

**1030.** We shall concentrate first on the inner loop of *main\_control*, deferring consideration of the other cases until later.

```

define big_switch = 60 { go here to branch on the next token of input }
define main_loop = 70 { go here to typeset a string of consecutive characters }
define main_loop_wrapup = 80 { go here to finish a character or ligature }
define main_loop_move = 90 { go here to advance the ligature cursor }
define main_loop_move_lig = 95 { same, when advancing past a generated ligature }
define main_loop_lookahead = 100 { go here to bring in another character, if any }
define main_lig_loop = 110 { go here to check for ligatures or kerning }
define append_normal_space = 120 { go here to append a normal space between words }

⟨ Declare action procedures for use by main_control 1043 ⟩
⟨ Declare the procedure called handle_right_brace 1068 ⟩
procedure main_control; { governs TEX's activities }
  label big_switch, reswitch, main_loop, main_loop_wrapup, main_loop_move, main_loop_move + 1,
    main_loop_move + 2, main_loop_move_lig, main_loop_lookahead, main_loop_lookahead + 1,
    main_lig_loop, main_lig_loop + 1, main_lig_loop + 2, append_normal_space, exit;
  var t: integer; { general-purpose temporary variable }
  begin if every_job ≠ null then begin_token_list(every_job, every_job_text);
big_switch: get_x_token;
reswitch: ⟨ Give diagnostic information, if requested 1031 ⟩;
  case abs(mode) + cur_cmd of
    hmode + letter, hmode + other_char, hmode + char_given: goto main_loop;
    hmode + char_num: begin scan_char_num; cur_chr ← cur_val; goto main_loop; end;
    hmode + no_boundary: begin get_x_token;
      if (cur_cmd = letter) ∨ (cur_cmd = other_char) ∨ (cur_cmd = char_given) ∨ (cur_cmd = char_num)
        then cancel_boundary ← true;
      goto reswitch;
    end;
    hmode + spacer: if space_factor = 1000 then goto append_normal_space
      else app_space;
    hmode + ex_space, mmode + ex_space: goto append_normal_space;
    ⟨ Cases of main_control that are not part of the inner loop 1045 ⟩
  end; { of the big case statement }
  goto big_switch;
main_loop: ⟨ Append character cur_chr and the following characters (if any) to the current hlist in the
  current font; goto reswitch when a non-character has been fetched 1034 ⟩;
append_normal_space: ⟨ Append a normal inter-word space to the current list, then goto big_switch 1041 ⟩;
exit: end;

```

**1031.** When a new token has just been fetched at *big\_switch*, we have an ideal place to monitor T<sub>E</sub>X's activity.

```

⟨ Give diagnostic information, if requested 1031 ⟩ ≡
if interrupt ≠ 0 then
  if OK_to_interrupt then
    begin back_input; check_interrupt; goto big_switch;
    end;
  debug if panicking then check_mem(false); gubed
  if tracing_commands > 0 then show_cur_cmd_chr

```

This code is used in section 1030.