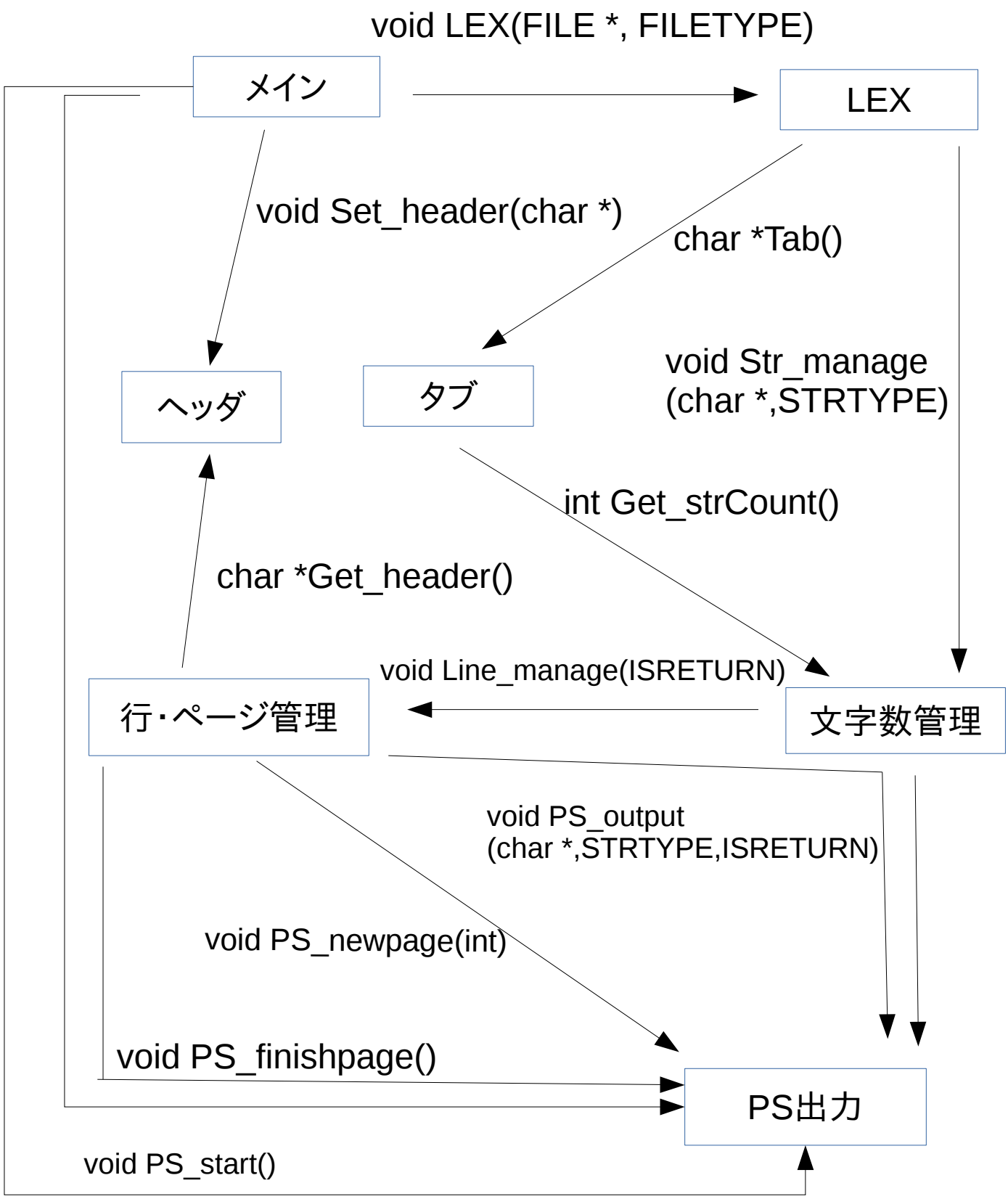
モジュール間関連図



enum型一覧

FILETYPE・・・{CFILE,TXTFILE}
STRTYPE・・・{ENG, ENG_B, JPN}
ISRETURN・・・{RETURN, NO_RETURN}

モジュール外部仕様書

メインモジュール

システム名:テキストフォーマッタ
モジュール名:メインモジュール
作成者:加藤拓洋
作成年月日:2018 年 6 月 28 日
最終更新:2018 年 7 月 12 日
Version:1.1

概要:
メインモジュールはヘッダモジュールへファイル名を渡し, LEX モジュールを呼び出す.

機能・処理:
コマンドライン引数を調べ,引数が指定されていた場合はそれを入力ファイル名としてファイルを開き,PS 出力モジュールの関数 PS_start を呼び出して Postcript マクロを出力する。そしてファイル名の文字列ポインタを引数としてヘッダモジュールの関数 Set_header を呼び出す.また,そのファイルの拡張子からファイルが C ファイルなのかそれ以外のファイルなのかを判断する.開いたファイルポインタを第 1 引数に,ファイルの型の情報を第 2 引数 (C ファイルなら CFILE, 標準ファイルなら TXTFILE) として LEX モジュールの関数 LEX を呼び出す.コマンドライン引数が指定されていなかった場合は PS 出力モジュールの関数 PS_start を呼び出したあと、ファイル名を noname としてその文字列のポインタを引数として、ヘッダモジュールの関数 Set_header を呼び出し, 標準入力ファイルポインタを第 1 引数に, TXTFILE を第 2 引数として LEX 関数を呼び出す. LEX モジュールの処理が終了したら PS 出力モジュールの関数 PS_finishpage を呼び出したのち OK_EXIT を返し, 処理を終了する.

インタフェース:
int main(int argc, char *argv[])

入力と出力:
ファイル名が入力される.
出力なし

前提:
引数で指定されたファイルが存在する.

エラー処理:
引数で指定されたファイルが存在しなかった場合、ファイル名が日本語であった場合、もしくはコマンドライン引数が正しく与えられなかった場合は標準エラー出力にエラーメッセージを出力して終了する.このとき返り値は ERR_EXIT とする.

内部構造:
・大域変数 (型, 名前, 初期値)

・静的変数 (型, 名前, 初期値)

・列挙定数　(名前, 値)
 OK_EXIT　0
 ERR_EXIT　1

FILETYPE
 CFILE　　0
 TXTFILE　1

・ヘッダファイル (ファイル名，構成要素)

- enum.h
  列挙型の定義を含む
- lex.h
  LEX モジュールの関数プロトタイプ宣言を含む
- header.h
  ヘッダモジュールの関数プロトタイプ宣言を含む
- ps.h
  PS 出力モジュールの関数プロトタイプ宣言を含む

Caller:
LEX モジュール
ヘッダモジュール
PS 出力モジュール

Callee:
なし

コメント:
機能・処理について PS_start 関数に関する記述を追記、加えてエラー処理を追加(7 月 12 日)

ヘッダモジュール

システム名:テキストフォーマッタ
モジュール名:ヘッダモジュール
作成者:加藤拓洋
作成年月日:2018 年 6 月 28 日
最終更新:2018 年 7 月 26 日
Version:1.2

概要:
ヘッダモジュールはメインモジュールと行･ページ管理モジュールによって呼び出される.

機能･処理:
ヘッダモジュールの関数 Set_header が呼ばれると引数の文字列ポインタを受け取って取得した文字列をファイル名とし , gecos フィールドから日付, ユーザ名を取得する.その後, ヘッダモジュールの関数 makeheader にこれらの引数を渡してファイル名, 日付, ユーザ名を 1 つにしたヘッダー用の文字列を作成し, headname に格納する.
行･ページ管理モジュールがヘッダモジュールの関数 Get_header を呼び出した際に headname を返す.

インタフェース:
void Set_header(char *)
char *Get_header()

入力と出力:
入力なし
出力なし

前提:
メインモジュールから文字列を受け取っていること.
gecos フィールドから日付とユーザ名を受け取っていること.

エラー処理:
なし

内部構造:
･大域変数 (型, 名前, 初期値)
char　　*headername = NULL

･静的変数 (型, 名前, 初期値)

･関数 (型, 名前, パラメータ)
char　　makeheader(char filename , char date , char username)
void　　Set_header(char *)
char　　*Get_header()

･列挙定数(名前, 値)
なし

･ヘッダファイル (ファイル名, 構成要素)
- header.h
　ヘッダモジュールの関数プロトタイプ宣言を含む.
Caller:
なし

Callee:

メインモジュール
行･ページ管理モジュール

コメント
変数 headername 静的変数から大域変数へ変更(7 月 5 日)
列挙定数に関する記述を削除(7 月 26 日)

LEX モジュール

システム名: テキストフォーマッタ
モジュール名: LEX モジュール
作成者: 坂田悠馬
作成年月日: 2018 年 6 月 28 日
最終更新: 2018 年 6 月 28 日
Version: 1.2

概要:
LEX モジュールはファイルから入力を読み込み、文字列ごとに状態遷移をして、文字列を文字数管理モジュールの関数 Str_manage に渡す。

機能・処理:
メインモジュールからファイルポインタとファイルの形式 FILETYPE を引数として関数 LEX が呼び出された場合、そのファイルポインタから文字列を読み込む。
まず、FILETYPE が CFILE であれば C ファイルの状態に、TXTFILE であればテキストファイルの状態に遷移する。テキストファイルであれば、特殊文字か、それ以外の半角文字か、全角文字か、タブか、改行かで場合分けをし、半角であれば STRTYPE を ENG として文字数管理モジュールの関数に Str_manage に文字列と STRTYPE を渡し、全角文字であれば JPN として文字列を渡す。特殊文字ならば、文字列の先頭に\(バックスラッシュ)を追加して STRTYPE を ENG として渡す。改行ならばそのまま STRTYPE を ENG として渡す。タブが来たなら、タブモジュールを呼び出し、その返り値を文字数管理モジュールの関数 Str_manage に引数として送る。STRTYPE は ENG とする。
C ファイルであれば、上記の処理に加えて、予約語やコメント、クオーテーションで囲まれた文字列がある。予約語があれば、STRTYPE を ENG_B として予約語の文字列を渡す。コメントや文字列が来た場合は、新たな状態として、その間はテキストと同じように扱い、終了したらもとの C ファイルの状態に戻る。
そうしてファイルの最後まで入力が終わったら処理を終了する。

インターフェース:
void LEX(FILE *, FILETYPE)

入力と出力:
ファイルから文字列が入力される。
出力はなし。

前提:
ファイル内の全角文字は EUC で与えられる。

内部構造:
・大域変数(型、名前、初期値)

・静的変数(型、名前、初期値)

・関数(型、名前、パラメータ)
void LEX(FILE *)

・列挙定数(名前、値)
FILETYPE
  CFILE 0
  TXTFILE 1

STRTYPE
  ENG 0

ENG_B 1
JPN 2

・ヘッダファイル(ファイル名、構成要素)
― lex.h
LEX モジュールの関数プロトタイプ宣言を含む

― tab.h
タブモジュールの関数プロトタイプ宣言を含む

― strmanage.h
文字数管理モジュールの関数プロトタイプ宣言を含む

― enum.h
列挙型の定義を含む

Caller:
タブモジュール、文字数管理モジュール

Callee:
メインモジュール

コメント:
LEX を用いて実現すること。

タブモジュール

システム名： テキストフォーマッタ
モジュール名： タブモジュール
作成者： 小杉駿介
作成年月日： 2018 年 6 月 28 日
最終更新： 2018 年 6 月 28 日
Version： 1.2

概要：
タブモジュールはファイルに水平タブ(\t)があった場合に、文字数に応じたスペースを出力する。

機能・処理：
LEX モジュールにおいて関数 Tab が呼び出された場合に、文字数管理モジュールの関数 Get_strCount を呼び出し、現在の行の文字数を受け取り、文字数が 8 の倍数となるようにスペースを返す。

インタフェース：
char *Tab()

入力と出力：
入力：なし
出力：なし

前提：

エラー処理：

内部構図：

・**大域変数**(型、名前、初期値)

・**静的変数**(型、名前、初期値)

・**関数**('型、名前、パラメータ)
char *Tab()

・**列挙関数**(名前、値)

・**ヘッダファイル**(ファイル名、構成要素)
- tab.h
関数 Tab のプロトタイプ宣言を含む。
- strmanage.h
関数 Get_strCount のプロトタイプ宣言を含む。

Caller:
文字数管理モジュール

Callee:
LEX モジュール

コメント：
特になし。

文字数管理モジュール

システム名: テキストフォーマッタ
モジュール名: 文字数管理モジュール
作成者: 坂田悠馬
作成年月日: 2018 年 6 月 28 日
最終更新: 2018 年 6 月 28 日
Version: 1.2

概要:
文字数管理モジュールは文字列を 1 行分に分割して PS 出力モジュールの関数 PS_output に渡す。また、行・ページ管理モジュールの関数 Line_manage を呼び出す。

機能・処理:
文字数管理モジュールは現在の行の何文字目に位置しているかを静的変数 strcnt として持っている。

LEX モジュールから文字列とフォントを決める STRTYPE を引数として関数 Str_manage を呼ばれた場合、現在の行の文字数が 0 であれば行・ページ管理モジュールの関数 Line_manage を呼び出す。そして引数として与えられた文字列を 1 行分である 80 文字をオーバーしないように分割する。もし 80 文字以上ならば、こえるまでの文字列と同時に初めに与えられた STRTYPE と改行を示す RETURN を引数として PS 出力モジュールの関数 PS_output を呼び strcnt を 0 に戻す。改行がきた場合は空文字列として同じ処理をする。そして残りの文字列についてもループをして同様に行う。もし 80 文字に満たない場合は、strcnt を更新して、文字列と STRTYPE と改行をしないことを示す NO_RETURN を引数として PS 出力モジュールの関数 PS_output を呼び処理を終了する。残りの文字列が空列となった場合はループを終了する。
STRTYPE が ENG や ENG_B であれば 1 文字を 1 とカウントし、JPN であれば 1 文字を 2 とカウントする。

タブモジュールから関数 Get_strCount が呼ばれた場合、strcnt を返す。

インターフェース:
void Str_manage(char *, STRTYPE)
int Get_strCount()

入力と出力:
なし

前提:
Str_manage の引数として与えられる文字列は空文字列ではない。またタブも現れない。文字列の最後は改行で終わらない。

エラー処理:

内部構造:
・大域変数(型、名前、初期値)

・静的変数(型、名前、初期値)
  int strcnt = 0

・関数(型、名前、パラメータ)
  void Str_manage(char *,STRTYPE)
  int GetCount()

・列挙定数(名前、値)
― STRTYPE

ENG 0
ENG_B 1
JPN 2

― ISRETURN
RETURN 0
NO_RETURN 1

・ヘッダファイル(ファイル名、構成要素)
― strmanage.h
文字数管理モジュールの関数プロトタイプ宣言を含む

― linemanage.h
行・ページ管理モジュールの関数プロトタイプ宣言を含む

― ps.h
PS 出力モジュールの関数プロトタイプ宣言を含む

― enum.h
列挙型の定義を含む

Caller:
行・ページ管理モジュール、PS 出力モジュール

Callee:
タブモジュール、LEX モジュール

コメント:
とくになし

行・ページ管理モジュール

システム名： テキストフォーマッタ
モジュール名： 行・ページ管理モジュール
作成者： 小杉駿介
作成年月日： 2018 年 6 月 28 日
最終更新： 2018 年 7 月 26 日
Version： 1.4

概要：
行・ページ管理モジュールは改行したときに行の初めに行番号を出力し、改ページした場合はヘッダを出力する。

機能・処理：
文字列管理モジュールから ISRETURN を引数とした関数 Line_manage が呼び出された場合に、行数をカウントする変数 LineCount の値が 0 の場合、ページ数をカウントする変数 PageCount をインクリメントし、変数 PageCount を引数として PS 出力モジュールの関数 PS_newpage を呼び出して次ページの初期化をしたのち、ヘッダモジュールの関数 Get_header を呼び出してヘッダを格納する文字列を取り出し、ページ数を追加してヘッダを出力する。そして変数 LineCount の値をインクリメントする。もし引数が RETURN の場合は、変数 LineNumber の値をインクリメントし、行番号:と ENG、NO_RETURN を引数として PS 出力モジュールの関数 PS_Output を呼び出し、行の初めに行番号を出力する。もし引数が NO_RETURN の場合は、半角 7 文字分のスペースと ENG、NO_RETURN を引数として PS 出力モジュールの関数 PS_Output を呼び出す。行数が 58 を越えた場合、PS 出力モジュールの関数 PS_finishpage を呼び出し改ページをして、変数 LineCount の値を 0 にする。

インタフェース：
void Line_manage(ISRETURN)

入力と出力：
入力:なし
出力:なし

前提：
文章が終了したときの処理はメインモジュールで行う。

エラー処理：

内部構図：

・**大域変数**(型、名前、初期値)

・**静的変数**(型、名前、初期値)
int LineCount=0
int LineNumber=0
int PageCount=0

・**関数**('型、名前、パラメータ)
void Line_manage(ISRETURN)

・**列挙定数**(名前、値)
STRTYPE:
ENG=0
ENG_B=1
JPN=2
ISRETURN:
RETURN=0

NO_RETURN=1

・**ヘッダファイル**(ファイル名、構成要素)

- linemanage.h
関数 Line_manage のプロトタイプ宣言を含む。
- ps.h
関数 PS_Output、PS_newpage、PS_finishpage のプロトタイプ宣言を含む。
- header.h
関数 Get_header のプロトタイプ宣言を含む。
- enum.h
列挙体 STRTYPE、ISRETURN の定義を含む。

Caller:
PS 出力モジュール、ヘッダモジュール

Callee:
文字列管理モジュール

コメント:
Line_manage に引数 ISRETURN を追加(7 月 12 日)
静的変数に LineNumber を追加、および機能・処理の内容を変更(7 月 26 日)

PS 出力モジュール

システム名:テキストフォーマッタ
モジュール名:　PS 出力モジュール
作成者:　岡崎真治
作成年月日:　2018年 6 月 28 日
最終更新:　2018年 7 月 12 日
Version: 1.2

概要:
出力モジュールは受け取った文字を PostScript 形式で標準出力する。

機能・処理:
PS_output 関数の引数から受け取った文字を、クーリエ(C)、クーリエボールド(B)、明朝体(K)のいずれかで PostScript 形式で標準出力する。出力する引数で受け取った文字は括弧で囲み、引数で受け取ったフォントの種類、マクロ命令の s を入力する。引数から改行をする命令を受け取った場合はマクロ命令の n を付け加える。もし引数で受け取った文字が空文字の場合 n を出力する。
PS_newpage 関数では%%Page: <label> <ordinal>を出力する。label と ordinal には引数から受け取った整数を対応させる。そしてページ初期化コマンドの b を入力する。
PS_finishpage 関数では改ページコマンドの f を入力する。
PS_start 関数では、標準出力に PostScript のマクロを出力する。

インターフェース:
void PS_output(char*,STRTYPE,ISRETURN)
void PS_newpage(int)
void PS_finishpage()
void PS_start()

入力と出力:
PostScript 形式で出力される。

前提:
引数で受け取る文字とフォントの種類、改行の有無が存在する。

エラー処理:

内部構造:
・大域変数 (型、名前、初期値)
・静域変数 (型、名前、初期値)
・列挙定数 (名前、値)
STRTYPE:
ENG　0
ENG_B　1
JPN　2

ISSRETURN:
RETURN=0
NO_RETURN=1

・ヘッダファイル (ファイル名、構成要素)
-enum.h
列挙定数 ENG,ENG_B,JPN,RETURN ,NO_RETURN の定義を含む

-ps.h
関数 PS_output、PS_newpage、PS_finishpage、PS_start のプロトタイプ宣言を含む

Caller:
なし

Callee:
行・ページ管理モジュール
文字数管理モジュール
メインモジュール

コメント:
PS_start に関する記述を追加(7 月 12 日)