力に溢れています。

5. プログラム言語

• プログラム言語の種類とその特徴を解説

6. e-mailの歴史  
UNIX MAGAZIN 1993.8 連載 インターネットの利用と仕組み 4、電子メール(その1)  
UNIX MAGAZIN 1993.9 連載 インターネットの利用と仕組み 5、電子メール(その2)

7. ニュースの歴史  
ニュースは情報サービスの中では公共性があるのに参加もできるメディアです。ニュースグルーブの広がりの歴史やニュースシステムの始まりを紹介すると大変面白いでしょう。

UNIX MAGAZIN 1993.10 連載 インターネットの利用と仕組み 4, NetNews  
UNIX MAGAZIN 1993.11 連載 インターネットの利用と仕組み 5, NetNews(2)

8. インターネットに支えられる研究活動の実態  
専門的な研究の分野ではインターネットを利用した地域規模のデータベースを活用しているものもあります。個人的な電子メールのやりとりからその様な大きなシステムまでを含めて現在はインターネットなしでは研究活動を維持するのは困難です。日経サイエンス4月号の特集等を参考にすると良いでしょう。

9. UNIXのフリーソフト  
フリーソフトウェアはどの様な人々がどの様な狙いで製作するのでしょうか?歴史、現状を分析し自分なりの感想を述べるととも面白い報告になるでしょう。

10. GNUプロジェクトの紹介  
The BASIC, 1990.01, 特集 GNUとは何か?, 技術評論社  
GNU ダイジェスト  
Think GNU, 引地信之・引地美恵子, ビレッジセンター出版局

11. 自分の作りたいシステム

- 情報システム
- 計算機
- ネットワークを利用した教育・研究システム
- ネットワークを利用した商業システム

### 技術レポート

1. 以下にリストされたタイトルに見劣りしない自由課題。

2. 個人環境の整備とデザイン  
mwm, twm, tewm, fwm...などのカインドウマネージャや.cshrc, .xinitrc のカスタマイズについて。

3. ftpについて

4. システムセキュリティ - パスワード -  
Practical Unix Security by Simon Garfinkel & Gene Spafford,

5. プリンタの利用法

6. スキャナの利用法

7. 会津大学サバイバル技術

• 健康に良い食事の技術  
• 単位を取得する技術  
• 楽しく暮らす技術, etc

8. インターネット利用の際のモラル, マナーとエチケット  
インターネットは通信メディアです。電子メール, ニュース, ftp, WWW, telnet などは利用者が気持ち良く利用でき、そして法律を侵さないようにするために守るべきルールがあります。

9. mathematica プログラミング技術  
Mathematica A system for doing mathematics by computer. Stephen Wolfram mathematica はコンピュータで数学をする道具ですが、C 言語や Fortran とは異った趣を持ったプログラム言語として扱うことができます。

10. X Window System

UNIX の環境で沢山のウインドウを利用できるのは X Window System というシステムのおかげです。図書館に沢山ある X の本を参考にして下さい。

11. ソフトウェアインストール報告  
自分でソフトウェアをインストールした目的、作業、結果を報告すると良いでしょう。

## 第20章 ファイル管理(圧縮), ウェブプラウザ, FTP(1)

- \* compress *file* → *file.Z* 元ファイルは消える
- \* uncompress *file.Z* → *file* 元ファイルは消える 拡張子の省略○
- \* zcat *file.Z*

### 20.1 Compress ファイル圧縮

ご存知だと思いますがディスクの容量は無制限にあるわけではありません。  
df コマンドで自分のホームディレクトリが配置されているディスクがどのくらい使用されているか確認できます。

学生のファイルサーバには平均して1人当たり3 Gigabyte 程度の容量があります。これをぎりぎりまで使用しても良いかと言うと、そうでもありません。一時的に、作られる作業ファイルのことなどを考慮すると、安定して動くためには、ある程度の余裕が必要です。そのため、現在、学生一人あたりのディスク使用量は、3GB を上限とっています。

プログラムをコンパイルするなどいろいろな作業をする場合にテンボラリーに大きなサイズのファイルを作成することがあります。このための空き容量がないと作業ができなくなってしまいます。ですから、

- 不必要なファイルは消して下さい。
  - また、普段使用しないファイルはUSBメモリー等に移しておいて下さい。
  - ファイルを圧縮して下さい。
- という訳で今回は、ファイルの圧縮をテーマに演習をしましよう。
- (1) テキストなど「圧縮」できるデータを圧縮する方法(とそれを戻す(伸張)方法)。
  - (2) 複数のファイルを「アーカイブ」と呼ばれるひとつつのファイルにまとめる方法(とそれを戻す方法)。この時にデータの圧縮も同時に行われることが多い。
  - (3) 自分のディスク使用量を知る:
- ```
du -sk ~/.
```
- で自分のホーム・ディレクトリ以下のファイル全ての合計容量を知ることができます。
- 等の知識や技術を身につけましょう。なお、名前は似ていますがzipとgzipは機能が違うので、注意してください。簡単にまとめるに以下になります。詳細な説明はオンラインマニュアル等で確認しましょう。
- 単一ファイルの圧縮／伸張をするもの。
 - compress, uncompress
 - compressで圧縮したものはuncompressで元に戻すか、zcatで標準出力に出力します。
-
- gzip, gunzip
 - gzipで圧縮したものはgunzipで元に戻すか、gzcatで標準出力に出力します。
 - * gzip *file* → *file.gz* 元ファイルは消える
 - * gunzip *file.gz* → *file* 元ファイルは消える 拡張子の省略○
 - * gzcat *file.gz*
- bzip2, bunzip2
 - bzip2で圧縮したものはbunzip2で元に戻すか、bzcatで標準出力に出力します。
- * bzip2 *file* → *file.bz2* 元ファイルは消える
 - * bunzip2 *file.bz2* → *file* 元ファイルは消える 拡張子の省略○
 - * bzcat *file.bz2*
- xz, unxz
 - xzで圧縮したものはunxzで元に戻すか、xzcatで標準出力に出力します。
- * xz *file* → *file.xz* 元ファイルは消える
 - * unxz *file.xz* → *file* 元ファイルは消える 拡張子の省略○
 - * xzcat *file.xz*
- 複数ファイルをアーカイブ(書庫)にするもの。
- zip, unzip - 圧縮あり
- * zip *backup file1 file2 ... → backup.zip* 拡張子は追加○
 - * unzip *backup.zip* → *file1 file2 ...* 拡張子の省略○
- gtar -(z つきの場合) gzipによる圧縮あり
- * gtar cvzf *backup.tgz* → *file1 file2 ...* 拡張子の省略×
 - * gtar xvzf *backup.tgz* → *file1 file2 ...* 拡張子の省略×
- gtar -(z つきの場合) gzipによる圧縮あり
- * gtar cvjf *backup.tbz* → *file1 file2 ... → backup.tbz* 拡張子の自動追加×
 - * gtar xvjf *backup.tbz* → *file1 file2 ...* 拡張子の省略×
- * gtar tvjf *backup.tbz* 内容確認
- gtar -(J つきの場合) xzによる圧縮あり
- * gtar cvjf *backup.tar.gz* → *file1 file2 ... → backup.tar* 拡張子の自動追加×
 - * gtar xvjf *backup.tar.gz* → *file1 file2 ...* 拡張子の省略×
- * gtar tvjf *backup.tar.gz* 内容確認
- tar - 圧縮なし
- * tar cvf *backup.tar* *file1 file2 ... → backup.tar* 拡張子の自動追加×
 - * tar xvzf *backup.tar* → *file1 file2 ...* 拡張子の省略×
 - * tar tvzf *backup.tar* 内容確認

20.2 課題 20

これまでに作成した ps ファイル, dvi ファイル, テキストファイルをそれぞれ 2 通り以上 の方法で圧縮し、その結果をファイル asshuku に次の形式で報告せよ。

```
# 元ファイル 圧縮コマンド 結果ファイル 圧縮比
kadai14.tex compress kadai14.tex.Z 480/557
```

20.3 ウェブブラウザ (Firefox) の利用法

ここでは Firefox の利用に関して注意すべき点をいくつか示します。

pdf 形式の画像データがリストしてあります。これをクリックするとファイルをダウンロードし、保存しますか、あるいはアプリケーションで開きますかと尋ねられる場合があります。ディスクに保存するを選択すると一旦コンピュータのディスクに保存します。アプリケーション



図 20.2. Firefox の Applications の確認、変更

20.3.1 Application

<http://web-int.u-aizu.ac.jp/course/lit1/> のページの中には dvi 形式や ps 形式、pdf 形式の画像データがリストしてあります。これをクリックするとファイルをダウンロードし、保存しますか、あるいはアプリケーションで開きますかと尋ねられる場合があります。ディスクに保存するを選択すると一旦コンピュータのディスクに保存します。アプリケーション



図 20.1. Mozilla Firefox: Opening pdf

ンで開くを選ぶと指定したアプリケーションを用いてファイルを開くことができます。“今後この種類のファイルは同様に処理する”をチェックすると、この設定は Firefox の設定ファイルに保存され“今後この種類のファイルは同様に処理することができます”。Preference の Application タブを開いて、この設定を確認したり、変更することができます。

20.3.2 Advanced タブ - Cache

Cache (キャッシュ) とはアクセスしたページをディスクに一時的に記憶して再利用する仕組みです。Cache を利用することでページをスマートにします。リロードしても Cache に蓄えられた古い情報を表示する場合は SHIFT キーを押しながらリロードボタンを押します。

“Advanced タブ”の Network サブタブでは Cache (キャッシュ) の設定が可能です。設定された Disk Cache のサイズまでホームディレクトリ以下にファイルを保存するので、あまり大きくなり適切な数値を設定しましょう。現在の設定値を確認して大きい値であれば変更しよう。10 MB 程度で充分です。

ホームディレクトリのディスクの使用量を超えてアカウントの一時利用停止状態になることがあります。Cache のサイズが大きくなっている場合はこの窓にある **Clear Cache** の機能を利用して Cache のファイルを削除すると良いでしょう。

Cache ファイルがどの程度の大きさなのかは

- Solaris の Firefox の場合は \$HOME/.cache/mozilla/firefox/、
- Mac OS X の Firefox の場合は \$HOME/Library/Caches/Firefox/

以下の Cache ディレクトリ内を見るとわかりますが、大雑把には

```
du -s ~/cache/mozilla/firefox/~/Library/Caches/Firefox/
```

a

をターミナル内で実行するとわかります。
~/.mozilla は Solaris の Firefox の個人用設定ファイルやブックマーク、Cache ファイルなどが置かれるディレクトリです。Mac OS X の場合は Cache のディレクトリとは異なり、



図 20.3. キャッシュの設定

以下にバックマーケットファイルなどが置かれます。

20.3.3 lock ファイル

Firefox を一旦終了させた後に、次に Firefox を起動したときに、エラーメッセージが出で起動できなことがあります。この様な状況が発生するのは、Firefox の lock ファイルが残留している場合があります。lock ファイルが残留するのは、終了が正常ではなかったからです。正常な終了とは Firefox のメニューにある終了を実行することです。(C-w を押すことで窓を消すことができます。最後の窓を消すと終了となります。) X Window System の多くの Window Manager によって提供される窓枠の終了ボタンは正常な終了をしないと考えてください。Solaris 版の Firefox では

```
find ~/mozilla/_name.lock
Mac OS X 版の Firefox では
find ~/Library/Application\_Support/Firefox/_name.lock
```

をターミナル内で実行して、lock ファイルの存在を確認できます。このファイルを削除すると問題が解決するでしょう。

lock -> 163.143.47.146:+20829
これは IP アドレスが 163.143.47.146 のホストでプロセス ID が 20829 の Firefox が動作しているということを明示します。実際に 163.143.47.146:+20829 という名前のファイルは存在しません。

Windows や Mac OS X では窓枠の終了ボタンをトリガーにして正常な終了プロセスを実行することができます。X Window System では様々な Window Manager が提供する終了ボタンは強制終了の意味の本当の kill 命令なのです。だから正しい終了プロセスを実行できない。そのような理由で lock ファイルが残留してしまう場合があります。

20.4 FTP (file transfer program/file transfer protocol)

ネットワークを利用してコンピュータ間でファイルを転送することができます。インターネット標準のファイル転送プロトコル(FTP)のユーチュイナーフェースのいくつかを紹介します。FTP の利用方法を理解しましょう。

FTP のサーバプログラムが動作しているコンピュータに FTP のクライアントプログラムを利用してアクセスすることができます。

/usr/bin/ftp	標準の ftp クライアントプログラム。
/usr/local/bin/ncftp	便利な機能を持つ ftp クライアントプログラム。
/usr/local/gnu/bin/wget	ファイルを FTP,HTTP,HTTPS を利用してダウンロードするプログラム。
/usr/local/bin/lftp	パスワードやデータを暗号化してやり取りする安全な ftp 各種 Web ブラウザ。

20.4.1 FTP の練習

FTP のサーバプログラムが動作するコンピュータにアカウントを持っていればそのアカウントを利用してアクセスすることができます。個人の Web ページのファイルを置くサーバ webis-ext (学外用)、webis-int (学内用) には sftp を使ってファイルをアップロードします。

20.4.2 SFTP

通常の FTP ではユーザー認証のために利用するパスワードは平文でネットワークを流れるので、安全ではありません。学外の計算機からインターネットを経由して学内のファイルサーバへアクセスし、ファイルを転送する場合、パスワード等を暗号化してサーバとやり取りをする SFTP を利用することがより安全です。

情報センターのホームページを参照して利用してください。

<http://web-int.u-aizu.ac.jp/labs/istc/ftp/topic/sshgate.sshgate.html>

```
prorsv1205fa-fujituj51: cat > newfile
It is new.
prorsv1205fa-fujituj52: sftp webfs-int
a-fujitu@webfs-int's password:
Remote system type is POSIX.
sftp> ls
htaccess/
public_html/
sftp> cd public_html
kuma.jpg
sftp> put newfile
newfile
sftp> ls -l
drwxr-xr-x 3 a-fujitu prof 12 Jan 23 4:08 .
drwxr-xr-x 4 root other 4 Sep 8 2:16 ..
-rw-r--r-- 1 a-fujitu prof 29907 Jul 17 2:27 kuma.jpg
-rw-r--r-- 1 a-fujitu prof 11 Jan 23 4:08 newfile
sftp> bye
prorsv1205fa-fujituj53:
```

```
| 11B | 1.1kB/s | TOC: 00:00:00 | 100%
200 PORT command successful
150 Opening ASCII mode data connection for file list
private
pub
u-aizu
ftp
226 Transfer complete
27 bytes received in 0.0015 seconds (17.14 Kbytes/s)
ftp> cd pub
250 CWD command successful
ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection for file list
tex
NetBSD
parallel
news
net
gnu
misc
org
x11
FreeBSD
cmd
Mac
...
CPAN
PC
SciEng
Lang
Ccp
226 Transfer complete
146 bytes received in 0.0026 seconds (54.61 Kbytes/s)
ftp> bye
221 Goodbye.
prorsv1205fa-fujituj52:
```

20.4.3 匿名FTP

FTPのサーバプログラムが動作しているコンピュータにアカウントを持たない者はゲストユーザにアクセスを許すサービスを、匿名(=anonymous)FTPサービスと呼びます。会津大学では

```
ftp://ftp.u-aizu.ac.jp

で匿名(anonymous)FTPサービスを提供しています。Webブラウザを利用して
ftp://ftp.u-aizu.ac.jp にアクセスしてみましょう。パスワードを尋ねられる
場合は、メールアドレスを入力します。次のサンプルは /usr/bin/ftp を利用してアクセス
したものです。
```

第21章 試験：ネットワークを利用する場合のエチケットや倫理、著作権およびコンピュータやネットワーク利用に関連する法律

21.1 競争的課題

講義の第1.10章の21ページに記載されている競争的課題の締め切りは今回の演習日の24:00です。

コラム4.

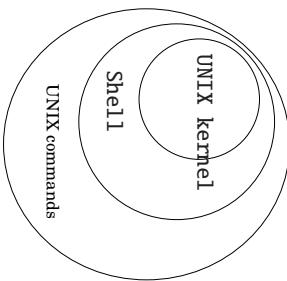
窓の中に表示される文字列をコピーしたり、貼り付けたりすることは、頻繁に行います。多くのアプリケーションの編集メニューには必ずCopy/Cut/Pasteがあります。X Window SystemではCopyはマウスの左ボタンを押しながら文字列をドラッグし、Pasteでは真ん中ボタンを押します。
しかししながら、WindowsやMac OS Xではどうでしょう？昔のWindows用のマウスは2ボタン、Mac用のマウスは1ボタンでした。X Window Systemのようには操作出来ません。また、Windowsのさまざまなアプリケーション、特にワードやエクセル等のOfficeではマウスの左ボタンを押しながら文字列をドラッグをした後に"Ctrl+C"でコピーをし、貼り付けたい場所で"Ctrl+V"でPasteする仕組みになっていました。Macの場合も同様で、Ctrlキーの代わりに"Command"キーを使用するのが異なるだけで後はWindowsと同じです。
次第に他のアプリケーションもそれを真似るようになります。何しろ大衆の求めにたいて妥協をしなければアプリケーションは使って貰えなくなります。そうして、X Window Systemで動作するアプリケーションでも"Ctrl+C"、"Ctrl+V"を利用する物が増えときました。Adobe Reader, Firefox, Thunderbird, Mathematica等がそうです。
では、困ることもあります。どちらの流儀で動作するのか解らない場合や、異なった流儀のアプリケーション間でどの様にして文字列をコピーペーストすれば良いか不明です。どちらの方法も有効であれば良いのですが、無効な場合は本当に困ります。

第 22 章 シェル：計算機利用環境

のシェル環境になります。exit で終了し、元のシェル環境に戻ることができます。zsh というものもあります。
ログインシェルを変更するためには chsh を使用します。

22.1 シェルとは

シェルという言葉を初めて耳にする人が大半だと思われます。シェルとは貝とか殻のことです。シェルによって表示されるコマンドプロンプトはコマンド入力のインターフェースです。ここにユーザが命令を書いてリターンを押すとシェルが命令を解釈し、命令に応じた処理が実行されます。または、シェルはファイルに書かれた命令を読んで実行します。シェルが動作していなければ UNIX はユーザが命令を読み解釈し、命令に応じて、ではなぜ貝殻と呼ぶことになったのでしょうか？UNIX は「一つ一つのプログラムは汎用性を高め、それらを上手く組み合わせて柔軟に複雑な処理ができる」という原理の元に開発されてきました。従って、一つのプログラムは一つの動作のみを行なうように作られてきました。そのように多くの単純なコマンドを組み合わせて複雑な動作をさせるには「コマンドの実行を制御する仕掛け」が必要です。それが /usr/bin/sh、または類似のコマンド(csh, tcsh, bash)です。その「コマンドの実行を制御する仕掛け」を利用して複雑な動作をする別のコマンドを作る(プログラムする)ことができます。そうしてできる複雑なコマンドが実行できるのは“UNIX 本体(カーネル)+多くの単純なコマンド+「コマンドの実行を制御する仕掛け」”という構造があるからです。



Solaris や Linux, Mac OS X などでは実際にシステム動作のためのプログラムの一部はシェルスクリプトで提供されています。システム起動時に実行される /etc/rc* 以下を参考にしてください。

これを見て中心に UNIX オペレーティングシステムプログラムのカーネル、そのまわりにシェルの本体の殻、殻の外にさらに多くの単純なコマンドの殻、その外側に複雑コマンドという構造を想像できるので殻構造、すなわちシェルと呼ぶことになり名前も sh となつたのでしょう。UNIX 本体(カーネル)だけではオペレーティングシステム(OS=Operating System)とは言えないから、シェルと基本的なコマンドは OS を特徴付けるものです。

ログインシェルを確認しましょう

```
getent passwd username
```

を実行して得られる出力の最終フィールドがログインシェルです。

22.2 Shell 環境設定上の注意

全ての人ではありませんが、かなりの人が先輩や友人に shell 環境設定ファイルを真似させてもらって、設定を変更しようとしているようです。何でもかんでも他人に頼る様なことでは何も身につきませんし、自分自身にとって理解できないブラックボックスを設定に利用することはセキュリティとしても危険です。自分自身が理解できる範囲に変更を制限する様に自制しましょう。

```
more ~stdskel/.bashrc
```

で表示されるように 1 年生の標準環境は設定されており、コマンドサーザパス等の基本的な設定に関して最新の設定になっています。講義の第 12.2.1 章を参照してください。
csh の場合は .cshrc です。

22.2.1 プロンプトを設定しましょう

ターミナル(bash)で以下を試してみて、一時的にプロンプトを変更してみましょう。
bash の場合は PS1 で設定します。

- export_PS1="\u@h:\w[\!]\\$\\"
- export_PSL="\u@h:\w[\!]\\$\\"

csh の場合は prompt で設定します。

- set_prompt="U-AIZU[_!_].%"
- set_prompt="Good_Job[_!_].%"

22.3 Desktop 環境設定上の注意

Solaris 端末でも Mac OS X 端末でもユーザの \$HOME/Desktop にあるファイルやフォルダが Desktop に表示されます。それ以外のアプリケーションアイコンや Gnome Panel、dock 等は \$HOME/Desktop に関係なく各システムが表示しています。

22.3.1 Gnome Desktop System

Solaris 端末のための標準環境である Gnome Desktop System では、以下がカスタマイズ可能です。

- Gnome Panel へ追加、削除
- Desktop への Launcher の作成
- Startup Program の追加、削除

22.3.2 Mac OS X

Mac OS X 端末の場合はドックのカスタマイズが可能です。Mac OS X 端末で X11 を実行する場合には\$HOME/.xinitrc.mac が実行され、quartz-wm が動作します。quartz-wm は Mac OS X 標準の Aqua インターフェースと親和性の高い Window Manager です。

22.4 作ってみましょう

各自の literacy に cd して下さい。そこでシェルスクリプトをやってみましょう。シェルスクリプトは一行が一つの命令になっています。ターミナルで行なうことのできる操作を集めて一行ずつ実行するプログラムだということができます。最初の簡単な例として mytest というファイルを編集します。

22.4.1 初級コース (mytest)

```
#!/usr/local-gnu/bin/bash
# comment is possible after '#'.
echo "This is a beginner course of shell programming."
echo "Please enjoy yourself with programming."
```

この文を書いて mytest に保存しましょう。次にファイルのパーミッションを確認してみましょう。

```
ls -l mytest
```

を実行します。オーナーの実行許可が出ていない場合はパーミッションを変更しましょう。

```
chmod u+x mytest
```

を実行します。オーナーの実行許可が出ていることを確認できたら実行してみましょう。

```
./mytest
```

を実行します。これで最も簡単なシェルプログラムができました。

22.4.2 会話的な処理 (Interactive Processing)

期待通りの動作をしない場合はタイアップミスがあるかも知れないのに注意してスペルを確かめましょう。シェルの命令は 改行 によって実行へ移されます。最終行に 改行 があります。どちらも確かめてください。

以下では csh と bash の両方のスクリプトを例示します。違いを観察しましょう。Bash (Bourne shell) は、Linux や Mac OS X ではユーザの表示的なログインシェルとして提供されています。また Bash ではユーザ関数も利用できるので柔軟なプログラミングが可能です。

```
#!/usr/local-gnu/bin/bash
echo "Are you fine today? (y/n)"
read henji
if [ "$henji" = "y" ]; then
    echo "That's good."
elif [ "$henji" = "n" ]; then
    echo "Bad."
fi
```

```
#!/usr/bin/csh -f
echo "Are you fine today? (y/n)"
set henji = $<
if ( $henji == y ) then
    echo "That's good."
else if ( $henji == n ) then
    echo "Bad."
endif
```

会話的な処理が便利な場合もあります。この内容を mytest2 に保存して実行してみましょう。この例文には

- csh では標準入力からのシェル変数への文字列の代入で \$<を使用する。

- bash では標準入力からのシェル変数への文字列の代入で read を使用する。

• 構造的 if 文

- csh では"=="などの関係演算子の前後は空白で開ける。
- bash では"="などの関係演算子の前後は空白で開ける。

が使用されています。
プログラミング言語には様々なものがありますが、大抵は手続きを登録(関数定義)することと、処理の順序をコントロール(処理制御)することに尽きます。
この様な単純なプログラムを通して、処理制御の基本的なこと

- およそ上から下にという順番で実行されるが、分岐もある。
- csh では変数に値を代入する時は '=' を使い、条件の判定には '==' を使用する。
- bash では変数に値を代入する時は '=' を使うが前後に空白は禁止。
- bash 条件の判定には '=' を使用する。
- 英語が基本になっている。

を理解してください。普通の事のようですが、納得しておかないと不安になるものです。

コマンドメニュー(gamemenu)

次は演習室1,2,3,4にインストールされているツールを選択するメニューのプログラムです。このcshの例文にはswitchが使用されています。bashの例文ではcaseが使用されています。

```
#!/usr/local/gnu/bin/bash
echo "1. soffice \
2. gnome-calculator \
3. gnome-dictionary \
4. gnome-screenshot \
5. gnome-text-editor \
6. exit"
echo -n "Choose one\
you want to play: "
read cmd
case "$cmd" in
  1) soffice &
     ;;
  2) gnome-calculator &
     ;;
  3) gnome-dictionary &
     ;;
  4) gnome-screenshot &
     ;;
  5) gnome-text-editor &
     ;;
  *) exit
     ;;
esac
```



```
#!/usr/bin/csh -f
echo "1. soffice \
2. gnome-calculator \
3. gnome-dictionary \
4. gnome-screenshot \
5. gnome-text-editor \
6. exit"
echo -n "Choose one\
you want to play: "
set cmd = $<
switch( $cmd )
  case 1:
    soffice &
    breaksw
  case 2:
    gnome-calculator &
    breaksw
  case 3:
    gnome-dictionary &
    breaksw
  case 4:
    gnome-screenshot &
    breaksw
  case 5:
    gnome-text-editor &
    breaksw
  default:
    echo "Good-bye."
endif
```

上の例では同じ様な書き方になっています。bashではselect文を使用するともっと短く書けます。

```
#!/usr/local/gnu/bin/bash
while [ "$1" != "" ]; do
  for f in $( ls *.${1} ); do
    echo $f
    done
    shift 1
done
```



```
#!/usr/bin/csh -f
while ($#argv)
  foreach f (*.${argv[1] })
    echo $f
  end
  shift
end
```

この例文には

22.5 バッチ処理

ユーザの入力を利用しないで予めコマンドライン引数を設定してそれを利用することも便利です。

```
#!/usr/local/gnu/bin/bash
echo "Are you fine today? (y/n)"
henji=$1
if [ "$henji" = "y" ]; then
  echo "That's good."
elif [ "$henji" = "n" ]; then
  echo "Bad."
fi
```

```
#!/usr/bin/csh -f
echo "Are you fine today? (y/n)"
set henji = $1
if ( $henji == y ) then
  echo "That's good."
else if ( $henji == n ) then
  echo "Bad."
endif
```

前節のmytest2の例をこの様に変更し実行の際に./mytest2 yを実行して変数の設定の様子を確認しましょう。

22.5.1 コマンドライン引数を利用する例

以下のいぢれかをfsearchという名前で保存しなさい。

- csh では argv 変数
引数の数 \$#argv, i 番目の引数 \$argv[i]
- shift 文 (引数の列を左に一つ移動し、余分は削除する。)
- csh では foreach 文、 bash では for 文
が使用されています。

```
./fsearch tex dvi ps
```

試して、fsearch を実行してみましょう。(.tex .dvi .ps ファイルを順に検索し、無い場合はその時点で終了します。)

22.6 課題 22

```
./count.bash N
```

と実行すると 1 から N までの整数を出力する bash のプログラム count.bash を作成せよ。

22.7 宿題 22

bash では 1 から 6 までの乱数 r を生成するために

```
let r=(RANDOM % 6)+1
```

を実行します。これを参考にしてサイコロを 2 つふるプログラム dice.bash を作成せよ。

第23章 ファイル管理(変換), バックアップ, 統計処理 R

確認しよう

Image Magick では convert の他に、
identify, composite, montage, compare, display, animate, import,
conjure, quantize, miff
等のツールがあります。利用法を確認しましょう。

convert が扱うファイル形式を講義資料の 69 ページの 11.2.1 節を参考にしましょう。

23.1 ファイルコンバート

文書や映像、音声などを保存するという様々な目的のファイルがあり、同一の目的のためであっても様々な種類のファイルの形式があります。すべてのファイルのタイプに対応したオールマイティなアプリケーションソフトウェアが存在すると便利なのですが、それは困難です。またそれぞれのファイル形式を扱えるアプリケーションをすべて用意せよというのも無理な注文です。

ワープロですら、文書保存形式は、ほとんど互換性が無いと言えます。そればかりか、互換性を制御するという戦略的な営業活動を行うことさえもあります。すべてのファイルのタイプに対応したオールマイティなアプリケーションソフトウェアが存在すると便利なのですが、それは困難です。また、手に入れた画像や音声データを他の目的に利用するためには特定の形式に変換(コンバート)する必要があります。例えば、

- 画像データを \LaTeX のファイルに取り込む場合などは、eps 形式。
- 画像データを HTML のファイルに取り込む場合などは、gif 形式、または jpeg 形式、

に変換する必要があります。

23.1.1 画像ファイル

23.1.1.1 convert (ImageMagick)

画像データの変換を行うツールの一つに convert があります。これは、ImageMagick という画像ツールの一つです。

画像データの保存形式には多くの種類があります。例えば 'man convert' を実行すれば 80 種類以上の画像フォーマットがリストされているのを見ることができます。これらが全てを網羅しているという保証はありませんが、convert¹ が便利なツールであると言ふことはできるでしょう。ImageMagick は Windows や Mac OS X などでも利用できます。

```
convert gazo.pnm gazo.jpg
```

Image Magick では convert の他に、
identify, composite, montage, compare, display, animate, import,
conjure, quantize, miff
等のツールがあります。利用法を確認しましょう。

23.1.1.2 netpbm

netpbm (portable bitmap format)を中心とした画像フォーマット変換ツール集です。最初のものは、1980 年代に作られました。convert よりも変換が速いです。
netpbm は Windows や Mac OS X などでも利用できます。

```
ls /usr/local/bin/*pnm* | wc
```

で ppm (portable bitmap format)に関する 73 個のプログラムが導入されていることがわかります。各種コマンドの使用方法は (<http://netpbm.sourceforge.net/>) を参照しましょう。

```
pnmtorjpeg gazo.pnm > gazo.jpg
```

23.1.1.3 bmeeps

eps → pdf 画像に変換する場合は bmeeps を使用すると良いでしょう。

```
bmeeps gazo.jpg gazo.eps
```

23.1.2 音声ファイル

音声データの保存形式には多くの種類があります。例えば 'man sox'² を実行すれば種類の音声フォーマットがリストされているのを見ることができます。sox は Windows や Mac OS X でも利用できます。(<http://sox.sourceforge.net/>)

```
sox recital.au recital.wav
```

sox が扱うファイル形式を講義資料の 70 ページの 11.2.2 節を参考にしましょう。

¹ See 'man convert': converts an input file using one image format to an output file with the same or differing image format.

² See 'man sox': SOund eXchange - universal sound sample translator

23.2 バックアップの方法

ファイルのリストアは大変手間のかかる作業です。また情報センターのファイルバックアップスケジュールとファイルを喪失した時間との関係によっては最新の状態に復旧することは不可能な場合があります。

個人が開発したプログラムや文書は宇宙でただ一つのオリジナルな作品です。このような種類の著作物のリストアは本人でさえも困難ですし、他人には不可能な作業です。それが大切であればあるほど、自分自身がファイルの喪失のリスクを下げる努力をするべきです。

個人的にバックアップをするためには、USB Memory を利用すると良いでしょう。

<http://web-int.u-aizu.ac.jp/labs/istc/ipc/release/howto/howto.html#usb>

USB のコネクタは耐久性が低く壊れやすいので、無理に抜き差ししてはいけません。500回程度の抜き差しが使用限界と言われていますので、WS 本体の USB コネクタはなるべく使いしないでください。演習室のキーボードが USB ハブとしての機能を提供していますので、キーボードにUSB のコネクタがあれば、これを利用してください。

USB ストレージとして利用した記憶メディアを放置した場合は個人情報を悪用される等の不利益が生じる場合があるので、各自が責任を持ってメディアを管理しましょう。

23.2.01 オンライнстレージ(Online Storage)

記憶媒体を手に持たなくともインターネットで提供されているオンラインストレージを利用することで、バックアップをしておくことが可能です。ただし、不利益を被らないためには、利用の規則や免責事項などを把握して、より安全なオンラインストレージを選択する必要があります。

- Google Drive <http://drive.google.com/> (無料)
- Windows OneDrive <http://onedrive.live.com/> (無料)
- Yahoo ボックス <http://box.yahoo.co.jp/> (無料)
- Dropbox www.dropbox.com (無料)

23.3 R

R³ は S および S-plus に互換な統計計算とグラフィックスのための言語であり、環境で 。gnome-terminal 等で  と入力して実行できます。 で終了します。

R は Windows, Mac OS X でもインストールし、利用できます。

23.3.1 R の使用例

23.3.1.1 代表値

データの分布の様子を表すための指標となる数値をいくつか紹介します。

```
> mean(c(0.5, 0.6, 0.4, 0.55, 0.45, 0.61, 0.41, 0.52, 0.48, 1.0))
> median(c(0.5, 0.6, 0.4, 0.55, 0.45, 0.61, 0.41, 0.52, 0.48, 1.0))
> q()
```

mean は平均を median は中央値(値を大きさの順でソートしたときに、全体の中央に入る値)を表します。いちいち値をたくさんいれるのは面倒なので、通常は以下のようにします。

```
> x <- c(0.5, 0.6, 0.4, 0.55, 0.45, 0.61, 0.41, 0.52, 0.48, 1.0)
> mean(x)
> median(x)
> min(x)
> max(x)
```

mean は平均、max は最大値、min は最小値を表します。

23.3.1.2 ヒストグラムの書き方

```
> y <- as.integer(runiif(10000,1,7))
> table(y)
> for (i in c(1:length(y))) { if (y[i] > 5) y[i] <- 5 }
> table(y)
> s <- "barplot sample"
> barplot(table(y), main=s)
> hist(y)
> postscript("filename.eps")
> hist(y)
> dev.off()
```

table 関数を利用すると頻度データを得ることができます。barplot で頻度データをプロットできます。hist でヒストグラムを表示できます。グラフを画像ファイルとして出力することができます。上の例では postscript() を使ってポストスクリプト画像をファイルに出力しています。dev.off() で出力先を画面に復帰します。他に pdf()、png()、jpg() などがあります。

³<http://www.r-project.org/>

barplot sample

ファイル /home/course/lit1/pub/literacy/math.csv を作業ディレクトリ (\$HOME/literacy) に複製し、以下の指示に従って R を利用して処理せよ。

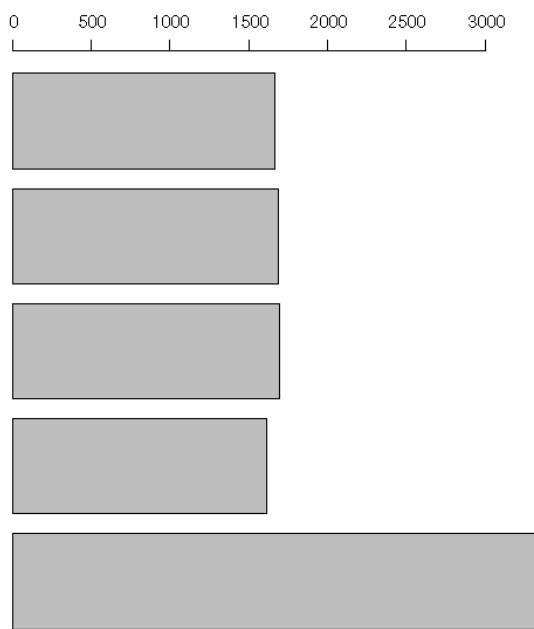


図 23.1. ヒストグラム

23.3.2 統計処理の注意点

統計処理をする上での注意点をいくつかあげておきます。

- ヒストグラムは、柱(bar)の数をいくつにするかで大きく変わってしまうことがあります。柱の数を5～15くらいの範囲で数種類書いてみて、適切なものを選ぶようにします。必要なら複数のグラフを併記したほうがよいです。
- ヒストグラムと累積度数分布も併記することを強く奨めます。
- 平均値は、異常値(内容が異常ということではなく、他のデータから離れたものを含みます)にひきずられやすい特徴があり、平均値だけで全体を代表させると危険です。
- 分散や標準偏差は、正規分布かそれに近い分布以外では、取り扱いがとても難しいです。ヒストグラムや累積度数分布図を見て、正規分布から外れているときは、注意して使うべきです。

23.5 宿題 23

取り外し可能で携帯可能な情報記録メディアを5種類以上調査せよ。ただし、それぞれのメディアについて以下の項目をなるべく詳しくレポートすること。レポートのファイル名は Removable-Media とせよ。

- | | | |
|---------|------------|--------------|
| • メディア名 | • 速度 | • 保存条件 |
| • サイズ | • 最大書き換え回数 | • 価格 |
| • 重量 | • IMByte | 当りの価格の5年間の変化 |
| • 容量 | • 動作条件 | そのほかの特徴 |

23.4 課題 23

ファイル /home/course/lit1/pub/literacy/math.csv を作業ディレクトリ (\$HOME/literacy) に複製し、以下の指示に従って R を利用して処理せよ。

```
math <- read.table("math.csv", sep=",", header=FALSE)
sc <- math[2]
plot(sc)
ssc <- sort(sc)
plot(ssc)
mean(sc)
median(sc)
min(sc)
max(sc)
var(sc)
sd(sc)
sqrt(var(sc))
table(sc)
barplot(table(sc))
hist(sc, breaks=seq(0, 100, 10))
hist(sc, br=6, col="#ab0000", border="red")
hist(sc, br=6, col="#ab0000", border="red")
hist(sc, breaks=c(0, 35, 50, 65, 80, 100), col="#ab0000", border="red")
```

- 上記の R の処理を \$HOME/literacy/R.bat に保存し、一行ずつ実行せよ。
- 各行の処理の意味と得られる結果を \$HOME/literacy/R.out に記録せよ。

3. バッチ処理 R CMD BATCH R.bat R.out2

を実行して出力 R.out2 を確認せよ。

第24章 シェルの便利な機能、シェルプログラミング(1)

- csh では 文字列修飾子、bash では文字列切り出しが使用されています。

24.1 シェルの便利な使い方

皆さんのログインシェルは C シェルか tcsh か bash に設定されているでしょう。シェルのコマンドヒストリ(履歴)を便利に利用するとキーストロークの数を減らせるので肩こり対策にも有効です。是非利用しましょう。

!! history	直前に実行した命令を実行する。 今まで実行した命令の履歴を表示する。
!5 !la ↑ 又は C-p ↓ 又は C-n	直前に実行した命令の中で la から始まる命令を実行する。 (tcsh, bash) 以前のコマンドを表示する。
ESC TAB	(csh) タイプの途中で押すと文字列を補間する。 (tcsh, bash) タイプの途中で押すと文字列を補間する。

24.2 シェルプログラミング

24.2.1 サンプル 1 (checkdvi)

これは .tex ファイルに対応した .dvi ファイルがカレントディレクトリにあるかどうかを調べるもののです。

```
#!/usr/local/gnu/bin/bash
for f in $(ls *.tex); do
    file=${f%.tex}.dvi
    if [ -e $file ]; then
        echo $f "is already compiled."
    else
        echo $f "is not yet compiled."
    fi
done
```

この例文には

- csh では foreach 文、bash では for 文
- if (-e file) 文 (file があるかどうかで真偽を与えます。)

24.2.2 少し複雑な例 – ディレクトリサーチ

次の内容を dsearch.bash というファイルに保存して実行してみましょう。

```
#!/usr/local/gnu/bin/bash
dlook="."
if [ "$1" = "" ]; then
    dlook=$1
fi
```

```
echo "Looking at $dlook directory."
echo "There are following subdirectories."
echo "Which directory do you want to see? (1=quit):"
dlist="quit,$(ls -AF $dlook | grep /)"
```

```
select dir in $dlist; do
    if [ "$dir" = "quit" ]; then
        exit
    else
        cd $dlook/$dir
        exec $HOME/literacy/dsearch.bash
    fi
done
```

```
#_Put_this_file_as_$HOME/literacy/dsearch.bash,_and_make_it_executable.
```

bash では select を用いて書くことができます。実行の際に ./dsearch.bash directory_name を実行して変数の設定の様子を確認しましょう。

csh では次の用に記述できます。

24.4 宿題 24

次のプログラムを完成させて dsearch2.bash として保存し、動作確認をせよ。

```
#!/usr/bin/csh -f
if ($1 == "") then
  set dlook=$1.
else
  set dlook=$1
endif
```

```
echo "Looking at \"$dlook\" directory."
echo "There are following subdirectories."
set dlist='ls -af $dlook | grep /'
fi
```

```
@counter=0
```

```
while ($counter < $#dlist)
```

```
  @counter++
  echo ${dlist[$counter]}:"$dlist[$counter]"
end
```

```
echo -n "Which subdirectory do you want to see?"
```

```
echo -n " (1-#${#dlist}, 0=quit):"
if (@henji < $counter+1 && $henji >= ${#dlist}) then
```

```
  if ($henji >= 1 && $henji <= ${#dlist}) then
    cd ${dlist[$henji]}
    exec ./literacy/dsearch.csh
```

```
  else
    cd ..
```

```
    select dir in $dlist;
```

```
    if ["$dir" == "quit"]; then
```

```
      break
```

```
    done
```

```
#_Put_this_file_as.$HOME/literacy/dsearch.csh_and_make_it_executable.
```

このプログラムでは

- コマンドライン引数からのシェル変数への文字列の代入
- while 文
 - 数変数 (@マークから始まる行)
 - リスト変数と配列変数
- PID を引き継いで子プロセスを実行する exec

が使用されています。csh ではこの様に少し長いソースになりました。

24.3 課題 24

プログラム dsearch.bash と dsearch.csh を完成させ、動作確認をせよ。

第25章 セキュリティ

- トロイの木馬 (Trojan Horse)
便利なソフトウェアに見せかけ、ユーザに被害を与える不正なプログラムです。ギリシャ神話の「トロイの木馬」にちなんで名付けられました。もっと多くのトロイの木馬の唯一の目的は、ユーザーのマシンからパスワードを盗むことです。便利なユーティリティソフトと見せかけて、キーボードの入力を監視しパスワードやクレジットカード番号を盗み見る、などといった悪質なもの。目立った活動はせず、不正な“攻撃者”がコンピュータに侵入するための入り口（バックドア）を仕込むものもあります。

25.1 Security Information

<http://web-int.u-aizu.ac.jp/istc/ipc/Security/> に会津大学情報センターからのセキュリティ情報が掲載されています。セキュリティとは何でしょうか？

- コンピュータやネットワークから正しい情報を取り出せること。
- コンピュータに正しい情報を保存できること。
- コンピュータやネットワークの情報サービスに適切な時間内にアクセスできること。
- コンピュータやネットワークに流れる情報を他人に盗まれないこと。
- コンピュータのプログラムの破壊がないこと。
- ネットワークの使用負荷が100%近くに達しないこと。
- 参考になるサイトなどがたくさんあります。
- セキュリティ入門 Internet Week 2000
http://www.soi.wide.ad.jp/iw2000/iw2000_tut/slides/03/index_bar.html

25.1.1 コンピュータウイルス

- <http://www.ipa.go.jp/security/isg/virus.html>

通商産業省(2001年1月6日より、通商産業省は経済産業省)が告示した「コンピュータウイルス対策基準」においては、コンピュータウイルスの定義を、『第三者のプログラムやデータベースに対して意図的に何らかの被害を及ぼすように作られたプログラムであり、次の機能を一つ以上有するもの』としています。

- 自己伝染機能
自らの機能によって他のプログラムに自らをコピーし又はシステム機能を利用して自らを他のシステムにコピーすることにより、他のシステムに伝染する。

- 潜伏機能
潜伏機能の特定時刻、一定時間、処理回数等の条件を記憶させて、条件が満たされるまで症状を出さない。
- 発病機能
プログラムやデータ等のファイルの破壊を行ったり、コンピュータに異常な動作をさせる等。

ウイルスの動作や感染のパターンによって次のように分類されます。

25.1.2 セキュリティホール

セキュリティホールとはソフトウェアの隠れた機能や欠陥(脆弱性、ぜいじやくせい=vulnerability)と言います。)が悪用された結果、情報保護システムにぽっかりと空いた穴のことを言います。コンピュータシステムへの外部からの不法侵入を許してしまった保安上の死角となります。
以下にリストしたサイトを参考にセキュリティホールを調べることができます。

- [US-CERT Alert](https://www.us-cert.gov/ncas/alerts)
<https://www.us-cert.gov/ncas/alerts>

- CERT/CC Vulnerability Notes Database
<http://www.kb.cert.org/vuls/> ソフトウェアの脆弱性のデータベースです。
- コンピュータ緊急対応センター <http://www.jpcert.or.jp/>
- IPA 情報処理振興事業協会 <http://www.ipa.go.jp/security/>

25.1.3 ホームページの改竄(かいざん)

パスワードを不正に取得されたり、コンピュータシステムにバックドア等があつたりすると、簡単にコンピュータ内のファイルを書き換えられます。ホームページの改竄とはウエバーサー内のファイルを書き換えることですから、単純な行為ですが、その被害はとても大きくなる可能性があります。

・コンピュータシステムを管理する会社のホームページが改竄されれば、その会社の管

- ある政府のホームページが改竄されれば、その政府が進める電子政府はとても不安なものになります。
- 電子商取引を行う会社のホームページが改竄された場合には、取引ができないために多額の損失を被ることになります。
- <http://www.jpcert.or.jp/ed/2001/ed010006.txt>

25.1.4 スパムメール

電子メールを使って、一方的に無差別かつ大量に送付されるメッセージ（ダイレクトメール）のことを一般に SPAM と呼びます。語源は、Hormel 社の有名な缶詰を題材に使ったテレビのコメディ番組 (*Monty Python*) によるとされています。最近では、**Unsolicited Bulk Email (UBE)**、**Unsolicited Commercial Email (UCE)** と表記されることも多くなっています。

SPAM は通常のダイレクトメールと異なり、送信者の金銭的負担が小さいのが特徴ですが、もちろんその分はメールの受信者が回線費用などで負担します。通常、受け手の興味に関係なく無差別に送り付けられるために、受信者にとっては迷惑以外の何者でもありません。SPAM メールがある組織に大量に送り込まれると、通信の混雑、メールサーバーのダウンを招く場合があります。そこまで至らなくとも、プロバイダーの場合は顧客からの苦情が寄せられ、会社組織の場合は生産性の低下につながります。こういったことから、一部の企業やプロバイダーでは、SPAM フィルターなどを用いて、SPAM メールを送信するサーバからの接続を遮断しているところもあり、下記の様な「SPAM に利用された」あるいは「第三者中継を行なう」サーバのデータベースを元に制御を行います。
<http://www.ipa.go.jp/security/ciad/antirelay.html>

25.1.5 ネットワークの盗聴

スニッフアとは、ソフトウェア、ハードウェアの区別なく、ネットワークに飛び交う情報を捉えるデバイスを指します (sniff: 「嗅ぎつける」、転じて「盗聴する」「傍受する」の意)。LAN の内部に、あるいはインターネットの通信経路の途中にスニッフィングをしているものがいるかもしれません。ですから、ネットワークにパスワードやクレジットカードの番号などを平文の形式（テキスト）で流すことは極めて危険です。無線 LAN はとても盜聴に対して無防備なネットワークです。

25.1.6 自分（のコンピュータ）を守る

- 添付ファイルを無闇に開かない。

- OS を含めてソフトウェアのセキュリティパッチを当てて、セキュリティホールを塞ぐ努力を持続する。Microsoft Windows の場合は "Microsoft Update" を実行する。

- 入手経路が不明等のいろいろな意味で、怪しいソフトウェアは使用しない。

- telnet や ftp 等のような平文がネットワークを流れるソフトウェアの使用を控えて、データを暗号化してネットワークに流す ssh や sftp などを使用する。
- ファイアウォールを設定し、コンピュータや LAN へ侵入してくるネットワーク通信パケットを制限する。
- 怪しい Web ページにアクセスしない。Web ページにアクセスしただけで感染するウイルスもあります。
- URL の詐称に騙されない。
- フィッシング (Phishing) 詐欺に騙されない。

25.1.7 組織（のコンピュータやネットワーク）を守る

- 組織内のコンピュータへのアクセス方法を外部に漏らさない。
- 組織内のユーザーアカウントリストを外部に漏らさない。
- 組織内のメールアドレスのリストを外部に漏らさない。

25.2 課題 25

自分が自身の健康、財産、コンピュータなどのセキュリティのチェックリストを作りなさい。またそれぞれのチェックポイントについて類似性を考察せよ。以上をファイル \$HOME/literacy/checkList に報告せよ。

25.3 宿題 25

自分が所有する PC や 演習室で利用している計算機のセキュリティパッチ (Microsoft 社の Windows であれば "Microsoft Update" のこと。) が発行された日付を現在までの 2 年間について全て調べよ。平均して何日に 1 件のセキュリティパッチが発生しているか？

自分が選択したコンピュータシステムについて調査し、\$HOME/literacy/security.patch に報告せよ。報告に含まれるべき以下の内容が採点対象である。調査対象としたシステム名、セキュリティパッチの ID と日付のリスト。分析結果。

第26章 ネットワークコンピューティング、プロセス管理、インターネット

(部屋の名前)(計算機の種類)(番号)

という形でつけられています。IP アドレスの最初の 163.143 は会津大学を意味します。後の二つをそれぞれ 0 から 255 の数字を使うことができます。163.143.24.X の様になっているホストは同一のサブネットに繋っているホストです。

26.1.2 異なる計算機資源

26.1 Network の利用法

会津大学の計算機ネットワークを利用すると、一つの端末から学内の全ての演習室のワークステーションを一度に利用して計算を実行させることができます。演習室 1-6だけでも現在、およそ 300 台のワークステーションがあるので自分一人だけで全てを利用できれば、ワークステーション 1 台で計算するよりも 300 倍の速さで計算を実行できることになり、ます。

夢のような話しさはこれくらいにしておいて、実際に多くの計算機資源をどのように利用すれば快適に作業ができるかを理解し、実践していくことを期待します。

26.1.1 分散化されている計算機資源

学内にどの様な計算機があるかを知るには命令 `getent hosts` を実行すれば良いで

しょう。これは、大学全体のネットワーク情報が 1 元管理されているおかげです。ホストとは計算機やプリンタ、ネットワーク機器の総称です。例えば演習室 3 教室だけをリストしようとすれば、命令 `getent hosts | grep std1 | sort` を実行します。その結果、

演習室 1,2	Desktop	Solaris
演習室 3,4	Sunray Client	Solaris
演習室 5,6	Desktop	MacOS X
ハードウェア実験室 1,2,3,4	Desktop	Windows
CALL 1,2	Sunray Client	Solaris
ilab 1,2	Desktop	MacOS X

かつてはその他に SONY 社の News, Silicon Graphics, Inc. (SGI 社) の O2, HP (Hewlett Packard) 社のワークステーションが設置されていたこともあります。また、研究室には Windows や Linux 等の様々な計算機があるでしょう。ワークステーション自身やそれに組み込まれているオペレーティングシステムが異なると、コマンドを置くディレクトリの構成やコマンドの名前、勿論コマンド自体も異なります。

例えば Solaris11 用にコンパイルされた Emacs を iMac で実行することはできません。それは皆さんが作ったプログラムでも同じことです。

また、機種によりディレクトリの構成などが異なる場合があり、異なる端末でも共通のホームディレクトリを使用するために、皆さんに配られているシェル環境設定ファイルである \$HOME/.profile (sh 用の設定ファイル), \$HOME/.bashrc (bash 用の設定ファイル), \$HOME/.cshrc (csh 用の設定ファイル) 等にはログインした機械の違いを認識して設定を調整するようプログラムされています。

26.1.3 更新

しばらくはあまり細かいことを気にする必要はありませんが、機械が違えば色々何かと違うものだと言うことを認識してください。それから、管理者が異なれば、やはりディレクトリの配置の方法を始め、ツールの使用方法なども何かと異なるということも事実です。

会津大学では、毎年少しずつ計算機の構成が更新されていきますから、変更に伴う情報センターからのアナウンス (<http://web-int.u-aizu.ac.jp/labs/istc/>) や ISTC のホームページ <http://web-int.u-aizu.ac.jp/labs/istc/isp/release/release.html> を注意して見て、変化に対応していく必要があります。

の様に出力されます。左から IP アドレス、ホスト名(ニックネーム)、別名(エイリアス)，コメントの順に書いてあります。IP とは Internet Protocol のことです。会津大学では教室の計算機のホスト名は

26.1.4 NFS

ホームディレクトリや共有ディレクトリなどは機種に拘らず、どの端末でも利用できるようになります。df コマンドを用いれば、以下が表示されます。第一フィールドがファイルシステム名(バイス名)です。その名前にコロン(:)があるものが NFS を利用してネットワーク経由で提供されています。最終フィールドがマウント先です。

% rsh stdrsrv1121 'df -h'	Filesystem	Size	Used	Avail	Use%	Mounted on
	rpool/ROOT/solaris-4	397G	4.8G	392G	2%	/
	Swap	42G	1.7M	42G	1%	/system/volatile
	/usr/lib/libc/libc_hwcap1.so.1	397G	4.8M	392G	2%	/lib/libc.so.1
	rpool/ROOT/solaris-4/var	400G	7.9G	392G	2%	/var
	rpool/ROOT/solaris-4/var/tmp	43G	178M	42G	1%	/tmp
	rpool/export	392G	93M	392G	1%	/var/tmp
	rpool/export/home	392G	32K	392G	1%	/export
	rpool/local	392G	31K	392G	1%	/export/home
	/dev/dsk/c4t0d0p0:1	411G	19G	392G	5%	/usr/local
	stdfsrv101:/vol/vol4/istc/istc	3.5G	2.1G	1.5G	59%	/media/ORACLE_SSM-1
	stdfsrv0812:/vol/vol1/home4/m51611xx	296G	130G	79G	61%	/home/grd/m51611xx
	stdfsrv0812:/vol/vol2/course	60G	49G	12G	81%	/home/course
	stdfsrv101:/vol/vol4/web-int-course	5.0G	3.9G	1.2G	78%	/home/web-int/course

この場合には/home/grd/m51611xx は stdfsrv0812 という名前のファイルサーバからネットワークを経由して提供されているということが確認できます。

26.1.5 リモートでの利用

分散化されている計算機資源をうまく利用する場合にはリモートで接続する命令を使用します。講義資料で既に触れてありますが、rlogin, telnet, rsh, ssh などがあります。

26.1.5.1 端末内で作業する

リモートでログインした場合、その端末(xterm, gnome-terminal 等)で実行する命令はすべてログイン先のワークステーションで実行されます。また、ログインした先でシェルの環境変数がどの様になっているかはシェル環境設定ファイルの中の設定次第です。設定を確かめるためにはシェルに関係なく env, printenv を利用して確認できます。bash の場合は export を引数無しで実行すると、確認できます。csh の場合は setenv を引数無しで実行する

```
std4dc1fa-fujitu51: rsh std4dc2
Last login: Fri Jan 15 12:35:21 from std4dc2
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
```

と、確認できます。
(勿論端末としての設定に異常があれば、困難な事態が生じることもあります。) 例えば、

Emacs を端末の中で使用したい場合には **emacs -nw** の様にノーウィンドウオ

プション(No Window)をつけて実行すること可能です。(この場合 & をつけてはいけません。つけた場合には、fg コマンドでフォアグラウンドに戻しましょう) この場合の Emacs は X 環境に対応しないので、通常のものとはマウスの操作の点で異なります。
(注意: Mac OS X にインストールされている Emacs を端末内で利用する場合には日本語表示のために環境変数 LANG を ja_JP.UTF-8 などにする必要があります。)

.rhosts について
.rhosts と言う名前のファイルが適切に設定してあると、非常に便利にネットワークで接続されている計算機資源を利用することができる様になります。.rhosts に各行にリストしてあるホスト名とユーザ名の組合せに対してパスワードの入力無しでログインを許可することになります。したがって、

rsh stdrsrv1 top

などという命令が実行可能です。ただし、不適切な設定は非常に危険なセキュリティホールとなるので、多くの研究所や大学でも .rhosts のサービスを利用不可能に設定しています。入学の時点で配布されている .rhosts や標準設定参照用の ~stdskel/.rhosts は学内の便用のために設定していますが、その内容の変更はユーザー自身の管理義務と責任の範囲で行なうことになります。また、.rhosts の内容は他人に見られてはいけないので、適正なアクセス許可(-rw-----)の設定が必要です。

26.1.5.2 X アプリケーションを利用する

X 用のアプリケーションをリモートログインで利用する場合には、X サーバ、つまり目的前にあるワークステーション A で X ウィンドウシステムが動作している必要があります。リモートログイン先 B で X ウィンドウシステムが動作している必要はありません。

次にシェルの環境変数 DISPLAY を正しく設定します。A 内で、

```
% env |grep DISPLAY
DISPLAY=:5.0
```

を実行し、X サーバ側のスクリーン番号の値を取得します。上の例の様に、5.0 と表示された場合には B 内で、その値を使用して

```
setenv DISPLAY A のホスト名:5.0
DISPLAY=:5.0
```

を実行します。

```
std4dc1[a-fujitsu51]: rsh std1dc1
Last Login: Mon Jan 18 11:40:28 from std4dc1
Sun Microsystems Inc., SunOS 5.10 Generic January 2005
std4dc2[a-fujitsu51]: kterm
kterm: Xt error: Can't open display:
std4dc2[a-fujitsu52]: setenv DISPLAY std4dc1:0.
[1] 15915
std4dc2[a-fujitsu54]: ■
```

さらに、Xサーバではリモートログイン先のワークステーションから送られてくるX用のアプリケーションの表示を可能にするために登録しなければなりません。xhost と言う命令を XサーバA の端末で引数無しで実行すると現在の設定が表示されます。この場合、

Xサーバ A 側の端末で、
xhost +(Bのホスト名)

を実行すれば B が xhost の

リストに追加され、BからのXアプリケーションを A の X ウィンドウシステム内で表示することができます。以上の2つの設定に成功したら、リモートログイン先 B の端末で、

xeyes & を実行すれば、xeyes の窓がこちらのディスプレーに表示され、利用でき

ることになります。この場合 xeyes のプログラム本体は B のワークステーションで実行されていて、その表示の部分だけをこちらのワークステーション A が(Xサーバとして)肩代りをしているということになります。

xhost + を実行してしまうとどのホストからの X アプリケーションに対しても表示を許可したことになってしまいます。このような不用意な設定は世界中の誰からでもキーボード入力を盗み見るプログラムを動かされたりするなど、セキュリティの上で非常に危険な状態なので、決してしてはいけません。

26.1.5.4 リモートで利用する場合のマナー

シングルユーザ、つまり自分一人で計算機資源を利用する際に特にマナーに気をつける必要はありませんが、マルチユーザーで利用する場合にはマナーを守らなければ多くの人と仲良く協力していくことはできません。計算機資源利用については、通常の人間社会で守られるべきマナー、特に公共物を利用する場合のマナーを適用して下さい。利用規定 <http://web-int.u-aizu.ac.jp/labs/istc/ipc/rules/regulation.html> には他の利用者に対する著しい迷惑行為についてはアカウント利用を一時的に停止すると明記されています。

- ・なるべく利用されていないところを選ぶ。
- ・授業が行なわれている場合には利用を避ける。

26.1.5.5 学外からのアクセス

学外から学内にアクセスするための入口は sshgate.u-aizu.ac.jp です。一旦ここにアクセスをしてそこから学内の他のコンピュータにアクセスできます。情報センターの技術情報を参考してください。
<http://web-int.u-aizu.ac.jp/labs/istc/ipc/topic/sshgate/sshgate.html>

sshgate.u-aizu.ac.jp をトンネルとして使用し学内のコンピュータにアクセスできます。

26.1.6 ネットワークを監視するためのツール

ネットワークの状態を監視するツールを紹介します。Windows や Mac でも同様のツールがあります。

traceroute	ネットワークの経路情報を表示する。
ping	ネットワークホストに ICMP_ECHO_REQUEST を送り、反応を表示する。
nslookup	DNS サーバに問い合わせをする。
netstat	show network status
arp	address resolution display and control

```
stdrsv1121 % xhost +std5dc4
std5dc4 being added to access control list
```

```
stdrsv1121 % ssh -X std5dc4 'env DISPLAY=stdrsv1121:16.0 xterm'
```

26.2 プロセス管理

演習 6.1.4 (p.112) の内容にジョブの管理がありました。今回はその復習です。ネットワークを利用してリモートアクセスをした計算機の利用状況を把握することはとても大事です。

F	<sys/proc.h> の中の処理に関係したフラッグ(旗)
S	プロセスの状態。O=running, S=sleeping, R=runnable, Z=zombie, T=stopped
UID	プロセスの所有者名。
PID	プロセス番号。
PPID	親プロセスのプロセス番号。
C	短期間の CPU 使用因子(スケジュールクラス)
CLS	スケジュールクラス
PRI	プロセス優先度
NI	プロセススケジュール値
ADDR	プロセスのメモリーアドレス。
SZ	データとスタックセグメントの合計サイズ
WCHAN	プロセスが待機しているイベント名
STIME	開始当日ならプロセス生成時刻。それ以外は生成日。
TTY	プロセスを制御する端末。
TIME	プロセスのCPU 使用時間の合計。

表 26.1. プロセスの属性

時間当たりの平均で CPU の利用度が大きいプロセスを順番にリスト表示する。現在動作しているプロセスを表示する。使用例 (/usr/bin/ps -elf)

現在動作しているプロセスを終了させる。C シェルから起動したジョブのリストと状態を表示する。カレントジョブあるいは [%job] で指定されるジョブをフォアグラウンドにする。あるいは[%job] で指定されるジョブをバックグラウンドにする。

- 個人的な連絡は個人宛の電子メールを利用して下さい。
また、個人的なメールのやりとりを相手の知らない場所で公開することや、相手に確認しないで公開するのは良くないことです。
- 学生部からのアナウンスはニュース (u-aizu.announce.gakuseib) で参照できます。

- ホームディレクトリは読んで字のごとく、その人の家の様なものです。ホームディレクトリ以下にあるものは読み出し許可がおりていても実際に本人に確かめてからでなければ、コピーをしたり覗いたりしてはいけません。訴えられた場合には不法侵入、覗きとして処罰される可能性があります。
- 会津大学内では公開されているファイルやディレクトリは全てコピーや閲覧ができます。

26.3.2 ウェブプラウザの利用

- コンピュータに接続されている (NFS も含む) ファイルシステムのファイルは

file://
で参照できます。他のユーザーのホームディレクトリはアクセスして始めて NFS によりマウントされるので、ブラウザで参照する前に terminal 等の端末内で

ls ~username

- という意味であることが解ります。次に主な表示項目の説明を表 26.1 にしました。
- | | |
|------------|---|
| -a | 全てのプロセスをリストする。 |
| -e | 現在動作中のプロセスを表示する。 |
| -l | 長い詳細な表示。F S UID PID PPID C PRI NI ADDR SZ |
| -f | コマンドの完全な表示。UID PID PPID C STIME TTY TIME |
| -u uidlist | CMD を表示する。
uidlist に含まれる ユーザが実行しているプロセスをリストする。 |

- 火狐 (firefox) を実行し、その後ブラウザで表示すると良いでしょう。あるいは、firefox を起動する際の引数として表示したいユーザーを指定すると良いでしょう。

```
firefox ~a-fujitsu/education/literacy/
```

26.3.3 会津大学共用 WWW サーバの利用

会津大学共用 WWW サーバの利用は
http://web-int.u-aizu.ac.jp/official/committee/pd/wsa/application_jp.html
 に記載されている案内に従い、利用申請の提出が前提条件です。会津大学 WWW ページ運用指針、WWW サーバの利用方法を熟読してください。ホームページの公開は <http://web-int.u-aizu.ac.jp/labs/istc/ipc/topic/web-server/web-server.html> を参考してください。

26.3.3.1 web-int.u-aizu.ac.jp

web-int.u-aizu.ac.jp は会津大学内向けのウェブサーバです。個人のホームページは

<http://web-int.u-aizu.ac.jp/~username/>

で参照できます。ホームページを掲載するためには webfs-int.u-aizu.ac.jp に FTP を利用してファイルをアップロードします。各自がコンテンツ格納サーバ (webfs-int.u-aizu.ac.jp) 上のホームディレクトリ配下の public_html 内にコンテンツを置きます。

26.3.3.2 www.u-aizu.ac.jp

www.u-aizu.ac.jp は外向けのウェブサーバです。個人のホームページは

<http://www.u-aizu.ac.jp/~username/>

で参照できます。ホームページを掲載するためには webfs-ext.u-aizu.ac.jp に FTP を利用してファイルをアップロードします。各自がコンテンツ格納サーバ (webfs-ext.u-aizu.ac.jp) 上のホームディレクトリ配下の public_html 内にコンテンツを置きます。

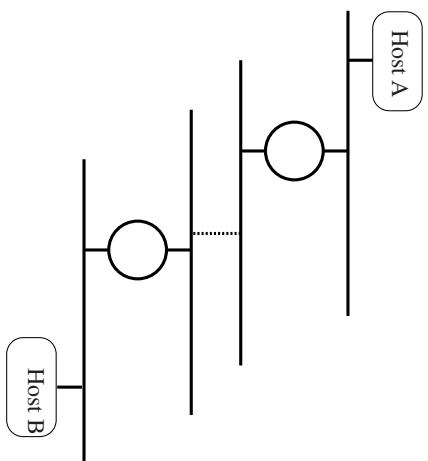
26.3.3.3 注意

ウェブブラウザを使うとマウスの操作だけで表示ディレクトリを移動できるのでウツカリ内容を覗いてしまうこともあるかも知れないでの気をつけ下さい。
 (/home/professor/a-fujitu の中の読み込み可能なものは見ても構いません。)

stdrv1.u-aizu.ac.jp が接続しているネットワーク 47 番 (163.143.11.0) と
 aplisvl.u-aizu.ac.jp が接続しているネットワーク 3 番 (163.143.3.0) の接続を
 traceroute 等を用いて調査せよ。結果をファイル \$HOME/literacy/net.txt に報告せよ。

以下にサンプルを示します。

```
\documentclass[a4paper]{article}
\title{} % 提案するサービス名
\author{} % 名前 学籍番号
\date{} % 作成日付
\begin{document}
\maketitle
\abstract{} % 概要
\section{利用目的} % 誰が利用し、また利用を通してどの様な効果を期待するかなど
\section{情報システム運用形態} % 誰がどんな方法で運用するか
\section{技術提案} % 情報サービスを実現するための技術提案。
\end{document}
```



26.5 宿題 26

リテラシーのホームページを見ると Web がどんなに簡単に情報を提供できるシステムか感じられたかもしれません。ネットワークを利用してどんな情報提供を行なったら便利でしようか。Web に限らず、学内のネットワークを利用した情報サービスの企画書を [TeX](#) で作成し提出しなさい。(項目を追加してもよろしいです。) 実現可能かどうかはあまり考える必要はありませんから、レポート作成の練習のつもりで、勝手放題に気ままに書いてみて下さい。

ファイル名は info-sys.tex とし、info-sys.tex, info-sys.dvi はディレクトリ \$HOME/literacy に保存すること。以下のテンプレートにしたがって企画書を作成して下さい。



ハンドアウト公開サービス(サンプル2)

藤津 明
2012年5月28日

Abstract

ハンドアウトを公開し、授業の枠を越えた情報収集を可能にする。

1. 利用目的 優れたハンドアウトを公開し、授業の枠を越えた情報収集を可能にするれば、その気になった時にいつでも集中的に勉強できる。また、教員間の情報交換もスムーズになる。
2. 情報システム運用形態 とりあえず、学内専用のページを作成し、そこで公開サービスを実施する。ページの管理者を適当な教員、または教員グループにお願いする。
3. 技術提案 普通の www で結構です。

第27章 シェルプログラミング(2), 表計算(OpenOffice Spreadsheet)

27.3 シェルプログラミング

27.3.1 作ってみましょう

各自の literacy に cd して下さい。そこでシェルスクリプトを作ってみましょう。今回は myscore というファイルを編集します。

27.1 シェルの便利な使い方 - その2

皆さんのがどの様な設定でシェル環境を利用しているかを確認することは重要です。

set	シェル変数の一覧を表示する。
env	環境変数の一覧を表示する。
alias	alias 命令で設定した内容を表示する。

プロセスを管理することは重要です。

ps	現在動作しているプロセスを表示する。
top	動作中のプロセスの中で cpu の使用率の高いものを表示する。
kill PID-number	PID-number で指定されるプロセスを強制的に終了する。

nice 10 commandname	commandname を実行する際の優先順位をコントロールする。
----------------------------	-----------------------------------

(注意) cpu の使用率は固定されているものではありません。ランニングプロセス同士でお互いに cpu の使用をシェアします。90%の様になっている場合はランニングプロセスがたった一つである場合が多いです。nice の値によってシェアの仕方はコントロールできます。90%という数字が意味するのはたまたまその瞬間に平均としてそれだけ使用したということです。他のプロセスに cpu を使わせないと言うことではありません。

27.2 前回の復習

前回のプリントに書かれていた例題は理解できたでしょうか。実際に作って、動作させることに成功したらプログラムの実行の流れや意味を理解することが次へのステップに繰り返します。
前回うまくいかなかつた人は、今日のこの時間を利用して一度トライしてみて下さい。
周囲にできた人がいるようでしたらその人にできないところを相談してみましょう。
相談を受けた人は自分の理解の助けにもなりますから、なるべく丁寧に教えてあげるよう努力して下さい。以上が終了してしまった人は次の節にチャレンジしてみましょう。

27.3.2 チャレンジコース

- bash では read を使用して、標準入力から入力される文字列を変数 number に代入します。
- このプログラムは リテラシー I の提出物の評価が表示されます。

```
#!/usr/local-gnu/bin/bash
sum=0
while [ "$1" != "" ]; do
    let sum+=$1
done
echo $sum
```

```
#!/bin/csh -f
@ sum = 0
while ($#argv)
    @ sum = $sum + $argv[1]
shift
end
echo $sum
```

実行すれば $1+2+3+4+5+6 (=21)$ を計算してくれるスクリプト plus を作ってみましょう。

csh のスクリプトでは \$argv はコマンド引数を格納する配列変数です。 \$#argv は配列の大きさを表します。 shift は \$argv を左にシフトして、n 番目の要素を $(n-1)$ 番目に移します。

今回の演習まで、大変良くがんばってこれた」という人も、「良くわからない」という人もいると思いますが、これまでの内容が十分に理解できて、自分の手足の様にワークステーションが使えるようになることを期待しています。

27.4 課題 27

- 上記の例を参考に平均を計算するスクリプト `avrg` を書いてみましょう。
- `bash` では割り算は

```
let avg=sum/num
```

の様に書きます。bc 等の外部のコマンドを使用すると小数の結果を得ることができます。
`csh` では割り算は

```
q=av=$sum./$num
```

の様に書きます。/ の前後には空白が必要です。

27.5 OpenOffice

Microsoft 社の `Office` ツールは現在非常にポピュラーで多くの人々が利用しています。
 その結果、`Office` ツールに固有のファイル形式で保存されたファイルを扱わなければ困
 ってしまう状況が多くの場面で発生します。

- MS-Word の文書ファイルを渡された。
- MS-PowerPoint のファイルを渡された。
- MS-Excel のファイルを渡された。

などの状況があります。特にメールに添付された場合などはちょっと困ります。運良く `MS-Office`
 を持っていたとしても最新のバージョンでなければ新しいファイル形式のファイルに対応
 は対応不能です。こんな困った状況で、どの様に対処すれば良いのでしょうか?

- 最新版の `MS-Office` を購入して使う。… これは大変コストのかかる方法です。な
 ぜなら最新版の `MS-Office` は、ほとんどの場合に、最新の `Windows` でしか快適に
 動作しないからです。しかも、最新の `Windows` は最新のハードウェアを要求します。
- `MS-Office` 互換の他社のソフトウェアを購入して使う。… 安くて、ちゃんと動作す
 るものを探すのが一苦労。
- 合法的に無料で取得できる `MS-Office` 互換のソフトウェアをインストールして使う。
 … ファイルの破壊や動作の不安定性について補償無し。充分承知して使うならこれ。
- ファイルをテキスト形式に変換したもの再度渡して貰うようにお願いする。上司や
 顧客にお願いするのは失礼かもしれません。人間関係が一番フラストレーションが溜
 るものです。
- あきらめる。… これでは解決になりません。

の中から選ぶことになるでしょう。
`MS-Office` 互換のソフトウェアは幾つか存在します。今回の演習ではその中の一つであ
 る `OpenOffice` を紹介します。

27.5.1 OpenDocument Format

`OpenOffice` で使われているファイル形式は標準化団体の `OASIS` が規格を標準化した
`OpenDocument Format (ODF)` というものです。これはソフトウェアメーカーには依存しな
 い、独立した標準規格なので、公的機関で使用される文書の標準形式として採用されること
 になるかもしません。また、各ベンダーも `ODF` を採用したり、`ODF` への変換ツールを提
 供する予定の様です。

- `OASIS`

<http://www.oasis-open.org/>

`ODF` に対応している `Office` ツールを調査してみましょう。

27.6 宿題 27

`/home/course/lit1/pub/literacy/` に置いてあるファイル
`math.csv`, `english.csv`, `physics.csv` を `OpenOffice` を利用して集計し、以下を求め
 よ。

- 各科目の平均値と分散。
- 3 科目の合計点の平均値と分散。
- 合計点の分布グラフ。(ヒストグラムでなくとも良い)

それぞれの問い合わせに対する回答を `soffice.pdf` にレポートせよ。ただし、以下の指示に従つ
 てください。

- `soffice.pdf` は `soffice` の `Export` の機能を使用して生成する。

第28章 Make, スペルチェック, プレゼンテーション (OpenOffice Presentation), MagicPoint

28.2 課題 28 GNU-HELLO

多くのプログラムパッケージはプログラムソースを tar でまとめて、gzip 等の圧縮ツールで圧縮されて配布される。ここでは、プログラムパッケージ GNU-HELLO をインストールする作業を実行し、標準的な手順を体験してみましょう。以下の指示に従って GNU-HELLO をインストールせよ。作業は \$HOME/literacy 以下で行います。

28.1 make

巨大なプログラムなどの部分を再コンパイルするべきかなどを自動的に判別して指定された処理を行う道具です。多くのソフトウェアはソースコードで配布され、Makefile に書かれた手順に従ってインストールできる様になっています。

Makefile の書き方は基本的に以下の通りです。target の前は空白があってはいけません。procedure の行の最初の文字は必ずタブでなければなりません。

```
+ target: sources
<tab>procedure@0
<tab>procedure1
```

変数を使用することもできます。以下は \LaTeX の処理をする Makefile の例です。

```
LATEX = platex
DVIPDF = dvipdfmx

all: tex

tex: beamer@.tex
      $(LATEX) beamer@.tex
      $(LATEX) beamer@.tex
      $(DVIPDF) beamer@.dvi beamer@.pdf

clean:
      rm *.dvi *.pdf *.aux *.log
```

28.3 Spell Check

英語の文書を書く場合に綴りを間違えることがしばしばあります。それは間違った綴りを思い込んで覚えていたり、指が他のキーを押したり、左手の指と右手の指の押す順番が微妙に入れ替わったり、等様々な理由で生じるもので、そこで綴り間違いを発見し訂正する作業が発生します。コンピュータを用いてその作業をするととても助かることがあります。今回の演習ではスペルチェックを紹介します。

28.3.1 spell

以下の命令を実行すると ConstitutionOfJapan.txt ファイル中にあるワードのスペルを調べ、スペルミスをレポートしてくれます。

```
spell /home/course/lit1/pub/literacy/ConstitutionOfJapan.txt
```

を実行すると clean に対応する処理が実行されます。引数無しで make を実行すると all に対応する処理が実行されます。

make clean

28.3.2 aspell

以下の命令を実行すると ConstitutionOfJapan.txt ファイル中にあるワードのスペルを調べます。spell コマンドとは異なり、こちらは、インターラクティブに操作できます。

```
aspell -c /home/course/lit1/pub/literacy/ConstitutionOfJapan.txt
```

28.3.3 Emacsでispell

Emacsのバッファ中でも ispellを使用することができます。

- M-x ispell-buffer … バッファ全体をチェックする。
- M-x ispell-region … 領域をチェックする。
- M-x ispell-word … カーソル位置の単語をチェックする。

28.3.4 Macの辞書

iMacには辞書があるので利用できます。また発音も聞くことができます。ターミナルで

say "I love you."

ヒタイフしてみましょう。

28.4 プrezentーション

新しい商品の説明や、企画イベントの説明、研究発表やセミナーでの報告、あるいは卒業論文の発表などの機会に、解りやすく上手に説明することが求められます。このような場合には、論文、報告書、企画書等の詳細な文章をそのまま提示することはできないので、理解しやすいように、要所を大雑把にまとめた概要のスライドを用いて発表することができます。プレゼンテーションスキルは企業や研究における相互理解を助ける必要な技術です。

プレゼンテーションスキルには、声の出し方や間の取り方、速度、姿勢、視線、表情、身振りなどのコミュニケーションや表現の技術が含まれますが、これはリテラシーでは教えるのは困難なので扱いません。リテラシーでは、プレゼンテーションに利用されるコンピュータの利用法を紹介します。すなわち、スライドの作り方と表示の方法です。

スライドを作り表示するツールとして多く利用されているのがMicrosoft社のPowerpointです。Powerpointの利用方法は研究室や個人のパソコンで習得できます。他にも多くのツールがあります。LaTeXではprosperやbeamerを用いてPDFのスライドを作成できます。Adobe Readerの場合はFullスクリーン表示をすることで、プレゼンテーションを行うことができます。

X Window SystemではMagicpointもプレゼンテーションに利用可能です。OpenOfficeはMicrosoftのOffice(MS-Office)とほぼ同等の機能を提供する互換性の高いソフトウェアです。

28.4.1 sofficeの起動

環境変数 LANG を日本語 (=ja_JP.UTF-8) に設定して soffice を起動するとメニューが日本語になります。

```
env LANG=ja_JP.UTF-8 soffice
```

28.4.2 presentation 利用開始

メニューから [ファイル(F)] → [新規作成(N)] → [プレゼンテーション(P)] を選びます。



で白紙のプレゼンテーション(F)を選び完了(C)を押すと、白紙の状態からスライドを作成できます。メニューから挿入(I)→ページ(E)や挿入(I)→ページの複製(D)を選ぶと新しいページを挿入できます。

テンプレートから(F)を選び次へ(N)を押した場合は、テンプレートを利用したスライドの作成を実行できます。

28.4.3 作業モード

作業の目的別に、以下の作業モードがあります。

- 図形描画モード… ページの作成とページの編集ができる。
- アウトラインモード… プrezentーションの各スライドにテーマを入力します。スライドを整理したり、項目のアウトラインレベルを変更したりします。
- スライドモード… スライドをライトボックスに並べたように縮小表示します。(サムネイルによるスライドの一覧を表示)
- ノートモード… スライドにノートを追加します。スライドショーの実行では、このノートは表示されません。
- ハンドアウトモード… ハンドアウトモードに切り替えます。スライドを縮小した形で1ページに並べられます。ページに印刷可能なスライド数を変更するには、書式 → ページレイアウトの変更を選択します。(職業に配布するスライドをテキストも含めて表示)

28.4.4 プレゼンテーションの公開

作成したスライドは、画面上で表示するだけではなく、ハンドアウトや HTML 形式ドキュメントとして出力することもできます。

28.4.5 スライドショウ

自動実行させることも手動操作することもできます。

28.5 MagicPoint

X Window System で動作するプレゼンテーションツールの一つである MagicPoint を体験してみよう。

- MagicPoint のおおまかな機能のデモ。
mgp /usr/local/lib/X11/mgp/sample/sample-jp.mgp
- MagicPoint の操作方法の説明。
mgp /usr/local/lib/X11/mgp/sample/tutorial-jp.mgp
- 背景のデモ。グラデーション。
mgp /usr/local/lib/X11/mgp/sample/gradation-jp.mgp
- mpg2ps を利用すると MagicPoint のファイルを PostScript に変換することができます。
mpg2ps -f a.ps /usr/local/lib/X11/mgp/sample/gradation-jp.mgp

28.6 宿題 28

1. /home/course/lit1/pub/literacy/ に置いてあるファイル einstein.odp, sakamoto.odp 等を \$HOME/literacy に複製し OpenOffice Presentation を利用して表示せよ。
2. これらを参考にプレゼンテーションを作成せよ。
3. スライドを PDF ファイルに export して presentation.pdf として保存せよ。

第29章 トラブル対処法

29.1.4 ファイルを見ることができない

`cat`, `more`, `less`などの命令ではテキストファイルでないと正常に表示されません。
`file` コマンドで確認しましょう。あるいは、読み出しの許可がないかも知れません。`ls -l` コマンドでパーミッションを確認しましょう。

29.1 トラブル対処法

コンピュータがトラブルの状態になった場合に、ユーザーの手で復旧できる場合とできない場合があります。可能な場合は自分で対処する方が良いでしょう。以下で可能な場合をいくつか挙げているので理解して下さい。また個人的にはトラブルと感じてもそうでない場合もあります。パニックに陥らずに原因を可能な範囲で把握する努力が重要です。

29.1.1 キーボードで入力ができない

ありそうな事態はキーボードの接続が正常でないこと、つまり、端子が抜けていたり、キーボードを制御するソフトウェア（ドライバ）が装置を認識出来なくなっている場合です。USB 接続のキーボード（やマウス）の場合は USB 端子を差し直してみましょう。

29.1.2 画面がおかしい

`xlock` が動作していることがあります。または Gnome Desktop のスクリーンセーバー（画面の焼付けを防止するためのツール）が動作している場合もあります。どちらもキーボードを叩いてください。それらの場合は、パスワードの入力を解除できます。どちらもキーボードを叩いてください。それらの場合は、パスワードの入力を正常に受け付けない状態になります。また、何かの原因で `X` ウィンドウが壊れている場合はキーボードの入力を正常に受け付けない状態になります。そんな場合には異常な動作をしている `X` を終了します。`X` を終了するには、他の端末からリモートでそのホスト（Sunray の場合は Sunray サーバ）にログインし、`X` のプロセスを `kill` します。プロセス ID の取得は以下の様に `ps` コマンドを使用します。

```
% ps -efl |grep username |grep X
```

注意 ユーザは `kill` コマンドを使って自分の所有するプロセスを終了することができます。他のプロセスは終了しません。必要な場合は暴走したプロセスを `kill` して下さい。

29.1.3 ネットワークの具合が悪い

`Network` は混雑てくると反応が悪くなるものです。しかし、あまり遅くなるようだと問題です。イーサネットの接続部分を点検してみましょう。自分の方と相手の方の両方を見る必要があります。また、相手方が動作していないという状況も考えられます。

29.1.5 印刷が暴走している

これはポストスクリプトファイルをテキストファイルとして印刷した場合に起こる症状です。資源の有効利用のため直ちにジョブをキャンセルする必要があります。

`lprm_-Pprinter-name.job-number`

印刷の命令を実行したのに印刷しない (`lpq` のジョブリストにも登録されない)。または、登録されたが自動的にジョブがキャンセルされる) 場合があります。テキストファイルや `EPS` の場合は `dvi` ファイルをそのまま印刷しようとしました場合などがそうです。テキストファイルの場合は `k2ps` などを利用してポストスクリプトファイルに変換してから印刷しましょう。`EPS` の `dvi` ファイルの場合は `dvips` を利用してポストスクリプトファイルに変換してから印刷しましょう。

29.1.6 コマンドが動作しない

「隣の人と同じことをやっているのに...」などという場合があります。コマンドが動作しない原因はコマンドの検索パスが足りない場合やコマンドの実行許可がない場合があります。
`which` 命令で探せる場合は検索パスに登録されています。`.bashrc` 等のシェル環境設定ファイルの中身を良く見てどのように PATH が設定されているかを確認しておくことが大切です。`echo $PATH` を実行して表示することもできます。
また、自作のプログラムやスクリプトが動作しない場合もあります。まさかと思っても、ファイルの実行許可を付け忘れている場合があります。
いずれにせよ、表示されるメッセージを注意深く見ることです。

29.1.7 ファイルを誤って消してしまった

ファイルを消すと元に戻らないことは肝に命じておきましょう。情報センターではホームディレクトリの全ての情報を定期的にバックアップを取っています。バックアップは2,3日のスケジュールなので運良くバックアップされていたらリストア（復旧）できます。ファイルのリストアは大変手間のかかる作業ですから、情報センターの業務スタッフに迷惑です。個人的なバックアップのためにリムーバブルメディアを利用します。

29.1.8 標準環境設定ファイル

会津大学の標準環境設定ファイルは `~stdskel` に置いてあります。

`.profile`, `.bashrc`, `.emacs.d/init.el`, `.Xresources`, `xinitrc.mac`, `.logout`

です。もしも、自分の環境設定ファイルを変更して異常が発生した場合はこれらを自分のホームディレクトリにコピーすれば入学した当初の状態になるはずです。詳しくは

`~stdskel/README`

を参照してください。

会津大学では毎年どこかの演習室の計算機システムが更新されます。その様な場合には環境設定ファイルを更新する必要があります。新しく導入された計算機システムが利用できない場合には`~stdskel` にあるファイルを参考にしましょう。

29.1.8.1 Mac OS X の標準環境

iLab や演習室 5, 6 では Mac OS X が利用できますが、利用者による操作ミスや OS のアップグレードにより、ユーザーの設定ファイルが適切な状態で無い場合が発生します。この場合は `/usr/local/bin/macenvcp` を実行した後に再 `login` します。また、利用環境を初期化するために `/usr/local/bin/initmac` を利用できます。

29.1.9 ISTC FAQ

ISTC のホームページ内 (<http://web-int.u-aizu.ac.jp/labs/istc/ ipc/faq/faq.html>) に FAQ のサービス (図 29.1) があります。技術的な問題を解決できるかも知れません。iLAB の Mac OS X 用の FAQ もあるので利用してください。

29.2 トラブルレポートの方法

自らの手でトラブルを解決し復旧できない場合は、トラブルを直ちに報告し、情報センターのサービス (<http://web-int.u-aizu.ac.jp/labs/istc/ ipc/faq/faq.html>) の Mac OS X 用の FAQ もあるので利用してください。

29.2.2 緊急の場合

計算機を使えない場合や緊急の場合には電話を利用して下さい。電話番号は端末に貼られたシールに記載されています。2.5 節 センターサービス時間を参照して下さい。情報センターアクセスの手引は <http://web-int.u-aizu.ac.jp/labs/istc/ ipc/service/document.html> からも参照できます。

といった内容があれば、適切な処置を素早く実行できます。



図 29.1. ISTC FAQ

29.2.1 SSB(System support Base)

情報センター（情報処理センター）の出先窓口（システムサポートベース）として障害、問い合わせ対応 及び、申請書受理、機器貸出サービスなどを行っています。研究棟 207-E の部屋に設置されています。（アカウンタロックの解除などの対応は SSB で行われています。）ただしサービスは平日の 9:00～17:00 に限られます。

<http://web-int.u-aizu.ac.jp/labs/istc/ ipc/ssb/ssb.html>

SSB に行き、サポートを受ける場合は学生カードが必要です。また、計算機のトラブルに 対応するため、計算機のホスト名 (`std2tc40` など) が必要ですから、貼られているシールに書いてあるホスト名をスタッフに伝えましょう。演習室からの電話の際もホスト名を伝えましょう。

(<http://web-int.u-aizu.ac.jp/labs/istc/ ipc/ssb/mail-form.html> を参考にしてください。)

- いつ (when),
- どのマシンで (where),
- 誰が使用中に (who),
- どの作業中に (what),
- なぜそうなったか (why),
- このようなメッセージが出てるか、どのような状況か,... (how)

といった内容があれば、適切な処置を素早く実行できます。

29.3 課題 29

第29.1節のトラブル対処法の項目と今まで個人的に遭遇したトラブル項目を \TeX の `tutorial` 環境や `tabularx` 環境を利用して以下のような表にしましょう。ファイル名を `trouble.tex` とし、`trouble.pdf` も作成して下さい。

トラブルの種類	トラブルの状態	原因	対処方法
Firefox	起動できない	不正終了した	ロックファイルを削除

29.4 HTML (Web ページを作る)

29.4.1 HTML の書き方

リテラシーのホームページ (<http://web-int.u-aizu.ac.jp/course/lit1/index.html>) は World Wide Web のためのマークアップ(印を付ける)言語、すなわち HTML(Hyper Text Markup Language)で書いてあります。色のついている部分をクリックすると、その部分に指定してある操作が実行されます。HTML はリテラシー I では詳しくは学習しません。ウェブラウザのページソースを見るための機能を利用すれば表示されているページのソースコードを読むことができます。それを真似をすればある程度は書けるようになります。HTML の仕様(標準的な書き方や機能を指定することです。)は利用され始めてから今日まで変化しました。またウェブラウザによっては標準仕様と合わない動作をするものもあります。HTML の仕様のドキュメントは <http://www.w3.org/> にあります。

29.4.2 HTML の練習

学内では Web のページを作り公開することで情報のプロバイダ(提供者)になることができます。以下のサンプルを参考にして HTML のファイルをファイル名 `literacy.html` と置いてください。

```
<html>
<head>
<title> sXXXXXX's literacy course page </title>
<body>
<h1> Welcome to my page </h1>My works.<br>
<ol>
<li><a href="kadai000.ps"> kadai000.ps </a> </li>
<li><a href="introduction.dvi"> introduction </a></li>
</ol>
<br>
Go to
<a href="http://web-int.u-aizu.ac.jp/course/lit1/">2017 Literacy I page</a>
<br>
</body>
</html>
<address> last updated by sXXXXXX June **, 2017 </address>
```

29.5 宿題 29(web-int)

会津大学の学内向けウェブサーバ (`web-int.u-aizu.ac.jp`) に自分のホームページ (<http://web-int.u-aizu.ac.jp/~username/literacy.html>) を載せるためには `webfs-int.u-aizu.ac.jp` に FTP を利用して、ファイルを転送する必要があります。練習で作成した `literacy.html` を `webfs-int.u-aizu.ac.jp` に 転送せよ。
 ファイルをアップロードした後、一定の時間が経過すると、ブラウザで確認します。

29.6 Maxima の利用

Maxima¹は1960年代の後期にMITで開発された伝説的なコンピュータ代数システムであった Macsyma の子孫です。また唯一、一般に公開され続け、かつ活動的なユーザコミュニティを持つ、オープンソースのソフトウェアです。Macsyma は当時に於いて革新的であり、その後に続く Maple や Mathematica に影響を与きました。

Maxima は Windows, Linux そして MacOS X 用にコンパイルされ、ソースも含めて入手可能です。Maxima は 1982 年から William Schelter によりメンテナنسされ、それは 2001 年の彼の死まで続きました。1998 年に GPL の下でソースコードをリースする許可を得た。この彼の努力とスキルにより Maxima は生き残ることが出来るようになりました。私達は彼の献身と彼の Macsyma code についての専門的な知識に感謝します。
`xmaxima` をターミナル内で実行すると maxima console が現れます。入力のプロンプトは (%in) です。出力のプロンプトは (%oN) です。代入はコロン(:)で表現します。

29.6.1 多項式

多項式の因数分解や展開を実行できます。

```
(%i1) factor( x^2 - y^2 );
(%i2) expand( (a+b)^2 );
```

¹<http://maxima.sourceforge.net/>

29.6.2 行列計算

```
(%i1) A:matrix([1, 1],[ 1,-1]);
(%i2) B:matrix([1,-1],[ 1, 1]);
(%i3) C:A.B;
```

の様にすると行列のかけ算をすることができます。

```
(%i4) A:matrix([1, -2, 3],[2, -3, 1],[3, -1, 2]);
(%i5) invert(A); (%i6) determinant(A);
(%i7) A:matrix([1, -2, 3],[2, -3, 1],[3, -1, 2]);
(%i8) B:matrix([4], [5], [6]);
(%i9) X:invert(A).B;
```

で逆行列や行列式を求めることができます。

で建立一次方程式 ($A \cdot X = B$) を行列を用いて解くこともできます。

29.6.3 プロット

関数のグラフを表示できます。

```
(%i10) plot2d (sin(x), [x, -%pi, %pi]);
```

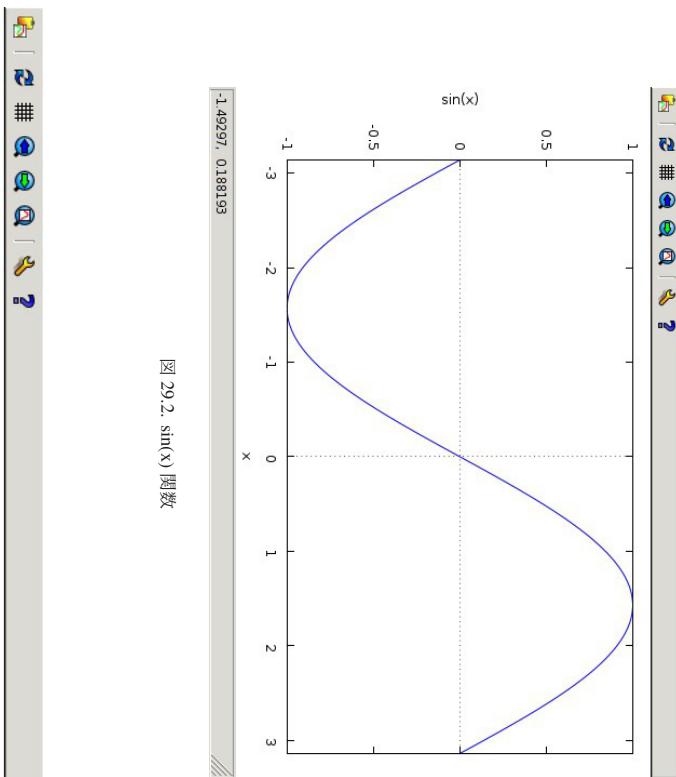


図 29.2 $\sin(x)$ 関数

```
(%i11) plot3d (2^(v^2-u^2), [u, -3, 3], [v, -2, 2]);
```

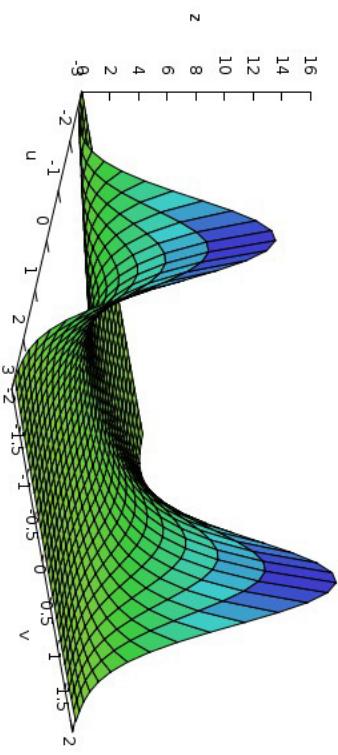


図 29.3 $z = \sin(x) * \cos(y)$ の 3 次元プロット

第30章 E-mail, Calculator, その他の便利な道具

30.1.2 再送

様々な理由によってメールの送信に失敗することがあります。

- 送信が不成功だったという通知が届く場合。
 - メールアドレス間に間違いがあると思われる所以で、もう一度確認する。
 - メールアドレスに間違いが無い場合、宛先のメールサーバに至るまでの経路上のネットワークの不調の可能性があります。ある程度時間をあけて再度、送信してみましょう。
 - それでも、失敗が続く場合は、宛先のメールサーバが受取り拒否をしている可能性があります。この場合は仕方がないのであきらめましょう。

30.1 電子メール (email)

電子メールはインターネットを利用した基本的な通信プログラムです。すでにメールを送ったり、受け取ったりしているでしょう。ここではもう一度メールについて説明します。

30.1.1 マナー

30.1.1.1 書き方

email を読む環境は人によって様々ですから、あまり迷惑をかけないような書き方をする必要があります。

- 一行を短くする。
多くの端末では半角文字80のサイズを利用することが標準です。したがって、それ以上に長く1行に書くのは好ましくありません。Emacsで書いている場合に右側にカーソルが届きそうになつたら改行キー(リターン)を押して改行して下さい。そうしながら見かけでは自動的に次の行に移行しているように見えますが、右端にバックスラッシュが表示されていて行が継続されていることが理解できます。
- Emacsではauto-fill-modeが効くなつていれば自動的に改行が挿入されます。
M-x auto-fill-modeを利用して有効/無効の切り替えを行うことができます。
- 相手の計算機環境に配慮する。
メールを受け取る相手の計算機環境を知っていますか？携帯電話が相手ならあまりサイズの大きなメールを送つては迷惑です。日本語を表示できない端末を利用している場合は日本語コードを含むメールを送りつけると読めない場合があります。
- エンコードの選択
2バイト文字のエンコードの方法はEUC, SJIS, JIS等があります。メールではJISコードで送る約束になっています。
- メールヘッダ情報に虚偽の情報を書かない。
になっています。メールヘッダを書き換えて嘘をついてはいけません。

30.1.3 添付ファイル

メールに写真や動画、音声、プログラム等を添付することができます。実行形式のプログラムや画像や音声のファイルをメールやニュースに発信する場合にはすべてのデータは印字可能なASCII文字列に変換されテキストデータとして添付されなければなりません。(RFC822)

30.1.3.1 MIME

ASCII文字列に変換するための方法はMIME¹インターネットメールフォーマット(RFC2046/2045)として規格化されています。そして多くのアプリケーションはこの規格に従って動作するように作られことが多くなってきました。

- ThunderbirdはMIMEに対応しています。

- sylpheedはMIMEに対応したメールツールです。

30.1.3.2 uuencode

MIMEが規格化される以前はuuencode²を利用してテキスト形式に変換してメールやニュースのために利用していました。現在でもuuencodeを利用してメッセージは一部で利用されています。uuencodeを利用して得られるテキストデータは以下のようにbeginで始まりendで終了します。

```
begin 644 login.jpg
MJUWS74\3D41!>ZR6.FD47MTL('#[J3T-G"STV@Z-58/T@0@S-IP;R-
M6;(O5G)CB)#+*-DD8AL-&T!>[O\GO:FRRH/Y]K4"?FTC_ST]-_
[中略]
...
MR06\Q21IVSTM?15-(K8;:8(C0.]T@5V-3ZW(HWS;2)"19(=71@07<E$7)&
end
```

30.1.1.2 メールの文書の取り扱い

- 個人の通信なので個人情報を第3者に開示しない。
メールのやりとりは個人的な通信です。ですから公開しないことが前提です。

これをMailなどのメッセージに取り込んで利用します。

¹See 'man mime' Multipurpose Internet Mail Extensions

²uuencode, uudecode - encode a binary file, or decode its encoded representation

30.1.4 メールを転送する

自分宛のメールを他のメールアドレスへ転送することができます。`$HOME/.forward` ファイルに次の様に書き、`chfwd` コマンドを実行しメールサーバの転送設定を変更すると、

```
\sXXXXXX@u-aizu.ac.jp
YYYYYY@zzzzzz.ne.jp
```

`sXXXXXX@u-aizu.ac.jp`宛のメールは `YYYYYY@zzzzzz.ne.jp` に転送されます。
ここで `\sXXXXXX@u-aizu.ac.jp` が無ければ転送されたメールはメールサーバの `sXXXXXX@u-aizu.ac.jp` のメールスプールには保存されません。

`$HOME/.forward` ファイルの作成と `chfwd` の実行を補助す `mkfwd` も提供されています。

携帯電話のメールアドレスに転送して、尚且つ学内にメッセージを残さない設定にすると携帯電話では大きなサイズのメールが読めなくて困ることがあります。学内の計算機にメッセージを残す設定を勧めます。

`$HOME/.forward` ファイルの設定ミスは大変な迷惑です。メールの配達ができないかったり、メール配達失敗を通知するメールが大量に発生することになります。`$HOME/.forward` ファイルの設定は必ず自分自身でメールの動作確認をしてください。確認の方法も思いつかない様な段階では、`$HOME/.forward` ファイルを設定するのは非常に危険です。情報センターでは `$HOME/.forward` ファイルの設定ミスによる問題が発覚した場合は緊急措置としてアカウント停止をしています。

情報センターの FAQ でもメール転送について説明されています。

```
http://web-int.u-aizu.ac.jp/labs/istc/ipc/faq/faq.html
```

`.forward` を作っても転送されない場合は、ホームディレクトリのパーミッションを確認しましょう。ホームディレクトリのパーミッションが `group` や `other` に対して書き込み許可の状態(`w`)の場合、またはホームディレクトリのパーミッションが `other` が読めない場合は、`sendmail` はメールを転送しません(できません)。また、`.forward` ファイルのパーミッションにも注意する必要があります。自分以外が読み書きできないよう設定してください。

```
% chmod go-rwx $HOME/.forward
```

携帯電話等への転送は、さらに注意が必要です。携帯への転送に失敗した場合、エラーメールが発信者に返ります。そのメールが例えば Mailing List (メーリングリスト) からのメールだった場合、メーリングリストのメンバーを含めて大勢の人による迷惑をかける危険性があります。また、機種によってはエラーメールから携帯の番号が分かってしまうこともあります。危険性を十分に認識して、慎重に設定を行ってください。

30.2 課題 30

`sylpheed` を用いて画像ファイルを添付したメールを自分宛に送り、送付されたメールをファイル `tempu` に保存せよ。作業方法は以下を参考にすること。

1. `sylpheed` を実行し `Subject` を `KADAI-tempu` とし画像を添付したメールを自分宛に送る。

2. `sylpheed` を利用してメッセージを受信する。

3. `grep KADAI-tempu $HOME/Mail/inbox/*` を実行しメッセージファイルのファイル名を確認する。

4. メッセージファイルを `$HOME/literacy/tempu` にコピーする。

5. `$HOME/literacy/tempu` のアクセスパームisionを探点可能な状態にする。

6. `grep Content $HOME/literacy/tempu` を実行してエンコードを確認せよ。

7. `less tempu` を実行し mime メッセージファイルを観察せよ。

8. `cat tempu | reformat -x` を実行し添付ファイルを抽出せよ。

MIME Content-Type の主なものを以下に示す。

種類

テキストファイル

HTML (Hyper Text Markup Language)

SGM (Standard Generalized Markup Language)

XML (eXtended Markup Language)

VML

Postscript

EPS (Encapsulated Postscript Format)

PDF (Portable Document Format)

TeX

Rich Text Format

PNG (Portable Network Graphics)

JPEG (Joint Photographic Experts Group)

GIF (Graphics Interchange Format)

TIFF (Tag Image File Format)

Compress

gzip で圧縮されたファイル

tar ファイル

application/postscript

application/postscript

application/pdf

application/x-latex

application/rtf

image/x-png

image/jpeg

image/gif

image/tiff

application/x-compress

application/gzip

application/x-tar

30.3 電卓

コンピュータは計算機ですから、いろいろな計算をする能力があります。ワークステーションに導入されているいろいろなソフトウェアを利用していろいろな計算をする例を紹介します。

- `xcalc`

`xcalc` は電卓です。

- `bc`

`bc` は任意精度の演算言語です。

- echo 4^2 | bc
- echo 'sqrt(2)' | bc -l

オプション `bc -l` をつけると、小数点以下20桁になります。また、このオプションをつけないときは、小数点以下0桁になります。端末で `bc -l` を実行すると `bc` でもインターラクティブに計算できます。

- perl
`perl` はプログラミング言語です。電卓としても簡単に使えます。
`perl -e 'print sin(3.14), "\n";'`
- mathematica
`mathematica` (マセマティカ) は 数学をする道具です。

- echo '4^2' | math -run
 ▶乗の計算。
- echo 'Inverse[{{1,2},{4,5}}]' | math -run
 逆行列。
- echo 'D[x^3+x^2,{x,1}]' | math -run
 微分。
- echo 'Integrate[x^3+x^2,x]' | math -run
 積分。

30.4 便利な道具

以下にこれまでの講義や演習で紹介できなかったツールを紹介する。自分で利用方法を調査してみましょう。

- du
 ユーザのディスクの使用量を調べる。

```
du -sk $HOME/.
```

- find
 パス名を検索する。

• cvs
 Concurrent Versions System. version control system.

- patch
 apply changes to files.
- gdb
 デバッガー

- doxygen
`C, Java, IDL (Corba, Microsoft, and KDE-DCOP flavors) ⇒ PHP, C#などのプログラムなどのドキュメントを作成するツール。`

付録A Sylpheed の設定

の様な出力を得た場合には
stdml4.u-aizu.ac.jp が受信用サーバです。

- 送信用サーバは受信用サーバと同じとします。

- A.1 はじめての利用**
- はじめて Sylpheed を利用する場合の手順を説明します。

env LANG=ja_JP.UTF-8 sylpheed

を実行すると日本語モード(メニューやメッセージが日本語)で sylpheed が起動します。日本語モードでなくとも日本語入力が可能になるように起動スクリプトが設定しています。はじめての場合はメールボックスの場所を尋ねできます。そのまま OK を選択します。

メールボックスの場所を指定してください。
よく分からぬ場合はそのままOKを選択してください。

このアカウントの名前 Account1 通常使用

個人情報

名前 _____

メールアドレス _____

組織 _____

サーバ情報

プロトコル POP3

受信用サーバー _____

SMTPサーバー(送信) _____

ユーザID a-t Fujii パスワード _____

このアカウントの名前 Account2 通常使用

個人情報

名前 _____

メールアドレス a-t Fujii@u-aizu.ac.jp

組織 _____

サーバ情報

プロトコル ニュース (NNTP)

ニュースサーバー _____

□ このサーバーは認証が必要

ユーザID a-t Fujii パスワード _____

パスワードを入力しないでおくと、メッセージを受信する場合に受信用サーバに接続する時にパスワードの入力が求められます。大学のメールアカウントの場合はログインのパスワードと同じものを使用します。

最後にアカウントの編集の窓が出来ます。これを閉じてセットアップは終了です。

次にメールアカウントの設定をする窓が現れます。ここでは、アカウントの名称、メールアドレス、プロトコルの選択、受信用サーバ、送信用サーバを設定する必要があります。

大学のメールアカウントを設定します。

- アカウントの名称は自由に決められます。
- メールアドレスは s125xxxx@u-aizu.ac.jp の様にします。
- 大学のメールサーバに合わせてプロトコルは POP3 のままにします。
- IMAP4 のサーバは IMAP4 にします。
- 受信用サーバは

```
openldapsearch -x -LLL
'(sendmailMTAkey=s125xxxx)',
```

を実行して出力される情報を利用します。

```
sendmailMTAlistvalue:
s125xxxx@stdml4.u-aizu.ac.jp
```

A.2 メールアカウントの追加

メニューからメールアカウントの追加を選んで、プロトコルの選択で、ニュース(NNTP)を選び、受信用サーバをあります。

A.3 ニュースアカウントの追加

メニューからメールアカウントの追加を選んで、プロトコルの選択で、ニュース(NNTP)を選び、受信用サーバを stdml4.u-aizu.ac.jp とします。このサーバでは認証は必要ありません。

このアカウントの名前 Account2 通常使用

個人情報

名前 _____

メールアドレス _____

組織 _____

サーバ情報

プロトコル ニュース (NNTP)

ニュースサーバー _____

□ このサーバーは認証が必要

ユーザID a-t Fujii パスワード _____

このアカウントの名前 Account1 通常使用

個人情報

名前 _____

メールアドレス _____

組織 _____

サーバ情報

プロトコル POP3

受信用サーバー _____

SMTPサーバー(送信) _____

ユーザID a-t Fujii パスワード _____

このアカウントの名前 Account2 通常使用

個人情報

名前 _____

メールアドレス _____

組織 _____

サーバ情報

プロトコル ニュース (NNTP)

ニュースサーバー _____

□ このサーバーは認証が必要

ユーザID a-t Fujii パスワード _____

付録B CUIベースのメール/

を実行すると、文字端末内で emacs を利用することができる
ます。

三九。

- GNUS
M-x grus を Emacs のミニバッファ中で実行すると
ニュースを読み書きする grus が動作します。

Sylpheed や Thunderbird などの GUI のツールは直感的な操作可能で使いやすいのが長所ですが、文字端末内などで動作できないので、文字端末を利用したりモードアクセス時には利用できません。CUI ベースのツールはその点で利用価値があります。

B.1 /bin/mail, /usr/bin/mailx の使い方

mail は文字端末内で利用できる最も簡単なメールアケーチョンの一つです。

- The screenshot shows a terminal window with the following content:

```
mail a-fujiitu@u-aizu.ac.jp
```

After pressing Enter, the terminal displays:

```
MAIL editor mode. For editing the tagged MING message.
```

Below this, the menu bar includes:

 - File
 - Buffers
 - Buffers
 - File
 - Tools
 - Edit
 - Search
 - Muse
 - Mainframe
 - Help

The message body is shown as:

```
In this mode, basically, the message is composed in the tagged MING format. The message tag looks like:
```

```
—[[text/plain; charset=UTF-8]](Title)
```

Below this, the message content is displayed:

```
The tag specifies the MING content type, subtype, optional parameters and character encoding of the message. Lines starting with '-' are messages and attachments according to the definition of the MING message format. Binary messages such as audio and image are usually transferred in a MING message when sending this mode.
```

Available characters depend on lines version being used. The following lists the available characters of each element:

 - From: [REDACTED]
 - To: [REDACTED]
 - CC: [REDACTED]
 - Subject: test

At the bottom of the terminal window, there is a status bar with the URL <http://gentoo/gmail>.

mailx も同じように利用できますがオプションや操作が若干異なります。

mail, mailx 共に直接メール受信サーバのメッセージファイルを読みます。メッセージを削除する操作をしなければメッセージは残ります。

sylpheed はメッセージファイルのメッセージを \$HOME/inbox/ 内に移し、メッセージファイルを空にする設定 있습니다。

B.2 Emacs を使用したメール/ニュース

- C-c C-m ⌘ Mime エンコードします。
 - C-c C-c ⌘送信します。

B.3 その他のツール

その他の、世の中には `nmh`, `mutt`, `alpine` などのメールクライアントがあります。これらは会津大学の環境には導入されていませんが、みなさんが他の環境で利用することもあるかも知れません。

MIME editor mode for editing the tagged MIME messages.

添付されたファイルをディスクに保存する場合は添付ファイルのエンティティにカーソルを合わせて、e_o をタイプします。

付録C タイピング練習 C言語

examples

C言語の基本的な使用方法を紹介するための簡単なサンプル集です。UTF-8の文字コード以外では採点しません。

ソースプログラム： prog1c.c

```
#include <stdio.h>

int main(void) {
    int n=1, sum=0;
    while (n <= 100) {
        sum += n;
        /* n++ は、n = n + 1 と同じ */
        n--;
    }
    printf("1+2+3+...+%d=%d\n", n, sum);
    return 0;
}
```

C.1 100以下の自然数の和

実行結果 はいざれも $1+2+3+\dots+100 = 5050$ です。

ソースプログラム： progla.c

```
#include <stdio.h>
```

```
int main(void) {
    int n, sum;
    sum = 0;
    while (n <= 100) { /* n <= 100 が繰り返しの条件 */
        sum += n; /* sum に和を求める */
        n++;
    }
    printf("1+2+3+...+%d=%d\n", n, sum);
    return 0;
}
```

C.2 100以下の正の偶数の和

実行結果 は $2+4+6+\dots+100 = 2550$ です。

ソースプログラム： prog1b.c

```
#include <stdio.h>
```

```
int main(void) {
    int n=1, sum=0;
    while (n <= 100) { /* sum += n は、sum = sum + n と同じ */
        sum += n; /* n += 1 は、n = n + 1 と同じ */
        n -= 1;
    }
    printf("1+2+3+...+%d=%d\n", n, sum);
    return 0;
}
```

ソースプログラム： prog1d.c

```
#include <stdio.h>

int main(void) {
    int n, sum=0;
    for (n = 1; n <= 100; n++) {
        /* n の初期値は1 */
        /* n <= 100 が繰り返しの条件 */
        /* ループが一回終了する度に、n に1を加える */
        sum += n; /* sum に和を求める */
    }
    printf("1+2+3+...+%d=%d\n", n, sum);
    return 0;
}
```

C.3 $1+2+3+\dots+n \geq 500$ を満たすの最小のnとそのときの和

実行結果は $n=32$ $sum=528$ です。

ソースプログラム： prog3.c

```
#include <stdio.h>
```

```
int main(void) {
    int n=0, sum=0;
    while (sum < 500) {
        sum += n;
        n++;
    }
    printf("n=%d sum=%d\n", n, sum);
    return 0;
}
```

ソースプログラム： prog2b.c

```
#include <stdio.h>
```

ソースプログラム： prog2a.c

```
#include <stdio.h>
```

```
int main(void) {
    int even=0, sum=0;
    while(even <= 100){/*even<=100 が繰り返しの条件*/
        sum = sum + even; /* sum に和を求める */
        even = even + 2; /* even に2を加える。
        2,4,6...100 となる */
        even = even - 2;
    }
    printf("2+4+6+...+%d=%d\n", even, sum);
    return 0;
}
```

C.4 $1+2+3+\dots+n < 500$ を満たす最大の n と
そのときの和

実行結果は n=31 sum=496 です。

ソースプログラム: prog4a.c

```
#include <stdio.h>

int main(void) {
    int n=0, sum=0;
    while (sum < 500) {
        n++;
        sum += n;
    }
    sum -= n;
    /* -[回分余計なので、引く] */
    printf("n=%d sum=%d\n", n, sum);
    return 0;
}
```

ソースプログラム: prog4b.c

```
#include <stdio.h>

int main(void) {
    int n=0, sum=0;
    int nNext=0, sumNext=0;
    while (nNext<0, sumNext<0; /* sum < 500 */)
        (sum < 500)*/ {
            n = nNext;
            nNext++;
            sumNext += nNext;
        }
    printf("n=%d sum=%d\n", n, sum);
    return 0;
}
```

C.5 500 400 300 200 100 0 と表示する

ソースプログラム: prog5.c

```
#include <stdio.h>
#include <complex.h>

int main(void) {
    for (n = 500; n >= 0; n -= 100) {
        int n=0;
        double complex w,z;
        double t=3.1415926;
        w=1.0;
        //z=cos(t/3)+sin(t/3)*I;
        z=z*exp(I*t/3);
        while (n <= 6) {
            printf("z=%f+%f*I\n",n,creal(w),cimag(w));
            w*=z;
            n++;
        }
    }
    printf("\n");
    return 0;
}
```

C.6 $1 + 2 + 3 + \dots + 10 \geq 1^2 + 2^2 + 3^2 + \dots + 10^2$

実行結果は $1 + 2 + 3 + \dots + 10 = 55$ $1^2 + 2^2 + 3^2 + \dots + 10^2 = 385$ です。

ソースプログラム: prog6.c

```
#include <stdio.h>

int main(void) {
    int Sum=0, SumOfQuarter=0, n;
    for (n = 0; n <= 10; n++) {
        Sum += n;
        SumOfQuarter += n * n;
    }
    printf("1+2+3+...+10=%d\n", Sum);
    printf("1^2+2^2+3^2+...+10^2=%d\n", SumOfQuarter);
    return 0;
}
```

C.8 ブール型の利用

ソースプログラム: boolean.c

```
#include <stdio.h>
#include <stdbool.h>

int main(int argc, char **argv) {
    bool yes=true;
    int num=10;
    int n=0;
    if(argc>1) num=atoi(argv[1]);
    while (yes) {
        printf("%d\n",n);
        n++;
        if(n>num) yes=false;
    }
    return 0;
}
```

C.7 複素数 $e^{i\frac{\pi}{3}}$ の n 乗の表示

ソースプログラム: complex.c

```
#include <stdio.h>
#include <complex.h>

int main(void) {
    int n=0;
    double complex w,z;
    w=1.0;
    //z=cos(t/3)+sin(t/3)*I;
    z=z*exp(I*t/3);
    while (n <= 6) {
        printf("z=%f+%f*I\n",n,creal(w),cimag(w));
        w*=z;
        n++;
    }
}
```

付録D 演習 参考資料 **LATEX**sample

D.2 フォントの例

```
latex2.tex
\documentclass[a4paper]{article}
\title{A \LaTeX ~sample [Font]}
\author{a-fujitu@u-aizu.ac.jp}
\date{\today}
\begin{document}
\maketitle
```

TeX の基本的な使用方法を紹介するための簡単なサンプル集です。サンプル中の

a-fujitu@u-aizu.ac.jp は自分のメールアドレスに置き換えてください。指示通りに置き換なければ、採点対象としません。UTF-8 以外の文字コードでは採点しません。

D.1 **Latex** の基本例

latex1.tex

```
\documentclass[a4paper]{article}
% 必ず最初にこのように宣言します。
% %より右側はコメントです。
\title{A \LaTeX ~sample for Beginners}
\author{a-fujitu@u-aizu.ac.jp}
\date{\today}
\begin{document}
\maketitle
\begin{center}
\begin{large}
\begin{tt type writer} \it italic \end{tt}
\begin{tiny tiny} \bf Bold face \end{tiny}
\begin{footnotesize footnotesize} \rm roman \end{footnotesize}
\begin{small small} \rm Gothic ゴシック \end{small}
\begin{large large} \rm Mincho 明朝 \end{large}
\begin{huge huge} \rm Slant Small Caps \end{huge}
\begin{sf sans serif} \rm Slant Small Caps \end{sf}
\end{large}
\end{center}
\end{document}
```

これは初心者のための**\LaTeX** のサンプルです。
改行はバックスラッシュ二つ\\で指定できます。
改段落は空行か **\par** でできます。
\end{document} % これで終了です。

D.3 章立ての例

latex3.tex

```
\documentclass[a4paper]{article}
\title{A \LaTeX ~sample [Section]}
\author{a-fujitu@u-aizu.ac.jp}
\date{\today}
\begin{document}
\maketitle
\section{第1セクション}
\subsection{第1-1サブセクション}
\subsubsection{第1-1-1サブサブセクション}
この様に文章の構造を簡単に指定できる。
% jarticle では section から指定できます。
% jbook では chapter も指定できます。
\subsubsection{第1-2サブセクション}
これは章の構造を使用する\LaTeX のサンプルです。
\section*{こんなこともできます。}
\end{document}
```

いろいろなフォント（文字の種類や大きさ）を簡単に使用できます。

D.4 箇条書きの例

latex4.tex

```

\documentclass[a4paper]{article}
\title{A \LaTeX ~sample [Itemize]}
\author{a-fujiru@u-aizu.ac.jp}
\date{\today}
\begin{document}
\maketitle
\section{Itemize}
\begin{subection}{Itemize}
\begin{itemize}
\item みかん
\begin{itemize}
\item[★] はっさく
\item[×] 伊予柑
\end{itemize}
\end{itemize}
\end{subection}
\begin{itemize}
\begin{itemize}
\item[☆] はな
\end{itemize}
\end{itemize}
\begin{itemize}
\begin{itemize}
\item[△] 台湾
\end{itemize}
\end{itemize}
\end{itemize}
\end{document}

```

D.5 作表の例

D.6 図の例

```

latex6.tex
\documentclass[a4paper]{article}
\usepackage[dvipdfmx]{graphicx}
% 画像ファイル (eps,png,jpeg, pdf) を取り込む場合には
% graphicx の使用を宣言します。
\title{A \LaTeX ~sample [Post Script]}
\author{a-fujitu@u-aizu.ac.jp}
\date{\today}
\begin{document}
\maketitle
\section{Postscript の取り込み}
\begin{center}
\begin{figure}[pth]
\begin{center}
\begin{tikzpicture}
\includegraphics[scale=0.5]{%
/home/course/lit1/pub/lit1/FIGURE/sylpheed-3.0.3.eps}
\end{tikzpicture}
\end{center}
\end{figure}
\end{center}
\caption{sylpheed3 を開いたところ}
\end{document}

```

figure と includegraphics のオプションの説明

```

\begin{center}
\begin{tabular}{c || c }
scale & 縦横同倍のスケール \\
bscale & 縦方向のスケール \\
vscale & 縦方向のスケール \\
width=10zw & 幅幅を全角 10 文字幅分にスケール \\
height=10cm & 高さを 10cm にスケール \\
angle=90 & 度回転 \\
origin=c & センターを中心に回転 \\
h & その位置に表示 \\
t & ページ上端に表示 \\
b & ページ下端に表示 \\
p & 単独ページに表示 \\
\end{tabular}
\end{center}

```

あまり使わないけど、図だけでなく文字の回転もできる。

D.7 数式の例

```

latex7.tex
\documentclass[a4paper]{article}
\begin{equation}
\int_a^{\infty} f(x) dx
\end{equation}
\begin{equation}
\label{number}
y = \frac{a_1 + a_3}{b_1 + b_3 + \cdots} + \sqrt{1 + x}
\end{equation}
\begin{array}{l}
\text{式}\backslash ref\{number\} \text{は... の様に引用する。} \\
\begin{eqnarray}
y &=& 1+2+3 \\
&& \& 1+5 \\
&& \& 6 \\
\end{eqnarray}
\end{array}
\begin{array}{l}
\text{文書中で使用したいときはドルマーク\$ ではなくと良いでしょう。} \\
\text{この様な \$sum_{n=1}^{23} 使い方です。}
\end{array}

```

D.8 グラフの例

```
latex8.tex
\documentclass[a4paper]{article}
\usepackage{pstricks-add}
\title{A \LaTeX ~PSTricks Sample}
\author{a-fujitu@u-aizu.ac.jp}
\date{\today}
\begin{document}
\maketitle
\begin{center}
\begin{psframebox[fillstyle=solid,fillcolor=blue!10,framesep=0pt]{}
\begin{pspicture}(-3,-2)(6,6)
\psaxes{>}(-3,-3)(6,6)
\pscustom[fillstyle=solid,fillcolor=yellow!40,linestyle=none,algebraic]{%
\psline{3,0}(1,0)
\psplot[-2.8]{6}{\sin(x)*cos(x)+sqrt(1+abs(x))+1}
\psplot[linewidth=2pt,algebraic]{-2.8}{6}{\sin(x)*cos(x)+sqrt(1+abs(x))+1}
\end{pspicture}
\end{psframebox}
\end{center}
\end{document}
```

D.9 文献参照の例

```
bunkent1.bib
@article{ritchie1974unix,
  title={The UNIX time-sharing system},
  author={Ritchie, D.M. and Thompson, K.},
  journal={Communications of the ACM},
  volume={17},
  number={7},
  pages={365--375},
  year={1974},
  publisher={ACM}
}
@ARTICLE{knuth1975,
  author={Knuth, D.E. and Rao, G.S.},
  journal={Computers, IEEE Transactions on},
  title={Activity in an Interleaved Memory},
  year={1975},
  month={sept.},
  volume={C-24},
  number={9},
  pages={943 - 944},
}
@book{shannon1962,
  title={The mathematical theory of communication},
  author={Shannon, C.E. and Weaver, W.},
  volume={19},
  number={9},
  year={1962},
  publisher={University of Illinois Press Urbana}
}
```

latex9.tex

```
\documentclass[a4paper]{article}
\bibliographystyle{plain}
\title{A \LaTeX ~Citation Sample}
\author{a-fujitu@u-aizu.ac.jp}
\date{\today}
\begin{document}
\maketitle

```

これは文献参照のサンプルです。

参考文献\cite{ritchie1974unix}は UNIX OS の論文です。
bunkent1.bibにリストしている文献リストの中で、
この論文が引用する文献リストを出力します。
%ここに文献リストを出力します。
\bibliography{bunkent1}
\end{document}

付録E 演習 参考資料 TGIF Manual

作図の補足として、マニュアルの一部を添付します。

E.1 四角形を描く。

四角形を描いてみます（図 E.2 参照）。

1. 「Mode メニュー」から「四角形」を選択して下さい（図 E.1 参照）¹.
2. カーソルが白くなります。
3. 「Canvas ウィンドウ」の左上の適当な場所で、「左ボタン」を押しドラッグして下さい。
4. 四角いラバーバンドが表示されます²
5. 適当な場所で、ボタンを離すと、四角形が描けます。

なお、一度描画モードになると、「Mode メニュー」から「矢印」を選択して「操作モード」にするまでは（つまり）、カーソルが「白い」（ちは）、同じ图形が続けて描画できます。操作モードになるとカーソルは、「黒く」なります³。

E.2 円を描く。

「Mode メニュー」から「円」を選択します。後は、四角形と同じです（図 E.2 参照）。

E.3 文字を入れる。

文字列を入れてみます（図 E.3 参照）。

- (1) 「Mode メニュー」から「T」を選択して下さい。
- (2) カーソルが「T」の字になります。
- (3) 文字を入力したい場所で、「左ボタン」をクリックして下さい。

¹なお、現在の「描画モード」は、「Choice ウィンドウ」（講義資料図2）に表示されます。
²この場合、ラバーバンドはある一定の間隔（グリッド間隔）でしか、大きくすることができます。このグリッドをうまく使うときれいな図が素早く描けます。

³一時的に、「操作モード」にしたい場合は、「E」（コントロールを押しながら「E」を押す。Eは小文字）とキーボーデュから入力して下さい。もう一度「E」を入力するまでは、カーソルが「黒く」なって「操作モード」になります。再度「E」を入力すると、元の「描画モード」に戻ります。

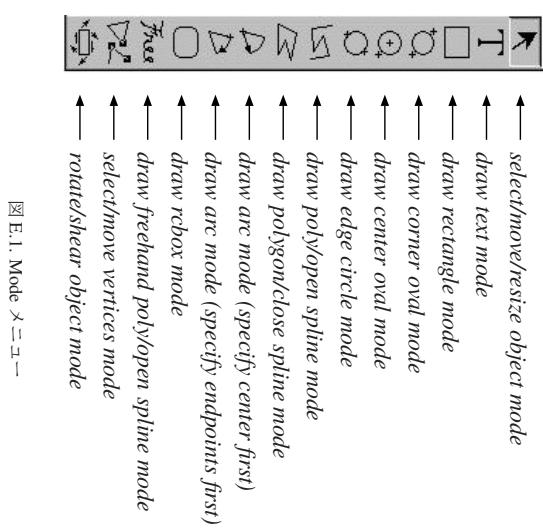


図 E.1 Mode メニュー

E.4 円の線を太くし、中を塗る。

このような操作は、图形の属性を変更することで行ないます。

- (1) まず、「Mode メニュー」の「矢印」を選択して下さい。

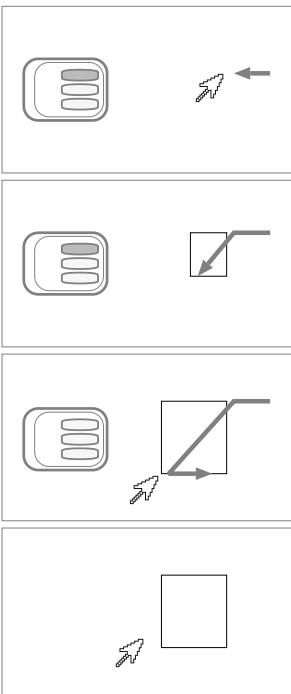


図 E.2 四角形を描く

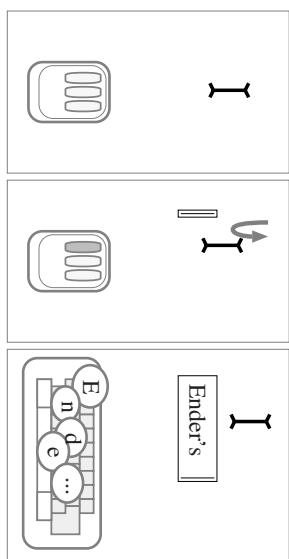


図 E.3. 文字を描く

- (2) カーソルが黒くなります。
- (3) カーソルを円周の上に持つていって「左ボタン」をクリックして下さい。
- (4) 円の回りに、「小さな四角」が表示されます（これを「図形を選択する」と言います）。されない場合は、粗い方が悪いのでもう一回やってみて下さい:-P.
- (5) まず、線の太さを変えます。
- (6) 「LineStyle メニュー」から適当な太さの線を選択して下さい。円の線の太さが変わったはずです（図 E.4 参照）⁴.
- (7) 次に中を塗りつぶします。
- (8) 「Fill メニュー」から適当なパターンを選択して下さい。円の中が塗りつぶされたはずです（図 E.5 参照）。

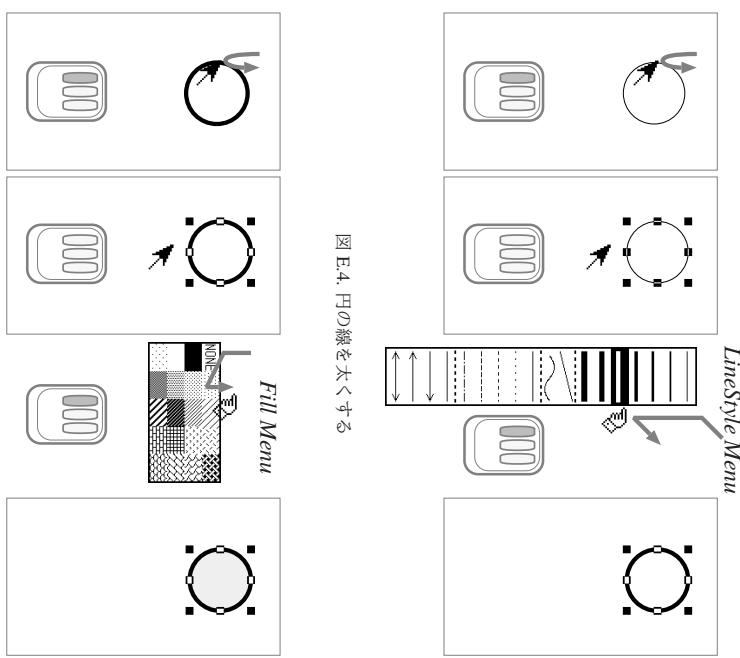


図 E.4. 円の線を太くする

E.5 矢印を描く。

矢印を描いてみます（図 E.6 参照）。

- (1) 「Mode メニュー」の「ポリライン」を選択して下さい。
- (2) カーソルが白くなります

E.6 矢印の線を太くし、ペンパターンを変える。

矢印の線を太くするのは、円の線を太くするのと同じです。
また、ペンパターンを変えるのは、「Pen メニュー」を使う以外は、円を塗るのと同じで
す（図 E.7 参照）。

- (3) 「Canvas ウィンドウ」の適当な場所を「左ボタン」でクリックして下さい。

- (4) マウスを動かすと線が引けます。
- (5) 「左ボタン」をクリックして下さい。折れ線になります。
- (6) マウスを動かすと線が引けます。
- (7) 「真中ボタン」をクリックすると線が確定します。

⁴この場合、「LineStyle メニュー」のアイテムのいくつかが反転表示されていると思います。これは、そのオブジェクト（円）の「LineStyle メニュー」に関するカレントの状態を反転表示しているます。

- (1) 「Mode メニュー」の「矢印」を選択して下さい。
- (2) 四角形の辺の上で、「左ボタン」をクリックしてください。
- (3) 四角形の回りに、「小さな四角」が表示されます。
- (4) 「小さな四角」の一つの上にカーソルを持っていき、「左ボタン」を押して下さい。

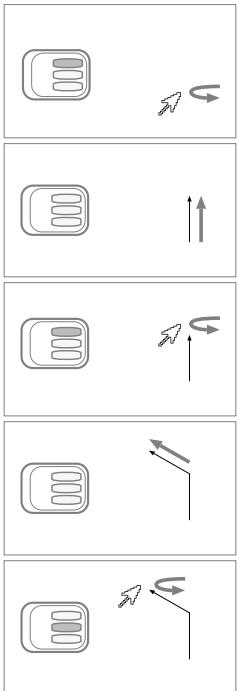


図 E.6. 矢印を描く

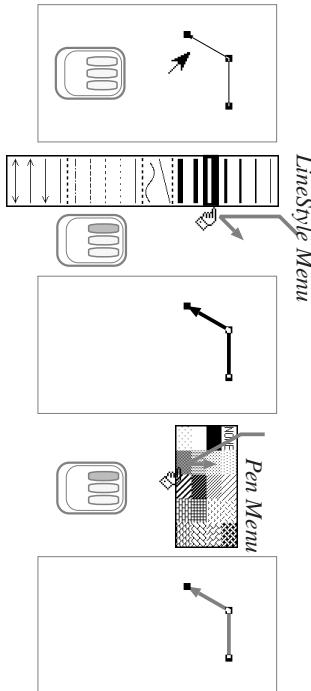


図 E.7. 矢印を大きくし、塗る

(5) 「左ボタン」を押したまま、マウスをドラッグすると大きさが変わります。

(6) ボタンを離すと、その大きさになります。

オブジェクトのリサイズの仕方は、他のものも同じです。

E.8 円を移動する。

円の場所を変えてみましょう（図 E.9 参照）。

(1) 円を選択します。

(2) 「小さな四角以外」の部分にカーソルを持っていき、「左ボタン」を押しながらドラッグして下さい（「小さな四角」を選択するとリサイズされてしまいます）。

(3) ボタンを離すとその場所に移動します。

オブジェクトの移動の仕方は、他のものも同じです。

⁵ 文字列はリサイズできません。大きくしたいときは、テキストサイズを変えて下さい（文字列を選択して、「FontSize メニュー」で、文字列の大きさを変更する）。また、ピットマップモリサイズできません。なお、ポリゴンとポリラインは、大きいではなく形状が変わります。リサイズする場合には、何か他のオブジェクトとダブルクリックしてからリサイズ下さい。

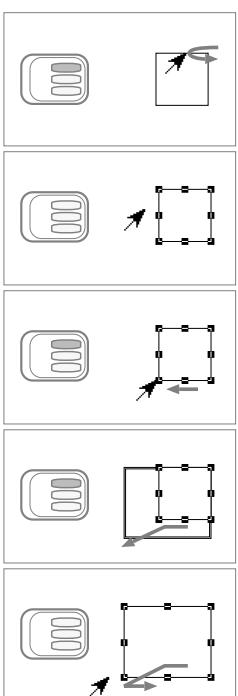


図 E.8. オブジェクトをリサイズする

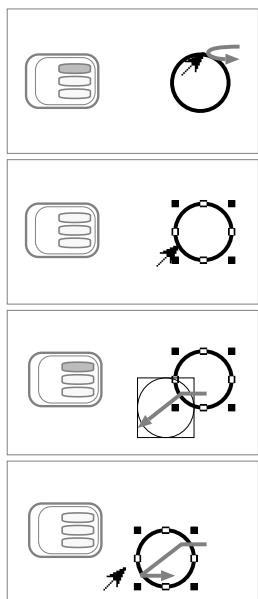


図 E.9. オブジェクトを移動する

E.9 文字のフォントを変える。

文字のフォントを変えてみます（図 E.10 参照）。

(1) 文字列を選択します。

(2) 「Font メニュー」から「Times」を選択します。

なお、フォントのスタイル（ローマン体、ポールド体、イタリック体…）を変えるには、「TextStyle メニュー」を使って下さい。

E.10 その他

ほかにもいくつか知つておいた方が良いものについて書いておきます。

- 「LineStyle メニュー」で、矢印を「スライン化」し、「矢印を直線」にしてみます（図 E.11 参照）。

スライン化は、ポリゴンに対しても可能です。

- デフォルトの変更

描画を行なった時に、どのような描画属性を持つオブジェクトが描画されるかは、「Choice ウィンドウ」（講義資料図 2 参照）に示されています。例えば、講義資料図 2 の状態で線を引くといつも、

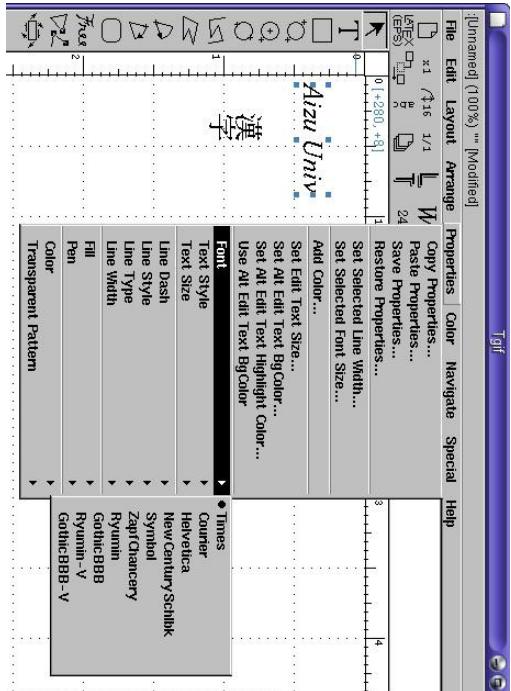


図 E.10. 文字のフォントを変える

- (a) 「線の端点形状」は「矢印」
- (b) 「ペンパターン」は「べた塗り」
- (c) 「線の幅」は「一番細い」

という属性の線が引かれます。
このデフォルトを変えるには、つぎの二つの方法があります。

- (a) 「Choice ウィンドウ」のアイコンを「左ボタン」でクリックする。
- (b) 何も選択していない状態で、メニューから設定したい属性を選ぶ。

とりあえず、「→」が表示されている「線の端点形状」のアイコンを、「左ボタン」で4回クリックすると図 E.12 のようになります、この状態で線を引くとデフォルトで「矢印なし」の線」が引ける。

E.11 まとめ

- 描画
 - 「Mode メニュー」から描画したいオブジェクトを選択する。
 - 「オブジェクト」の選択
 - 「左ボタン」をクリックする。
 - 操作対象のオブジェクトを選択してから行ないます。

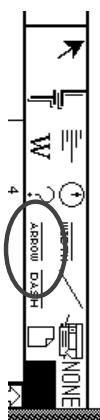


図 E.12. デフォルトの変更

- 属性の変更
 - 選択したオブジェクトに対してメニューで変更する。
 - 移動
 - 拡大（縮小）
 - 選択したオブジェクトを「左ボタン」で、「小さな四角」以外の部分をドラッグする。

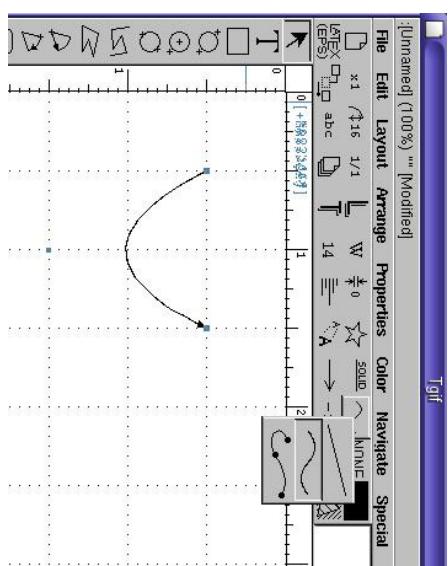


図 E.11. スライン化

付録F AINSの利用

F.1 大学の標準環境の利用方法

大学の演習室の標準環境を演習室以外で利用する方法を紹介します。

- 研究室の端末を利用できます。
- 個人の端末を利用する。

F.1.1 研究室の端末や個人の端末

研究室への出入りを許可されて研究室内の端末を利用できる場合は、その端末を利用し
て演習室の標準環境を利用できます。また、 WiFi を利用して AINS に接続している場合
も演習室環境を利用できます。

- 卒研用 SunRay 端末の場合はそのまま利用できます。

個人が所有しているパソコンで演習室の標準環境を利用できます。sadrsv1 から
sadrsv5, aplisv1, aplisv2 は学内からリモートアクセスして利用できます。

- Mac OS X や Linux, Solaris の様に X11 や ssh がインストールされている
場合は X11 を起動し、 xterm 等の中で、

```
ssh -X username@aplisv1.u-aizu.ac.jp
```

→ aplisv1 にログインすれば、

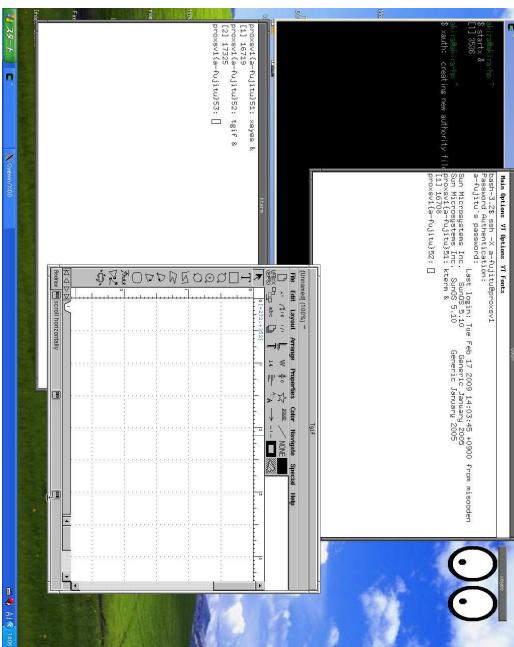
```
gnome-terminal &
```

などが実行できます。

- Windows の場合は X11 も、 ssh も標準ではインストールされていないので、両
方ともインストールする必要があります。Cygwin を利用する例は以下の通り。
 - * Cygwin をインストールする。(x11, openssh を追加する)
 - * Cygwin を起動する。

startx &
* などで X11 を起動する。

* 後は同様。



stdvnccsv1.u-aizu.ac.jp で利用できる VNC サーバを利用すると Solaris のデス
クトップ環境を PC や Mac で利用できます。



F.1.2 学外から利用する。

大学の外から学内の stdvncsv1.u-aizu.ac.jp にリモートアクセスして VNC 等を利用すれば学内の標準デスクトップ環境を利用できます。

- sshgate から stdvncsv1 にリモートログインする。

```
rsh stdvncsv1
```

- vnc パスワードを未設定の場合は vncpasswd で vnc パスワードを設定する。

- stdvncsv1 で VNC サーバを起動する。autokill は 1 回利用後に自動終了するためのオプション。

```
% vncserver -autokill -geometry 800x600
New 'stdvncsv1:1 (s125xxxx)' desktop is stdvncsv1:1
```

```
Starting applications specified in ~/.vnc/xstartup
```

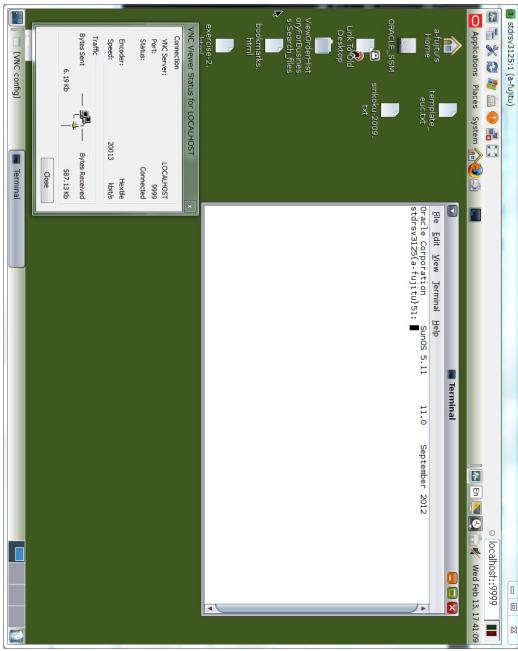
```
Log file is ~/vnc/stdvncsv1:1.log
```

- 起動時メッセージからデスクトップ番号を確認する。ここでは 1 番とする。

- 上記 Log file からポート番号を確認する。ここでは 5901 番とする。
通常は(ポート番号)=5900+(デスクトップ番号)である。

- Mac の場合は Safari で vnc://stdvncsv1.u-aizu.ac.jp:5901 にアクセスする。

- Windows の場合は vncviewer などの vnc クライアントを起動し、ホスト名 stdvncsv1.u-aizu.ac.jp、ポート番号 5901 で接続する。



F.2 学内での個人PCのネットワーク接続

大学内で個人所有のコンピューターを利用することは可能です。無線 LAN や特定の部屋に設置してある有線接続の認証ネットワークの様に個人のユーザ認証が提供されているものは利用可能ですが、以下の場合は禁止されています。

- 演習室の端末に接続されているケーブルを利用してはいけない。
- 研究室で教授の許可を得ずに個人の PC 等を勝手に接続してはいけない。

8. vnc クライアントを使用終了後、stdvncsv1.u-aizu.ac.jp で Xvnc プロセスが終了していないければ、手動で VNC サーバを終了する。デスクトップ番号 1 番の場合。

```
vncserver -kill :1
```

付録 G 自分の PC で自分の UNIX 環境

毎年、複数の学生が「学習用に個人所有のパソコンでも大学の標準環境と同等なものを利用したい」という要求をしてきますので、実現方法の一部を紹介します。

1. Mac OS X の場合

Mac OS X は UNIX ベースです。X11 も ssh も標準で利用できます。UNIX で利用されているソフトウェアを追加するためには MacPorts などを利用すると良いでしょう。

2. Windows OS の場合

Cygwin を導入すると良いです。Cygwin には X Window System や OpenSSH などが含まれています。

3. 値構化技術を利用する場合

VirtualBox などの値構化デスクトップ環境を利用して Mac OS X や Windows のコンピュータをホストとして様々なゲスト OS を導入できます。ゲスト OS として Linux や Solaris をインストールするとホスト OS とは独立した UNIX 環境を構築して利用できます。



5. FreeBSD や NetBSD の場合
BSD 系の UNIX と言えばこれ等のことを持ちます。

6. Oracle Solaris の場合
大学の標準環境と同等です。

G.1 大学の WS で Windows や Linux を動かす

一部の演習室 (std5, 6, ilab1, 2) では VirtualBox を利用できます。仮想マシンのファイルは 1GByte を超えるサイズになるので個人所有のポータブル HDD や USB メモリに仮想マシンを導入すれば、それを演習室の端末で動かすことが可能です。



HDD 等のリムーバブルストレージ内に vbox フォルダを作成し、次の様に記述されたスクリプトを用意し、実行することで利用が可能となります。

```
env VBOX_USER_HOME=/mount_point/vbox VirtualBox
```

4. Linux の場合
Linux の各種ディストリビューションで使用されているソフトウェアは殆んど UNIX と同等です。

(8) その他センター長が別に定める事項に違反してはならない。

2 利用者は、所定の認証手続を経てなされたAINS資源の利用の結果について責任を負わなければならない。

* (AINS資源の返却) *

第9条 利用者は、次に掲げる事項に該当する場合には、該当するAINS資源を直ちに返却しなければならない。

(1) 利用者が、第4条で定めるAINS資源の利用資格を失ったとき

(2) 当該AINS資源の利用が不要となつたとき

(3) その他センター長が必要と認めたとき

* (運用管理の妨げの禁止) *

第10条 利用者は、AINS資源の運用管理の妨げになるような行為を行つてはならない。

2 センター長は、前項に定める行為を防止するための措置を講ずることができる。

* (報告等) *

第11条 センター長は、必要に応じて、利用者に対しAINS資源の利用に関する報告を求めることができる。

2 利用者は、AINS資源に障害が発生するおそれがあると認められる場合及び障害が発生した場合は直ちにセンター長に報告しなければならない。

* (利用制限又は禁止) *

第12条 センター長は、利用者がこの規程に違反した場合又はAINS資源の運用管理に重大な支障を及ぼした場合は、当該利用者のAINS資源の利用を制限し、又は一定の期間その利用を停止することができる。この措置は、予告なく通信を遮断することによって行なうことがある。

2 センター長は、ある利用者の行為及び行為の結果を放置すれば、この規程の禁止事項に抵触すると判断される場合、当該状態が改善されるまでの間、当該利用者のAINS資源の利用を制限し、又は停止することができる。この措置は、予告なく通信を遮断することによって行なうことがある。

3 前2項の規定により、AINS資源の利用を制限し、又は停止した場合は、センター長は、速やかに関係部局の長に報告するものとする。

4 利用者は、利用の制限又は停止に対して、関係部局の長を経由して、センター長に異議申立てをできる。

* (利用者責任) *

第13条 利用者は、AINS資源に損害を与えた場合、特別な事情があると認められた場合を除き、センター長の指示に従い、原状回復、返却又は弁償の責任を負わなければならぬ。

* (運用管理上の制限又は停止) *

第14条 センター長は、AINS資源の運用管理上必要を認められる場合、A I N S 資源の利用を制限し、又は停止することができる。また、障害などにより、A I N S 資源の利用が制限される場合がある。

* (機密保護) *

第15条 センター長及びAINS資源の運用管理に従事する者は、AINS資源上にある利用者の機密の保護に努めるものとする。

2 センター長及びAINS資源の運用管理に従事する者は、この規程で定める事項の維持、保全の目的以外には職務上知り得た利用者の情報を利用し、又は第三者に開示してはならない。ただし、AINS資源の運用管理上必要となる場合その他公益上やむを得ない理由がある場合はこの限りでない。

* (補則) *

第16条 この規程に定めるもののほか、AINS資源の利用その他必要な事項については、センター長が別に定めることとする。

附則

1 この規程は、平成18年4月1日から施行する。

2 既に第4条の利用承認を得ているものとみなす。ただし、センター長が改めて利用の承認手続が必要と認めた場合を除くものとする。

附則

1 この規程は、平成24年4月1日から施行する。

AINS 利用に関する注意事項

<http://web-int.u-aizu.ac.jp/labs/istc/ipc/rules/caution.html> を以下に記載します。

情報処理センター

| 会津大学 | 学内限定 [情報処理センター AINS 利用に関する注意事項]

AINS 利用に関する注意事項

- 各種申請の処理に要する期間について
- 利用者の卒業、退職等の場合の申請事項の抹消について
- 教室の端末使用と **xlock** に関する注意
- 学生のディスク使用量について
- パスワードの頻繁な変更について
- 不正、不適切な利用事例

各種申請の処理に要する期間について

情報センターが利用者からの各種の申請を処理するには、営業日で少なくとも数日の期間がかかります。申請内容によってはそれ以上の期間を要する場合もあります。「至急」の申請は原則として認められませんのであらかじめご了承ください。

利用者の卒業、退職等の場合の申請事項の抹消について

利用者が会津大学教職員／学生の身分を失う場合はその利用者が行なった申請事項、あるいはその利用者が管理者／代表者となっている申請事項は全て抹消されます。申請事項を引き続き使用する場合は、身分を失う前に管理者／代表者の変更手続きや再申請を行なって下さい。

*[メーリングリスト、UNIX グループについての特別措置] *

- メーリングリストの管理者の変更
 - メーリングリストの管理者がアカウントを失ってしまった場合、アカウント失効後 1 カ月間に限り以下の条件で管理者の変更依頼を認めます。
 - メーリングリストの残ったメンバー内で新管理者が承認されていること。このことはメンバー全員に変更依頼のメールを CC することで確認する。
 - サークルのためのメーリングリストでは当該サークルが学生課で認められた公認団体であること。
 - メーリングリストのメンバーには少なくとも 1 名の教職員が含まれること。サークルのためのメーリングリストでは顧問教員がメンバーに含まれること。

教室の端末使用と **xlock** に関する注意

利用者は教室の端末を使用する場合、以下の事項を守って下さい。

- 端末を使用していて席を外す場合には、必ずログアウトすること。
- 上記 a の例外として数分以内で確実に戻ってくる場合にのみ **xlock** を使用して良い。(目安として 5 分以上席を外す場合はログアウトすること)

情報センターでは以下のようないふたつの処置による不利益について補償しませんので了承下さい。

- xlock** が授業に支障をおよぼす場合
- 長時間放置されている場合

2. サークル用の UNIX グループの代表者の変更

サークル用の UNIX グループの代表者は常に顧問教員となります。顧問教員がアカウントを失う場合、そのサークルが引き続き公認団体として活動するには他の教員が新たに顧問となる必要があります。その場合に情報センターで新顧問を確認し、その教員に代表者を変更します。当該サークルが非公認となる場合は UNIX グループは削除されます。

3. サークル用以外の UNIX グループの代表者の変更

サークル用以外の UNIX グループの代表者がアカウントを失ってしまった場合、アカウント失効後 1 カ月間に限り以下の条件で代表者の変更依頼を認めます。

- UNIX グループの残ったメンバー内で新代表者が承認されていること。この事はメンバー全員に変更依頼のメールを CC することで確認すること。
- UNIX グループの新代表者は教職員であること。

4. 共有ディレクトリ以下のファイルの取り扱い

UNIX グループに対応する共有ディレクトリ以下のファイルの所有者がアカウントを失い、残ったメンバーが当該ファイル操作を出来なくなってしまった場合、以下の条件で **chmod**, **chown**, **chgrp** などの操作の依頼に対応します。

- 共有ディレクトリがサークル用の場合
顧問教員、またはサークル代表学生からの操作の依頼にのみ対応する。ただしサークル代表学生から依頼する場合、依頼メールを顧問教員とグループメンバー全員へ CC しなければならない。
- 共有ディレクトリがサークル用以外の場合
対応する UNIX グループの代表者から操作の依頼にのみ対応する。**chown** の際の変更先は代表者、**chgrp** の際の変更先は当該 UNIX グループとする。

メーリングリストや UNIX グループ、共有ディレクトリのファイルについては、本来このような措置が必要となる前に当事者同士で話し合い、管理者／代表者の変更、ファイル操作などの対処を行なうべきものです。利用者はこれらの対処を依頼しなくても済むよう十分注意して下さい。

各種申請書は「情報処理センター > 各種申請書」にあります。

教室の端末は学内公共の資源であり、占有が許されるものではありません。上記のような処置が行なわれる事のないよう注意して使用して下さい。

学生のディスク使用量について

学生のディスク使用量の限度は以下のとおりとします。

1. 学部学生 : 3.0GB
2. 大学院生(博士前期課程) : 3.0GB
3. 大学院生(博士後期課程) : 3.0GB

各自の使用量を常にチェックしこれらの限度を越えないよう注意して下さい。限度超過が確認された場合は以下の対応を行なうものとします。

*対応方法 : *

- * 容量超過が確認された場合ユーザに対して情報センターは警告のメールを送付する。

ディスク容量の確認方法、不要なファイルの削除方法については「ディスク容量についての注意」<http://web-int.u-aizu.ac.jp/labs/istc/ipc/topic/disk-usage/disk-usage.html>」を確認してください。
また、情報センターでは、ディスク容量不足等への緊急措置として以下の作業を行なう場合があります。

1. 限度を超過している学生を対象とし、ファイル名、ファイルサイズ等のファイル属性の検査を行なう。(ファイルの内容を見る訳ではない)
2. 単一ファイルで使用量限度を超えるものを削除する。
3. core と言う名前のファイルを削除する。
4. 使用量限度を超えるファイルの存在が他の学生の授業などに支障となる場合、別の場所(外部メディアを含む)にファイルを移動する。

以上の対応で生ずる不利益について、情報センターでは補償しませんので了承下さい。

パスワードの頻繁な変更について(2004.7.5)

パスワードの頻繁な変更を行うことにより、より一層のセキュリティ確保を行ないます。最後のパスワード変更から90日以内に変更しないユーザには以下の対応を行なるものとします。(平成16年度第1回情報センター運営委員会にて決定)

*対応方法 : *

1. 最後のパスワード変更から90日以内に変更しないユーザに対して情報セン

ターは警告のメールを送付する。
2. 1から一定期間経過後も変更しないユーザに対してはアカウントロックする。
3. アカウントロックの解除はユーザ本人がSSBに出向き、本人であること及びパスワード変更を確認することにより行うものとする。

不正、不適切な利用事例(2006.2.17)

ユーザの AINS 利用に関して、不正、不適切又は好ましくないものと判断し、指導した事例の一部を紹介します。適切な AINS 利用のための参考として下さい。

-> 不正、不適切な利用事例 (Japanese only) <jirei.html>

更新日: 2017.01.06
問い合わせ先: 情報処理センター
E-mail: ipc@u-aizu.ac.jp

ネットニュース利用に関する注意事項

を計つてください。

<http://web-int.u-aizu.ac.jp/labs/istc/ipc/rules/netnewsguide.html> を以下に記載します。

情報処理センター

| 会津大学 | 学内限定 | Japanese / English

[情報処理センター ネットニュース利用にあたって]

ネットニュース利用にあたって

ローカル・ニュースグループを利用するにあたって、留意すべき点について記述します。適切な利用の参考としてください。

1. 留意点
2. 不適切な使用例

留意点

1. *営利目的利用の禁止*

* ATIS 上での営利活動や商行為は利用目的に関して定めた 会津大学情報センター(情報処理センター) 利用規程(平成 18 年規程第 28 号) の *第 2 条*に違反します。

ニユースグループを利用する場合にも、投稿が営利目的や商行為とならないよう注意してください。

不適切な使用例 を掲載しますので参考にしてください。

* ニュースグループ *u-aizu.wanted* では利用者の生活支援のために以下のような情報がやりとりされています。

- アルバイト情報
- 引っ越しセール等の個人レベルの不用品の売買

これらも使用例を参考として「営利目的利用の禁止」に触れない範囲で利用してください。

2. *目的に合ったニュースグループの利用*

利用目的に合ったニュースグループを選んで利用しましょう。ニュースグループにそぐわない投稿は他の利用者の迷惑になります。
→ローカル・ニュースグループ カテゴリー一覧

3. *グループ管理者の役割*

- ニュースグループ管理者は、そのグループが目的に沿って使われるよう心掛け、必要に応じ議論の誘導やコントロールを行なってください。また、定期的にそのグループの目的や方針を投稿するなどで利用者への周知

4. *利用頻度の低いグループの削除*

情報処理センターでは、利用頻度が低いニュースグループ(6ヶ月間以上投稿のないもの)を削除する場合があります。

不適切な使用例

ローカル・ニュースグループの利用として*不適切な例*を以下に示します。
適切なネットニュースの利用のための参考としてください。

* *例 1 「中古自転車を 30 台格安で譲ります。」「引っ越しセール等個人レベルの不用品」の範疇を越える 大量の物品の販売で「利用者の生活支援」から逸脱しています。また「営利目的の利用」と認められます。

* *例 2 「私の知合いが中古車を売りたがっています。」「利用者の生活支援」から逸脱します。

* *例 3 「居酒屋 XX では現在特別価格期間実施中です。」「お店や会社等のセール、イベントの宣伝、公告 であり、営利目的の利用にあたります。

* *例 4 「XX アパートの入居者の募集の案内です。連絡は XX 不動産へどうぞ」「不動産業者の営業活動の代理投稿で営利目的の利用にあたります。

* 更新日*: 2006.03.24
問い合わせ先: 情報処理センター
E-mail: std-help@u-aizu.ac.jp

\$', 50
\$#argv, 175
\$\$, 51
\$*, 85
\$<, 175
\$argv, 175
\$HOME, 85
\$path, 84
&, 105, 113
*, 78
. , 12, 78
. , 12
.bashrc, 76, 158, 181
.basrc, 180
.csrc, 158, 169
.emacs.d/init.el, 42, 181
.forward, 185
.login, 75
.logout, 181
.mozilla, 155
.plan, 93
.profile, 75, 84, 181
.rhosts, 31, 170
.sympheed-2.0, 26
.xinitrc.mac, 181
.Xresources, 181
| ('ハイフ'), 78, 106, 115
/, 11
<, 115
>, 78, 107, 115
&, 78
>&, 78
?, 78
>>&, 78
Authenticated Network, 32
Authenticity, 67
auto-fill-mode (Emacs), 184
autokill, 200
Availability, 67
awk, 17, 116, 118
\alpha (α), 126
\frac, 126
\infty (∞), 126
\int (' \int '), 126
\pi (π), 126
'('と')' オーバー引用), 79
~, 12

~user, 12
7-bit JIS, 34
7bit, 70

a2ps, 30, 108
a2ps_gnu, 30, 108
abstract, 60

ACM, 60

Accountability, 67

Acrobat Reader, 72, 109, 111

acrobat reader, 30

addtocounter, 19

Adobe, 56, 71, 72, 108, 109

Adobe Reader, 139, 178

ATNS, 48, 68, 199

alias, 76, 81, 175

AMS-LATEX, 19

angelftp, 20

announcement, 120

anonymous, 74

anonymous ftp, 47, 74, 122, 156

aplia, 23

Aqua, 104

archive, 72

argv, 160

arp, 15, 171

ARPANET, 45

array, 129

arxiv, 44

ASCII, 25, 26, 34, 37, 70, 125, 184

aspell, 177

Atok, 35, 98

audio, 70

Authenticated Network, 32

Authenticity, 67

auto-fill-mode (Emacs), 184

autokill, 200

Availability, 67

awk, 17, 116, 118

backslash, 136

barplot (R), 163, 164

bash, 19, 158

bc, 185

beamer, 64, 138, 140, 178

begin ... end, 126

bg, 105, 106, 112, 172

bibliographystyle, 63

bibtex, 63

binary, 69

bit, 34

Blog, 43

bmeps, 30, 58, 70, 150, 151

BoundingBox, 71

bunzip2, 72, 153

Byte, 34

byte-order, 70

bzcat, 72, 153

bzip2, 72, 153

C-/ (Emacs: undo), 96

C-@ (Emacs: mark set), 96

C-d (Emacs: Delete), 96

C-g (Emacs: Cancel), 97

C-i カタカナにする (ATOK), 37

C-i 文節の長さを短くする (kkc_kinput2,SKK, ことえり), 36

C-k (Emacs: Kill line), 96

C-k 文節の長さを長くする (ATOK), 37

C-l 文節の長さを長くする (ATOK), 37

C-n 確定 (ATOK), 37

C-n 次候補を表示する (kkc_kinput2,SKK, ことえり), 36

C-o 半角カタカナ、ローマ字にする (ATOK), 37

C-o 文節の長さを長くする (kkc_kinput2,SKK, ことえり), 36

C-p 前候補を表示する (kkc_kinput2,SKK), 36

C-p 全角ローマ字にする (ATOK), 37

C-r (Emacs: Search Backward), 116

C-s (Emacs: Search Forward), 116

C-Space (Emacs: mark set), 96

C-Space (IIM: trigger key to activate/deactivate), 98

C-w (Emacs: cut region), 96

C-x 2 (Emacs: Split Window), 96

C-x 5 2 (Emacs: Make New Frame), 96

C-x C-c (Emacs: Exit), 97

C-x 1 (Emacs: Insert File), 97

C-x k (Emacs: kill-buffer), 97

C-y (Emacs: yank), 96

C-z, 106

Cache, 154

cal, 15, 95, 113

calc (EMACS), 53

canna, 35, 36
 Carbon Copy, 47, 122
 case, 160
 cat, 25, 78, 99, 100, 112, 115
 cd, 11, 12, 93, 103, 112
 center, 126, 129
 CERT/CC, 167
 CGI, 16
 chfwd, 185
 chmod, 12, 14, 15, 75, 78, 84, 85, 98, 103, 114
 chsh, 158
 CID, 56
 circ, 53
 cite, 62, 63, 129
 cline, 135
 Close, 95
 color, 139
 colorbox, 139
 Command Key, 91
 compress, 72, 153
 Confidentiality, 67
 configure, 177
 convert (ImageMagick), 30, 58, 69, 141, 142, 162
 copyright, 70, 74, 143
 cp, 15, 25, 99, 100, 103, 112
 cpu, 172
 cron, 32
 csh, 19, 158
 csv, 118
 CUPS, 30, 109
 cvs, 186
 Cygwin, 15, 199, 201
 date, 79, 95
 DDNS, 16
 description, 52
 Desktop, 151
 df, 28, 153, 170
 diff, 116
 DISPLAY, 170
 display (ImageMagick), 113, 151
 DNS, 22, 29, 171
 documentclass, 50
 doxygen, 186
 Dropbox, 73
 DTP, 72
 du, 153, 186
 duplex, 30, 109
 dvi, 30, 49, 108
 dvil2tty, 121, 137
 dvipdfmx, 65
 dvips, 18, 30, 49, 58, 59, 65, 108, 139, 180
 echo, 35, 86, 115
 eepic, 58
 egrep, 14
 eject, 73
 eps2eps, 71
 epsffit, 109
 eqarray, 129
 equation, 129, 139
 Esc w (Emacs: copy region), 96
 EUC, 34, 184
 euc-jp, 97
 evince, 30, 50, 72, 110, 113, 139
 exit, 113
 Export (Mathematica), 142, 150
 export, 76, 170
 export (OpenOffice), 179
 export (sh), 84
 Facebook, 43
 FAQ, 43, 120, 185
 fg, 106, 112, 170, 172
 fgrep, 14
 fg2eepic, 58
 figure, 129
 file, 69, 180
 find, 41, 42, 155, 186
 finger, 14, 24, 93, 112
 firefox, 27, 98, 154
 fleamarket, 144
 float (TeX), 59
 flushleft, 126
 flushright, 126
 for, 82
 foreach, 86, 161
 format, 69
 fplot3d (Octave), 145
 FreeBSD, 15
 FS = Field Separator (Awk), 118
 FSF, 17
 ftp, 15, 47, 74, 122, 155, 168, 173, 182
 gawk, 118
 gcc, 80
 gdb, 186
 gdin, 113, 114
 gdm-login, 113
 gedit, 37, 113
 gettent, 15, 17, 19, 24, 107, 112, 158, 169
 Ghostscript, 72
 ghostview, 72
 gif, 142
 gimp, 55, 132
 GNOME, 114
 Gnome Desktop System, 114
 gnome-screenshot, 151
 gnome-terminal, 36, 92, 95, 105, 112, 113, 125
 GNU, 15, 22, 72, 96, 152
 gnuplot, 147, 149
 grus, 27
 GNUTera, 66
 goo, 43
 google, 43
 Google Drive, 73, 163
 Google Groups, 43
 GPL, 17
 GPS, 16
 Grab (ImageMagick), 151
 graphicx, 58, 62, 64
 grep, 15, 25, 40, 82, 90, 106, 112, 115, 116, 118,
 119, 121, 169
 gs, 72
 gtar, 72, 153
 gunzip, 72, 153
 gv, 59, 72, 108, 109, 111, 113, 132, 133
 gzip, 153
 gzip, 72, 153, 177
 head, 25, 112
 Help Browser (Mathematica), 150
 hist (R), 163, 164
 history, 40, 76, 80
 hline, 53, 135
 hspace, 135
 HTML, 9, 182
 hyperref, 139
 IEEE, 60
 IERT, 46
 ifthen (TeX), 53
 IGNOREEOF, 76
 iim, 35, 36, 98, 132
 iimstart, 36, 98
 iimstop, 98
 ImageMagick, 30, 151, 162
 iconv, 58
 includegraphics, 58
 info, 41
 Input Method, 36, 98
 Insert File (Emacs: C-x i), 97
 Integrity, 67
 Internet Explorer, 27
 Internet Protocol, 169
 IP, 29, 169
 ispell, 137
 ispell-buffer, 178
 ispell-region, 178
 ispell-word, 178
 itemize, 52
 iwm, 35

- jararticle, 17, 50, 61
 jbook, 50
 JIS, 34, 184
 jobs, 95, 96, 105, 106, 112, 172
 join, 16, 119
 jpeg, 69, 162
 JUSTSYSTEM, 35
 k2ps, 15, 30, 108, 180
 kill, 95, 106, 112, 172, 175, 180
 kill-buffer (Emacs C-x k), 97
 kinput2, 35, 36, 98, 132
 kkc, 36, 97
 label, 54, 129
 LANG, 93, 176, 178
 LATEX, 9, 18, 19, 49, 108, 111, 125, 130, 138, 180, 191
 latex, 19
 latex-mode, 14
 LDAP, 22, 24, 28
 less, 25, 112
 Linux, 15, 23
 Lisp, 96
 list (LATEX), 19
 ListAnimate (Mathematica), 142
 ln, 15
 localhost, 200
 lock, 15, 155
 login, 24, 89
 logout, 48, 90, 113, 123
 lossless, 72
 lossy, 72
 lower case, 117
 lpq, 29, 108, 113, 180
 lpr, 29, 30, 59, 108, 109, 113
 lprm, 29, 108, 113, 180
 LPRng, 30, 109
 ls, 12, 15, 25, 40, 93, 98, 99, 103, 112, 115, 180
 Mac OS X, 27, 34, 155
 MacPorts, 201
 Macro Virus, 167
 Mac-UIM, 36, 98
 MagicPoint, 74, 178, 179
 /bin/mail, 188
 Mailing List, 185
 /usr/bin/mailx, 188
 make, 177
 Make New Frame (Emacs), 96
 Makefile, 177
 man, 24, 41, 93, 100, 112, 113, 115
 MANPATH, 76
 math, 150, 186
 mathematica, 141, 147, 150, 186
 Matlab, 74, 144
- Microsoft Open Value Subscription-Education Solutions Agreement, 8
 Microsoft Update, 68, 168
 MIME, 25, 71, 144, 184
 MIME Content-Type, 185
 MIME encoding, 71
 mircpage, 135
 mkdir, 12, 93, 103, 112
 mkfwd, 185
 more, 25, 106, 108, 112, 114
 mount, 73
 mp3, 70
 MS-Excel, 118, 176
 ms-imc, 35
 MS-Office, 74, 176, 178
 MS-PowerPoint, 176
 MS-Word, 176
 MTA, 16
 MUA, 15, 20
 multicolumn (LATEX), 18, 135
 MULTICS, 22
 musixtex, 53
 mv, 25, 99, 100, 112
 multicol (LATEX), 18, 135
 ncftp, 155
 Netiquette, 45, 46, 122
 NetNews, 9, 16, 26, 120, 152
 netpbm, 162
 netstat, 15, 171
 Network, 169, 180
 newcommand, 53, 136, 137
 newcounter, 19
 news, 42, 172
 newtxtext, 62, 64
 NF = Number of Fields (Awk), 118
 NFS, 22, 28, 47, 172
 nb1.u-aizu.ac.jp, 28
 nice, 48, 112, 123, 171, 172, 175
 NIS, 16, 22, 28, 107
 nkf, 34, 97
 NWPP, 187
 noindent, 136
 NR = Number of Records (Awk), 118
 nslookup, 29, 171
 OASIS, 176
 Octave, 74, 144
 od, 35
- OFI, 176
 OECD, 67
 Online Storage, 73, 163
 OpenDocument Format, 176
 openldapsearch, 187
 openldapsearch, 28
 OpenOffice, 8, 16, 176, 178, 179
 OpenOffice Presentation, 178, 179
 OpenOffice Spreadsheet, 20
 openssh, 31, 200
 pageref, 54
 par, 136
 paragraph, 61
 ParametricPlot (Mathematica), 142
 parentlock (Firefox), 96
 passwd, 94
 paste, 16, 119
 paste (Emacs), 100
 patch, 186
 PATH, 75, 84, 115, 180
 PDF, 30, 64, 72, 109, 113, 139
 pdf2ps, 30
 pdfbook, 110
 pdffonts, 65
 pdfjam, 110
 pdfjoin, 110
 pdfrnup, 110
 pdftudio, 30, 72, 113
 PDI, 56
 perl, 19, 116, 117, 186
 perl-mode, 14
 permission, 47
 Phishing, 68, 168
 PICT, 70
 PID, 19, 96, 172, 175
 ping, 15, 171
 pipe, 78, 114
 pLATEX, 18
 platex, 19, 49, 113, 125, 139
 Plot (Mathematica), 150
 plot (Gnuplot), 149
 plot (Octave), 145
 plot (R), 164
 plot (maxima), 183
 Plot3D (Mathematica), 150
 plot3d (maxima), 183
 png, 69
 PostScript, 30, 56, 71, 108
 Powerpoint, 178
 PPD, 15, 172
 PPM, 70
 pr, 31
 preview, 50, 71
 previewer, 50

Print Screen, 151
 printenv, 170
 procedure (Make), 177
 prompt, 158
 prosper, 178
 ps, 77, 90, 95, 96, 112, 114, 172, 175
 ps–plot, 140
 PS1, 158
 ps2eps, 58
 ps2pdf, 65, 139
 psbook, 109
 pscircle (TeX), 140
 pscurve (TeX), 140
 psfrag, 58
 psline (TeX), 140
 psmerge, 109
 psnup, 109, 111
 P2P, 66
 pspicture, 58, 140
 psplot (TeX), 140
 psresize, 109
 psselect, 109
 pst-node, 53, 64
 pstops, 109, 111
 PSTricks, 53, 64, 140
 pstricks-add (TeX), 140
 psutil, 109, 111
 pTeX, 18
 pwd, 11, 12, 93, 103, 112
 pxfonts, 64
 quail, 36, 97
 Quiet (Mathematica), 142, 150
 Quit (tgif), 58
 R, 74, 163, 164
 redirect, 78
 ref, 54, 129
 reformime, 71, 185
 Refresh (Window Manager), 106
 regexp, 116
 rehash, 84
 Reliability, 67
 renice, 171
 replace-regexp, 116
 replace-string, 116
 Restart (Window Manager), 106
 RFC, 16, 46, 71, 184
 rlogin, 31, 95, 96, 113, 123, 170
 rm, 25, 40, 99, 100, 103, 112
 rmdir, 12, 14, 93, 103, 112
 root, 103
 rsh, 113, 170
 RSS, 43
 rule, 136
 rusers, 15
 S, 74, 163
 Splus, 74, 163
 sadrv, 199
 Safari, 27
 SciLab, 74, 144
 scp, 20
 sd (R), 164
 section, 18, 61
 section*, 18
 Security, 67, 167
 security hole, 167
 sed, 39, 112, 116
 select, 160
 sendmail, 185
 seq, 17
 set, 19, 84, 170, 175
 setcounter, 62
 setenv, 19, 170
 setuid, 19
 sftp, 31, 155, 168
 SGI, 70
 SGML, 9
 sh, 158
 shift, 161, 175
 Shift-SP (tgif), 132
 shifted-JIS, 34
 Sin (Mathematica), 150
 SJIS, 184
 SRK, 35, 36, 97
 slide, 139
 sniff, 168
 SMS, 43
 softice, 176, 178
 Solaris, 169
 sort, 19, 25, 106, 112, 114, 169
 source, 84
 sox, 70, 162
 SPAM, 26, 168
 spec.org, 16
 spell, 177
 Split Window (Emacs), 96
 splot, 147
 Sort (Mathematica), 150
 SSB, 48, 181
 ssh, 15, 29, 31, 113, 168, 170, 199
 ssh tunnel, 32, 200
 sshgate, 29
 sshgate.u-aizu.ac.jp, 31, 171
 SSI, 32
 std-help, 181
 std-telnet, 181
 stdin, 24, 115
 stdout, 24, 115
 stdskel, 158, 181
 steganography, 144
 Sticky bit, 19
 strings, 116
 su, 17
 subparagraph, 61
 subsection, 61
 subsubsection, 61
 Sunray, 16, 29, 73
 Switch, 160
 Syipheed, 98
 Table (Mathematica), 142
 table (R), 163, 164
 tableofcontents, 62
 tabular, 18, 52, 135, 136, 182
 tabularx (TeX), 135
 tail, 25, 112
 tar, 72, 153, 177
 target (Make), 177
 TCP/IP, 16, 45
 tcsch, 19, 158
 telnet, 123, 168, 170
 terminal, 104, 105, 106
 test, 83, 86
 TeX, 9, 108
 textcolor, 139
 textwidth, 136
 tgif, 53, 55, 56, 58, 59, 132, 133
 thebibliography, 129
 Thunderbird, 101, 184
 tiff, 30, 70
 time, 15
 toodepth, 62
 top, 112, 170, 172, 175
 tr, 117
 traceroute, 15, 171, 173
 tramp, 20
 Trojan Horse, 167
 tumble, 30, 109
 Tutorial (Emacs), 97
 Tutorial (Mathematica), 150
 Twitter, 43
 Type1 font, 65
 typist, 92
 u-aizu.announce, 42, 120
 u-aizu.announce.gakusebu, 42, 120, 172
 u-aizu.course.litl, 89
 u-aizu.istc, 42, 120, 169, 172
 u-aizu.student.all-student, 42
 UBE, 168
 UID, 172
 umask, 75
 unalias, 81
 uncompress, 72, 153
 underline, 18
 Unicode, 34
 UNIX, 22
 unset, 79
 unxz, 72, 153

- unzip, 153
 Upper case, 117
 USB, 73
 USB Memory, 72, 163
 usepackage, 58, 62, 135
 utdiskadm, 73
 UTF-8, 34
 UTF-8 Mac, 34
 uuencode, 71, 143, 144, 184
 var (R), 164
 verb, 136
 verbatim, 126, 129
 vi, 15, 113
 VirtualBox, 201
 Virus Hoax, 167
 VNC, 31, 200
 VNC server, 23
 vncpasswd, 200
 vncviewer, 200
 vspace*, 53, 135, 136
 vulnerability, 167
 w3m, 27
 wc, 25, 84, 85, 99, 100, 113
 Web Browser, 73
 web-int.u-aizu.ac.jp, 28, 173, 182
 websfs-ext.u-aizu.ac.jp, 173
 webfs-int.u-aizu.ac.jp, 173, 182
 wget, 27, 155
 which, 79, 85, 113, 115, 180
 who, 24, 84, 95, 112, 113
 whoami, 24, 95, 112
 WiFi, 32
 Wikipedia, 43
 Window Manager, 106, 155
 Windows, 27, 34, 155
 Windows OneDrive, 73, 163
 WinMX, 66
 Wireless LAN, 32
 wmm, 35
 World Wide Web, 172
 worm, 167
 WWW, 16
 www.u-aizu.ac.jp, 28, 173
 www.w3.org, 182
 WYSIWIG, 49
- xdvi, 102, 113
 xeyes, 15, 95, 105, 106, 113, 171
 xf86, 53, 55, 132
 xhost, 171
 xim, 35
 xinit, 31
 xlock, 31, 48, 123, 180
 xlsclients, 95
 xmgrace, 17, 147, 149
 xpdf, 30, 50, 72, 113
 xterm, 95, 112
 Xvnc, 200
 xwininfo, 105
 xz, 72, 153
 xzcat, 72, 153
 Yahoo, 43, 73, 163
 zcat, 72, 153
 zip, 153
 zsh, 19, 158
 zw, 135
- アーカイブ, 72, 153
 アイコン, 104, 105
 会津グリーン, 15
 会津大学 WWW サーバ, 173
 IPアドレス, 169
 アウトラインプロセッサ, 62
 アウトラインモード, 178
 アカウント, 89, 94
 アカウント停止, 185
 アカウントロック, 48, 94, 181
 アクセス許可, 114, 123
 アクセス記録, 15
 アクセス制限, 98
 アクセス制限管理, 98
 アクセスベースミッション, 122
 アクセント記号, 126
 アクティブarend, 95
 ASCII 文字列, 184
- 圧縮, 72, 153
 圧縮方法, 70
 アップロード, 173
 アニメーション, 141
 アブストラクト, 60
 アンダーライン, 18
 アンドウー (Emacs: C-), 96
 アンバーソンド, 113
 イーサネット, 29
 異常値, 164
- 印刷, 29, 108
 印字可能, 70
 引用, 62, 123
 ワイルズ, 68
 ウィンドウマネージャ, 152
 ウィンドウメニュー, 104
- xdvi, 102, 113
 ウエブ掲示板, 27
 ウエブサーバ, 16, 28, 173, 182
 ウェブラウザ, 27, 173, 182
 SSH トンネル, 32
 ssh トンネル, 200
 エスケーピング, 81
 エラーメール, 26, 185
 エンコード, 25
 円周率, 17
 オーナー, 12
 オープンソーシュティング, 17
 オープンソース, 176
 オープンソースソフトウェア, 17
 観測データ, 12
 音声データ, 70
 音声フォームット, 70
 オンラインストレージ, 73, 163
 オンライスマニアブル, 24, 105, 112
- カーネル, 96
 カーネル, 158
 改行コード, 35, 137
 書き込み許可, 12, 114
 学術情報センタ, 144
 学情ネット, 144
 拡張子, 49, 153
 拡張 UNIX コード, 97
 業譜, 53
 学務システム, 27, 120
 仮想化技術, 201
 画像データ, 69
 画像フォーマット, 69
 フタカナ (ATOK), 37
 カタカナ (kkc), 36
 かな漢字変換, 35
 かな漢字変換クライアントプログラム, 35
 仮のパスワード, 90
 カレントディレクトリ, 11, 12
 カレントワーキングディレクトリ, 112
 環境変数, 75, 76, 115, 170
 関係演算子, 159
 漢字コード, 34
 漢字サーバ, 28
 記号 (kinput2), 36
 逆行列, 145, 183
 キヤブチヤ, 17
 キヤンセル, 101
 域面調和関数, 20
 行頭, 39, 116
 行番号, 17
 行列, 145, 183

- 行列式, 145, 183
 空白, 18
 クオーティング, 81
 区切り文字, 39, 81
 矩形領域削除, 15
 組版, 49
 クライアント, 29
 クラッカー, 10
 グラフ理論, 53
 クリック, 105
 グループ, 98, 114
 クロスポスト, 101
 揭示板, 16, 144
 携帯電話, 26, 184, 185
 経路情報, 171
 ゲストアカウント, 156
 ゲストユーザ, 156
 検索エンジン, 43
 後方検索, 116
 互換 (S), 163
 互換性 (OpenOffice), 178
 互換性 (Postscript), 72
 國際標準, 22
 國立国会図書館, 44
 個人情報, 144
 ことえり, 36, 98, 132
 コンバート, 30, 108
 コンパイル, 49
 サーバ, 29
 最小化ボタン, 104
 最小値, 164
 最大化ボタン, 104
 最大値, 164
 作業ディレクトリ, 11
 查読, 60
 サブネット, 169
 サムネイル, 178
 3次元プロット, 145, 183
 残留プロセス, 90
 シエル, 158
 シエル変数, 75, 159
 シグネチャ, 122
 JIS コード文字, 125
 システム管理者, 89
 自治体, 16
 実行許可, 12
 実行優先度, 171
 修飾子, 80, 81
 出力装置, 114
 肖像権, 66, 143
 情報記録メディア, 164
 情報検索, 27
 情報処理, 114
 情報センター利用の手引, 181
 使用優先度, 48
- 職業倫理, 9
 書庫, 153
 シヨフ番号 (印刷), 108
 伸張, 153
 森林資源, 109, 110
 数式モード, 50
 スーパーユーパー, 89
 スキナー, 15
 スクリプトファイル, 117
 図形描画モード, 178
 ストリークエディタ, 116
 スパイエア, 68
 スペムメール, 168
 スペルチェック, 177
 スライド, 64
 スライドショウ, 179
 スライドモード, 178
 正規表現, 37, 38, 116
 正規分布, 164
 脆弱性, 167
 生体情報, 16
 セキュリティ, 10, 67
 セキュリティバッチ, 168
 絶対パス, 11, 115
 絶対パス名, 115
 全角文字, 25
 前方検索, 116
 増減表, 18
 相互参照, 54, 55
 送信可能化権, 143
 相対バス, 11
 ソース, 126
 属性, 114
 草業論文, 17
 外付けHDD, 72
- 2 バイトコード, 122
 2 バイト文字, 184
 日本工業規格, 34
 日本語入力, 178
 日本語モード, 176
 ドラッグ, 104
 ドメイン名, 25, 29
 ドラッグ&ドロップ, 182
 ドラブル効果法, 182
 ドラブルレポート, 181
- 日本語モード (OpenOffice), 178
 ニュース, 172
 ニュースサーバ, 28
 人力装置, 114
 認証システム, 16
 認証ネットワーク, 32, 48
 ねぎみ講, 66, 144
 ネットニュース, 26, 27, 101
 ネットワーク障害, 17
 ネットワークブリッジ, 28, 29, 108
 ネットワーク利用規定, 68
 ノーウィンドウオプション, 170
 ノートブック (Mathematica), 150
 ノートモード, 178
- パーティション, 12, 99
 背景色, 15
 バイナリファイル, 69, 116
 バイブ, 24, 106, 114
 パス名, 11
 パスワード, 10, 89, 90, 94
 パスワードの有効期限, 123
 バックアップ, 40, 73, 123, 163
 バックグラウンド, 105
 バックグラウンド, 76
 バックグラウンド, 105

- バックグラウンド処理, 17, 106, 113
 バックドア, 167
 バッケージ, 62
 バッヂ処理 (R), 164
 バッヂ方式, 49
 ハンドアウト, 110, 179
 ハンドアウトモード, 178
 PDF ファイルの印刷, 30
 引数, 81, 85
 ヒストограм, 20, 163, 164
 表組, 135
 標準環境, 158
 標準環境設定ファイル, 181
 標準出力, 24, 78, 115
 標準設定, 170
 標準入出力, 114
 標準偏差, 164
 フアイアーオール, 68
 フアイヤウオール, 168
 ファイル形式, 162
 ファイルサーバ, 170
 ファイルシステム, 115
 ファイル転送プロトコル, 155
 ファイルリストア, 40
 ファイル名展開, 82
 フィールド, 11
 フィッシング, 68, 168
 フィルタ, 24, 107
 フィルタリング, 107
 フリー工成分, 20
 フォアグランドジョブ, 106
 フォアグランド, 170
 フォーマット, 31
 フォーマット, 69
 フォントサイズ, 92
 フォントサイズ (Mathematica), 20
 複製, 179
 不正アバシー, 66
 ブラグ, 93
 ブラットフォーム, 70
 プリインブル, 50, 58, 62, 135, 137
 プリントサーバ, 29
 プルパス, 115
 プレゼンテーション, 138, 139, 178, 179
 パーツ方式, 49
 ハンドアウト, 110, 179
 ハンドアウトモード, 178
 PDF ファイルの印刷, 30
 平均値, 164, 176
 ページ記述言語, 56, 71
 ページソース, 182
 ペースト (Emacs), 96
 別名, 81, 169
 ポイント, 50
 法律, 46, 66
 法令データ提供システム, 67
 ポートレート, 30, 109
 ホームディレクトリ, 11, 12, 172
 ホームページの改竄, 167
 ホームボジョン, 92
 ポストスクリプト, 30, 108, 113
 ホスト名, 169
 保存形式, 69
 ポップアップ, 104, 105
 マーク (Emacs), 96, 100
 マークアップ言語, 49, 182
 マウス, 95
 マウスカーソル, 104
 マクロ, 53, 62, 136
 マナー, 46
 マルチタスク, 77, 112
 マルチポスト, 101
 マルチユーザー, 77, 112
 ミラーサイト, 47, 122
 民事訴訟, 143
 無償, 22
 無線 LAN, 16, 32, 168
 名誉毀損, 66, 143
 迷惑メール, 26
 メーリングリスト, 26, 42, 185
 メールアドレス, 28
 メールサーバ, 28
 メール転送 (メールフォワード), 185
 メール配達失敗, 185
 メールヘッダ, 184
 メタキャラクタ, 37, 78, 81
 免責事項, 73, 163
 目次, 18
 文字列検索, 116
 文字列置換, 116, 117
 プロバイダ, 67
 プロンプト, 158
 文献リスト, 60
 分散, 164, 176
 ページ記述言語, 56, 71
 ページソース, 182
 ペースト (Emacs), 96
 ライセンス, 17
 亂数, 17
 ランドスケープ, 30, 109
 リージョン (Emacs), 100
 リストア, 40, 163
 リダイレクト, 15, 24, 78, 106, 114
 リモートログイン, 31, 170
 利用規程, 68
 両面印刷, 30, 109
 リリース, 104
 履歴, 76, 80, 165
 リンク, 85, 112
 論理, 9
 累積度数分布, 164
 ルータ, 15
 ルート, 11, 89
 ルートウインドウ, 95, 151
 ルートディレクトリ, 11
 建立一次方程式, 145, 183
 ローカルファイル, 172
 ログアウト, 89, 90, 91
 ログインウインドウ, 95
 ログインサービス, 113
 ログインシェル, 19, 165
 ログインスクリーン, 17, 89, 91, 95, 113, 114
 ログイン名, 89
 論文, 60
 論理構造, 9, 49, 60
 猿藝物陳列, 144
 猿藝物陳列罪, 66
 ワイルドカード, 40, 78