

GENERAL PURPOSE SWITCHABLE GATE TRANSFER VECTORS  
by David P. Reed

Introduction

I have generated a set of commands which are useful in dealing with the problem of substituting for certain supervisory interfaces. With these commands, you may direct a selected subset of calls to a supervisor gate, such as `hcs_`, to another program, without recompiling any programs.

This task is accomplished by inserting a ring 4 transfer vector in the path of all calls to the gate. In the normal case, this transfer vector will transfer directly to the gate itself, but selected entries in the transfer vector may be made to transfer to user programs instead. The transfer vector is very efficient, and introduces no glitches into the user's process.

Usage

In order to use the subsystem, one must first prepare a gate transfer vector. There is already such a gate transfer vector for the gate `hcs_` in my search directory, `>udd>CompSys>Reed>e`. If you wish to substitute for any other gates, see the following section on preparing a gate transfer vector.

First, one replaces the gate with the gate transfer vector. This is done with the `switch_gate` command. Its usage is as follows:

```
switch_gate <gate_name> <gate_transfer_vector_path>
```

This command makes the gate transfer vector specified by the second argument have the reference name `<gate_name>`, and gives the original gate a reference name composed by concatenating "real\_" and `<gate_name>`. Thus if programs need to call the supervisor gate itself, they can do so by referring to `real_<gate_name>$entrypoint`.

Now one is able to redirect selected entrypoints of the gate transfer vector to call interface procedures directly. This is done with the `switch_gate_entry` command. Its usage is as follows:

---

This note is an informal working paper of the Project MAC Computer Systems Research Division. It should not be reproduced without the author's permission, and it should not be referenced in other publications.

```
switch_gate_entry <gate_entry> <target_entry>
```

This command takes a gate entrypoint, such as hcs\_\$tty\_index, and causes all calls to the program to go to the specified target entrypoint, which might be my\_tty\$index.

Remember, my\_tty\$index may wish to refer to the real hcs\_\$tty\_index, in which case it should use the name real\_hcs\_\$tty\_index.

One can reset particular entrypoints so that they do go directly to the real gate, by using the reset\_gate\_entry command:

```
reset_gate_entry <gate_entry>
```

Finally, one can remove the transfer vector from the path of calls to the gate by simply terminating it with the tmr command.

An example of its use to replace the two validation level programs by user-written interfaces follows:

```
switch_gate hcs_ >udd>CompSys>Reed>e>hcs_
switch_gate_entry hcs_$level_set my_level_set
switch_gate_entry hcs_$level_get my_level_get
... running with new level get and set ...
reset_gate_entry hcs_$level_set
reset_gate_entry hcs_$level_get
```

CAUTION: Since the switched gate entrypoints are used by the command loop and system commands, it is possible to get into severe problems if the programs substituted for gate entrypoints don't work. Consequently, be sure that programs do work.

### Creating a Gate Transfer Vector

A gate transfer vector is prepared from the standard system gate source by assembling it with a particular set of gate macros. Thus, one gls's the source for the system gate, such as phcs\_.mexp, and then executes the mexp command to macro expand it with the file gate\_macros.incl.mexp in my subdirectory >udd>CompSys>Reed>e. Then one can assemble it with the alm assembler. Such a gate transfer vector is pure and sharable among many people. Switching a particular gate entry only affects the process in which the gate entry is switched.

One might thus create a gate transfer vector for the system gate phcs\_ by the following steps:

```
link >udd>CompSys>Reed>e>gate_macros.incl.mexp
gls -sys hard phcs_.mexp
mexp phcs_
alm phcs_
```

## Notes on Support

All commands and subroutines having to do with switchable gates are found in my subdirectory >udd>CompSys>Reed>e. I will keep them up to date with the current system, so that there is no need to copy your own version. Suggestions for improvements will be welcome.

