

```

with Ada.Text_IO;
with Ada.IO_Exceptions;
with Ada.Command_Line; use Ada.Command_Line;

with a_lexic; use a_lexic;
with a_sintactic; use a_sintactic;

with decls; use decls;
with decls.d_tnomms; use decls.d_tnomms;
with decls.d_tsimbols; use decls.d_tsimbols;

with semantica; use semantica;
with semantica.c_arbre; use semantica.c_arbre;
with semantica.c_tipus; use semantica.c_tipus;
with semantica.g_codi_int; use semantica.g_codi_int;
with semantica.g_codi_ass; use semantica.g_codi_ass;

procedure main is
  function substring(s: in string; sc: in character) return String is
    i: integer;
  begin
    i:= s'First;
    while i < s'Length and then s(i) /= sc loop
      i:= i+1;
    end loop;
    if i < s'Length then
      i:= i-1; --eliminam sc
    end if;
    return s(s'First..i);
  end substring;

  i: natural;
  err: boolean;
begin
  i:= Argument_Count;
  if i = 0 then
    Ada.Text_IO.Put_Line("No s'ha especificat quin fitxer processar");
  elsif i > 1 then
    Ada.Text_IO.Put_Line("Sols es pot processar un fitxer alhora");
  else
    -- Buidam ts i tn, inicialitzam nv, np, ne i preparam la generacio
    -- de codi intermitg i assemblador
    semantica.prepara_analisi(substring(Argument(1), '.'));
    a_lexic.Open(Argument(1));
    YYParse;
    a_lexic.Close;
    semantica.c_tipus.comprovacio_tipus(err);
    if not err then
      semantica.g_codi_int.gen_codi_int;
      semantica.g_codi_ass.gen_codi_ass;
    end if;
    conclou_analisi;
  end if;
exception
  when Ada.IO_Exceptions.Name_Error =>
    Ada.Text_IO.Put_Line("No s'ha trobat el fitxer especificat '"&Argument(1)&"'");
end main;

```