

Compiler Construction

BPDC

Lab - 08
(Weightage 0%)

I/O Functions

Incorporate `read(scanf)` and `write(printf)` functions to your compiler.

1. The template of *scanf* function is: *scanf*("formatstring", var1, var2, ...); where the format string should contain the right sequence of format specifiers, %d(int), %c(char), %f(float), %lf(double) or %s(character array) that exactly matches the data types of those associated variables (this has to be verified by referring to the symbol table). If the user declares a variable as a character array (say, *char str*[30];), and refers to it using the format specifier %s, then this should be treated as an exceptional case and the user should be permitted to refer to the character array using the name alone (say, *scanf*("%s", *str*); is valid), and otherwise should be forced to refer using indexing (say, *scanf*("%c", &*a*[i])).
2. Further, for all those format specifiers other than %s, *scanf* requires to be provided with the address of the variable (*scanf*("%c%d", &*str*[i], &*x*); and *scanf*("%s", *str*); are valid expressions while *scanf*("%d", *x*); is an invalid expression assuming *x* to be of type *int* and *str* to be a *char array*).
3. With reference to Lab 06, in general, the parameters (other than the first which is format specifier string) to *scanf* should have pointer depth 1 and should be of the appropriate type.