Real-Time Systems Lecture 3
These were some of the more important things written out during the class. They are meant to be read in conjunction with the PowerPoint slides and the textbook.

Considered so far: $d_i = p_i$

relative deadline

What if $d_i \leq p_i + i$?

Condition becomes $W_i(t) \leq t$

for at least some $t \leq d_i$

RMA is an optimal static priority algorithm (under our assumptions)

if any static priority algorithm can feasibly schedule a task set, then RMA will do so as well.
Assume $A^*$ hypothetical algorithm which schedules task set $J$.

But that RMA cannot schedule $J$.

$T_i$ sched. under $A^*$

$\Rightarrow W_i(t) \leq t$ for some $t \in [0, P_i]$.

Proof approach:

There must exist $T_a, T_b \in J$ such that

$T_a \not\preceq_{A^*} T_b$, but

$T_a \preceq_{RM} T_b$.

Then show that if $J$ was schedulable under the $A^*$ priority assignment, then $J$ will also be sched. if $T_a < T_b$ is used.
By a sequence of such priority swaps, we'll end up with the same priority assignment as RM. \[ \Rightarrow \] RMA can schedule J, a contradiction.

Lemma 3.8 - EDF

Proof by contradiction: Assume the lemma is false and that the processor is idle for part of this interval.

\[ \text{t}_\text{last} \]
\[ \text{t}_\text{miss} \]
The idea is to show that a deadline would have been missed earlier than $t_{\text{miss}}$, thus contradicting our assumption that $t_{\text{miss}}$ was the earliest such instant. Since there is only a finite number of possible instants at which a deadline can be missed, this quickly leads to the conclusion that no deadline could have been missed if the processor had not been continuously busy over $[0, t_{\text{miss}}]$. 