

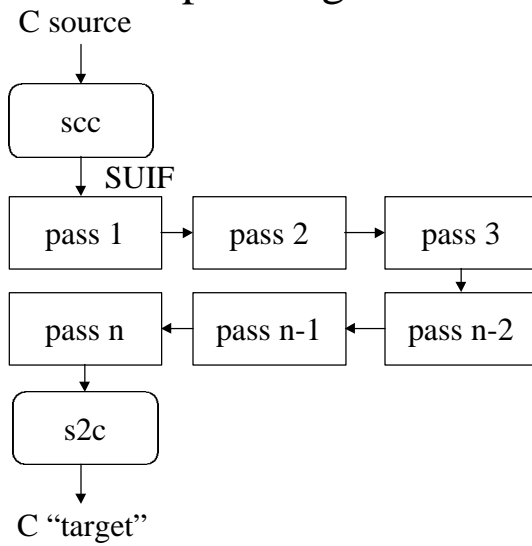
Introduction to SUIF

Amer Diwan

What is SUIF

- Stanford University Intermediate Format
- A free infrastructure for compiler research
- Supports C and FORTRAN

Compiler organization



What is a pass?

- A pass takes $k \geq 1$ pairs $\langle \text{infile}, \text{outfile} \rangle$
- infile and outfile are all SUIF files
 - Allows passes to be reordered easily--good for experimentation

How do passes communicate?

- By transforming the SUIF file
- By adding annotations
 - Annotations can be added to almost anything in SUIF
 - Annotations are like comments
 - Don't change the semantics of annotated object
 - Relay some information about the annotated object
 - A very simple but powerful mechanism

What does the SUIF representation look like?

An example: counting tree nodes

- Problem:
 - Count top level tree nodes in each procedure and put that as an annotation on the procedure

The main routine

```
main (int argc, char **argv) {
    start_suif (argc, argv);
    assert_msg(argc == 3, ("example1 infile
outfile"));
    fileset->add_file(argv[1], argv[2]);
    // Register the annotation
    ANNOTE(k_count_annotate, "tree_count", TRUE);
    visit_procs avisitor;
    avisitor.process_fileset(fileset);
    exit_suif();
}
```

The workhorse

```
char *k_count_annotate;

class visit_procs: public suif1_visitor {
public:
    void process_procedure(proc_sym *p) {
        unsigned int n_tree_nodes = 0;
        tree_node_list *tnl = p->block()->body();
        tree_node_list_iter tnl_iter(tnl);
        while (!tnl_iter.is_empty()) {
            tnl_iter.step();
            n_tree_nodes++;
        }
        immed_list *ann_data = new immed_list;
        ann_data->push(n_tree_nodes);
        p->append_annotate(k_count_annotate, ann_data);
    }
};
```

What next

- Read additional documentation on SUIF
- Try out my example
- Write a simple pass that identifies basic blocks