$ + **GNU Chess Move Generator**

New move Generation algoritm:

Revision: 1989-09-06

Author: Hans Eric Sandstroem.

This algortim is the result of an attempt to make an hardware move generator, but since I newer had the time and resources to build the hardware I wrote a software version and incorporated that one into gnuchess. This was the best way I could think of sharing this algorithm with the computer chess community.

If there is anybody out there with the time and rescources to build a hardware move generator I will be glad to assist.

The general idea behind this algoritm is to pre calculate a lot of data. The data that is pre calculated is every possible move for every piece from every square disregarding any other pieces on the board. This pre calculated data is stored in an array that looks like this:

```
struct sqdata {
  short nextpos;
  short nextdir;
};
struct sqdata posdata[8][64][64];
/* posdata[piecetype][fromsquare][destinationsquare] */
example:
```

---

move_gen
$ GNU Chess Move Generator
+ index:0030

To handle the differeces between white and black pawns (they move in opposite directions) an array ptype has been introduced:

```
static const short ptype[2][8] = {
  no_piece,pawn,knight,bishop,rook,queen,king,no_piece,
  no_piece,bpawn,knight,bishop,rook,queen,king,no_piece};
         ^^^^^
```

And it is used like this:

```
   piecetype = ptype[side][piece]
```

When generating moves for pieces that are not black pawns, piece can be used directly in posdata. As in the example above.

Thus the only thing one has to do when generating the moves is to check for collisions with other pieces. the move generation to do this looks like this: (for non pawns)

```
    p = posdata[piece][sq];
    u = p[sq].nextpos;
    do {
      if (color[u] == neutral) {


      }
      else {


      }
    } while (u != sq);
```

- I`nt this just beautiful!

The array posdata is initialized in the routine Initialize_moves. This routine is called just once and it works so no time has been spent on the structure of this code. GenMoves and CaptureList generates the moves but the routines ataks, BRscan, Sqatakd, KingScan and trapped also relies on the move generation algoritm so they have also been rewritten.