

- 4
- a) - statically defined schedule
- "no" computational overhead at runtime - always knows what to run next.
 - probably over-provide CPU resource to create schedule
 - use rate monotonic + define periods as multiple of 2 * smallest period.
- 8
- b) - processes executing OS of highest priority
- * priority depends on what they are doing or were waiting for.
 - user processes may start off at same priority or may have a possible range of user-allocated priorities
 - multiple priorities \rightarrow multiple queues.
 - timeslice and re-queue according to CPU time used
- 8
- examples - UNIX + NT may be given.
- c) - mixture of periodic + best effort applications.
- processes arrive dynamically \rightarrow admit?
 - deadlines are soft + QoS may be reduced to some minimally acceptable value (audio/video)
 - user knows relative importance of BE + MM apps.
 - priority schemes not sufficient - eg. video may be of least importance to user
 - EDF with deadline = end of period for MM
- 8