Static and Dynamic Relocation

Introduction

This shows the basic hardware instruction cycle for a machine that uses static relocation and for one that uses dynamic relocation.

Static Relocation

Static relocation refers to address transformations being done before execution of a program begins. A typical hardware instruction cycle looks like this:

```
while true do begin
    w := M[instr_ctr];          (* fetch instruction *)
    oc := Opcode(w);
    adr := Address(w);
    instr_ctr := instr_ctr + 1;
    case oc of
    1:  reg := reg + M[adr];    (* add *)
    2:  M[adr] := reg;          (* store *)
    3:  instr_ctr := adr;       (* branch *)
    ...
    end
end (* loop *)
```

Dynamic Relocation

Dynamic relocation refers to address transformations being done during execution of a program. In what follows, the function *NLmap* (for Name Location map) maps the relocatable (virtual) address va given in the program into the real (physical) storage address pa:

```
pa := NLmap(va)
```

So, a typical hardware instruction cycle looks like this:

```
while true do begin
    w := M[NLmap(instr_ctr)];       (* fetch instruction *)
    oc := Opcode(w);
    adr := Address(w);
    instr_ctr := instr_ctr + 1;
    case oc of
    1:  reg := reg + M[NL_map(adr)];  (* add *)
    2:  M[NL_map(adr)] := reg;        (* store *)
    3:  instr_ctr := NL_map(adr);     (* branch *)
    ...
    end
end (* loop *)
```