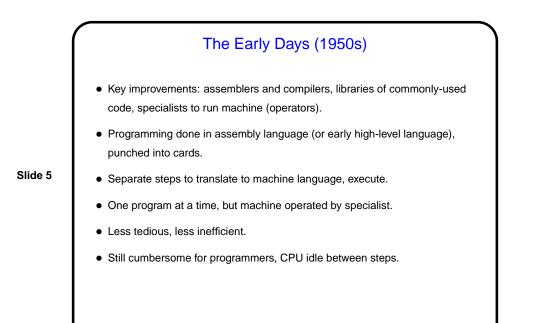
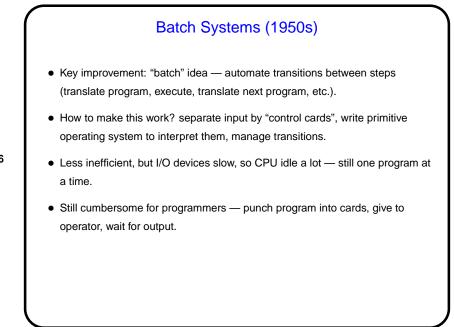


The Early Days (1940s – 1950s)
Key improvements: stored-program concept, punch cards.
Programming done by encoding machine language into cards.
Program included code to start up computer, read rest of program into memory, do all input and output, etc. (no operating system).
One program at a time, machine operated by programmer.
Better, but still tedious and inefficient!





```
Control Cards — Example
         //jobname JOB acctno,name, ....
         //stepname EXEC PGM=compiler_name,PARM=(options)
         //STEPLIB DD
                          DSNAME=path_for_compiler
         //SYSUT1
                    DD
                          UNIT=SYSDA, SPACE=(subparms)
Slide 7
         //SYSPRINT DD
                          SYSOUT=A
         //SYSLIN
                    DD
                          DSNAME=object_code,UNIT=SYSDA,
         11
                          DISP=(MOD, PASS), SPACE=(subparms)
         //SYSIN
                    DD
                          *
         source code
         /*
         //stepname EXEC PGM=load-and-go
         . . . .
         .... input data for program ....
```

Multiprogramming Systems (1960s – ?)
Key improvement: "multiprogramming" — more than one program in memory, so when one has to wait another can run.
How to make this work? requires much more complex operating system — must share memory and I/O devices among programs, switch between them, etc.
Efficient use of hardware.
Still cumbersome for programmers — no real changes here.
Example: IBM mainframe (picture here, 1964); keypunch machine (picture here); line printer (picture here).

