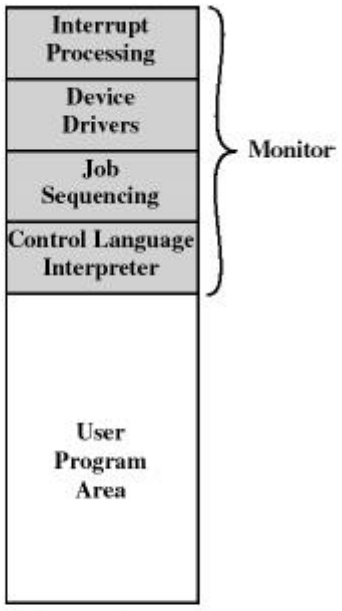
Simple Batch Systems

- Are the first operating systems (mid-50s)
- The user submit a job (written on card or tape) to a computer operator
- The computer operator place a batch of several jobs on a input device
- A special program, the monitor, manages the execution of each program in the batch
- Resident monitor is in main memory and available for execution
- Monitor utilities are loaded when needed

The Monitor

- Monitor reads jobs one at a time from the input device
- Monitor places a job in the user program area
- A monitor instruction branches to the start of the user program
- Execution of user pgm continues until:
  - end-of-pgm occurs
  - error occurs
- Causes the CPU to fetch its next instruction from Monitor



Memory Layout of Resident Monitor

Job Control Language (JCL)

- Is the language to provide instructions to the monitor
  - what compiler to use
  - what data to use
- Example of job format: ------->>
- `$FTN` loads the compiler and transfers control to it
- `$LOAD` loads the object code (in place of compiler)
- `$RUN` transfers control to user program

```
$JOB
$FTN
. . .
FORTRAN
program
. . .
$LOAD
$RUN
. . .
Data
. . .
$END
```


Job Control Language (JCL)

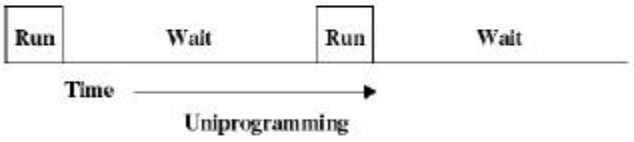
- Each read instruction (in user pgm) causes one line of input to be read
- Causes (OS) input routine to be invoke
  - checks for not reading a JCL line
  - skip to the next JCL line at completion of user program

Batch OS

- Alternates execution between user program and the monitor program
- Relies on available hardware to effectively alternate execution from various parts of memory

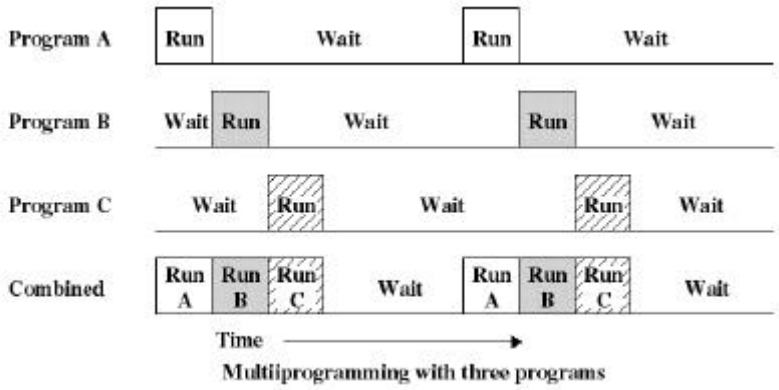
Multiprogrammed Batch Systems

- I/O operations are exceedingly slow (compared to instruction execution)
- A program containing even a very small number of I/O ops, will spend most of its time waiting for them
- Hence: poor CPU usage when only one program is present in memory



Multiprogrammed Batch Systems

- If memory can hold several programs, then CPU can switch to another one whenever a program is awaiting for an I/O to complete
- This is multitasking (multiprogramming)



Multiiprogramming with three programs

Desirable Hardware Features

- Memory protection
  - do not allow the memory area containing the monitor to be altered by user programs
- Timer
  - prevents a job from monopolizing the system
  - an interrupt occurs when time expires

Desirable Hardware Features

- **Privileged instructions**
  - can be executed only by the monitor
  - an interrupt occurs if a program tries these instructions
- **Interrupts**
  - provides flexibility for relinquishing control to and regaining control from user programs

Requirements for Multiprogramming

- **Hardware support:**
  - I/O interrupts and (possibly) DMA
    - in order to execute instructions while I/O device is busy
  - Memory management
    - several ready-to-run jobs must be kept in memory
  - Memory protection (data and programs)
- **Software support from the OS:**
  - Scheduling (which program is to be run next)
  - To manage resource contention

Time Sharing Systems (TSS)

- Batch multiprogramming does not support interaction with users
- TSS extends multiprogramming to handle multiple interactive jobs
- Processor's time is shared among multiple users
- Multiple users simultaneously access the system through terminals

Time Sharing Systems (TSS)

- Because of slow human reaction time, a typical user needs 2 sec of processing time per minute
- Then (about) 30 users should be able to share the same system without noticeable delay in the computer reaction time
- The file system must be protected (multiple users…)