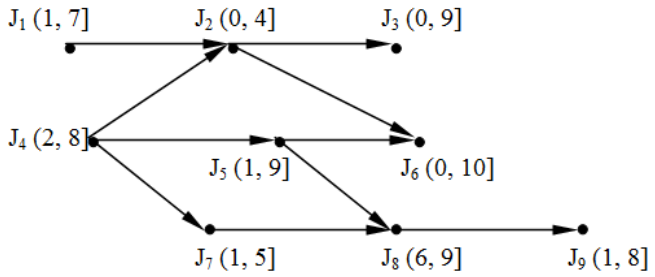
CIS 721 - Real-Time Systems
Homework #1
Spring 2015

Due: Friday, Sept. 11, 2015, at 11:59 pm – upload via K-State OnLine.

1. (5 points) The feasible interval of each job in the precedence graph (shown below) is given next to its name. The execution time of all jobs is equal to 1.

   a. Find the effective release times and effective deadlines of the jobs in the precedence graph. Recall that run time is not used in computing effective release times and effective deadlines.
   b. Find a feasible schedule of the jobs on a single processor using their effective release times and effective deadlines. To find a feasible schedule, use the fact that each job runs for 1 time unit and adjust the effective times to compute possible release times and deadlines; e.g., the effective release time of job J_2 is 2, but the earliest possible release time of job J_2 is 3 because job J_4 will run for 1 unit of time; if it is released at time 2, it won't finish until time 3.



2. (10 pts.) Consider the task set consisting of the following three preemptive, periodic tasks (denoted using the notation $\tau_i = (p_i, e_i, D_i)$):

   $\tau_1 = (8, 2, 8)$
   $\tau_2 = (16, 3, 16)$
   $\tau_3 = (16, 5, 8)$

   The system is to be scheduled and executed using a fixed cyclic schedule.

   a) Using a frame size of 8, define a network flow graph that can be used to find a fixed (static) cyclic schedule of the tasks using a maximum network flow algorithm. Hint: Remember to indicate maximum allowable flows on all edges in the graph used as input. Upload as hwk1.inp.
   b) Use hwk1.inp as input the hi_pr to determine the maximum network flow on each edge using the command: hi_pr < hwk1.inp > hwk1.out. Upload the output as hwk1.out.
   c) Convert the output to graphics format using the custom convert code: convert 4 2 hw1.out hwk1.gr and then to graphics using the GraphViz command: dot -Tpng -o hwk1.png hwk1.gr, or the editor Gvedit available online at http://www.graphviz.org.

3. (10 points) Consider the synchronous task set shown in Table 1 (on the next page). This task set comes from an embedded signal processing application for an anti-submarine warfare (ASW) system. More specifically, the task set is a modified subset of the tasks that implement the Directed Low Frequency Analysis and Recording (DIFAR) acoustic signal processing application from the Airborne Low Frequency Sonar (ALFS) system of the U.S. Navy's SH-60B LAMPS MK III anti-submarine helicopter. The ALFS system processes low frequency signals received by sono-buoys in the water. Its primary function is to detect and track submarines and to calculate range and bearing estimates to each target. The task set shown in Table 1 represents an implementation of a portion of the DIFAR application on a Mercury PowerPC 6U VME board with a 200MHz 603e processor. This portion of the DIFAR application processes five bands of Constant Resolution (CR) data from each of five sono-buoys. In this assignment, we will assume all tasks are independent.

Task ID	Phase in ms	Period in ms	Time/Exec in ms	Relative Deadline	Processing Primitive
1	0	250	6.4545	250	FLW
2	0	250	30.1303	250	BDFC
3	0	250	0.3437	250	MASTERMCS
4	0	250	0.1022	250	SLAVEMCS
5	0	250	5.7349	250	DIFARDAD
6	0	250	5.7557	250	DIFARDAD
7	0	250	5.7974	250	DIFARDAD
8	0	250	5.8807	250	DIFARDAD
9	0	250	6.0472	250	DIFARDAD
10	0	250	4.3071	250	DIFARDAD
11	0	250	7.7672	250	DIFARDAD
12	0	250	14.6875	250	DIFARDAD
13	0	250	7.183	250	CRFIL
14	0	250	7.3999	250	CRFIL
15	0	250	7.8337	250	CRFIL
16	0	250	8.7012	250	CRFIL
17	0	250	8.7012	250	CRFIL
18	0	250	8.1264	250	CRSPECANAL
19	0	250	8.1264	250	CRSPECANAL
20	0	250	8.4815	250	CRSPECANAL
21	0	250	9.1918	250	CRSPECANAL
22	0	250	9.1918	250	CRSPECANAL
23	0	250	3.217	250	ALLBANDMERGE
24	0	250	3.5179	250	SAD
25	0	250	3.6363	250	GRM
26	0	250	5.1914	250	BBC
27	0	250	0.1496	250	GRAMMERGE
28	0	500	3.3671	500	CRDETECT
29	0	500	3.3671	500	CRDETECT
30	0	500	3.3671	500	CRDETECT
31	0	500	3.3671	500	CRDETECT
32	0	500	3.3671	500	CRDETECT
33	0	2000	3.1913	2000	ALI
34	0	2000	5.1122	2000	BRG
35	0	2000	0.5047	2000	ALIMERGE
36	0	2000	0.5906	2000	BEARMERGE
37	0	6000	2.4799	6000	AUTODETECT
38	0	6000	0.199	6000	BINMERGE
39	0	6000	0.6898	6000	AUTODETMERGE
40	0	6000	40.823	6000	EXTRAMERGE

Table1: Synchronous signal processing task set for a 200MHz 603e PowerPC.

a. For all i, $1 \leq i \leq 40$, what is the utilization of task T_i? What is the total system utilization?
b. What is the hyperperiod (H) of the task set?
c. What are the possible frame sizes that could be used to create a static cyclic schedule for this task set?
d. Create a static cyclic schedule for the task set (you may want to create a program to do this using a Maximum Network Flow algorithm). Turn in the schedule created. If you wrote a program to generate the schedule or the input to a maximum network flow algorithm, please turn it in as well. You can retrieve a text file containing the task parameters in 4-tuple form online as HW1table1.txt.
e. What is the frame size f of your schedule? How many minor cycles are there in each major cycle?

Hints: Andrew Goldberg's routines to compute maximum network flow (HIPR) are available at: http://avglab.com/andrew/soft/hipr.tar.