CS 444 Homework 1
Posted Thursday, February 6, 2025
Due 11:59 pm Thursday, February 13, 2025

Homework must be typeset, converted to PDF and placed in your course directory on the CS Linux servers. If you have not logged into the server or have not registered for cs444 in the cs portal to get a course directory, email operator@cs.umb.edu for immediate help.

This homework is to be named **hw1.pdf** – the filename must be exactly **hw1.pdf**, otherwise it will not be collected by our script. Common mistakes for filenames include Hw1.pdf, HW1.pdf, hw1.hw1.pdf.

If you want to submit answers to questions separately, use names hw1-q1.pdf, hw1-q2.pdf and hw1-q3.pdf and the script will collect it. If you run short on time and request an extension, you must have some work done and on the server for the extension to be granted without a late penalty.

Upload the file to your home directory on the CS Linux server and use the linux command for copy to move it to your course directory folder named cs444 ("cp hw1.pdf /cs444/". If you have trouble with uploading, email operator@cs.umb.edu. Not being able to upload to the server is not a valid excuse for an extension because this is a course on operating systems! In your course directory, use the list command ("ls -l") to check that hw1.pdf is there, has the appropriate number of bytes and is readable by the grader group – the second group of permissions must contain 'r'.

For homework1, be sure you have at least one .c file in your course directory, edited and compiled on the server, with an executable that runs. This has been done in class.

1. Huffman Coding. The following table lists the character frequencies in a file.

char	0	1	2	3	4	5	6	7	8	9
freq	37	31	29	19	17	13	11	7	3	2

(a) Apply Huffman's algorithm to the data, and draw the prefix code tree. The leaves should be annotated with the characters and their frequencies, and internal nodes should be annotated with the total frequencies of their subtrees. When merging two subtrees, put the lower-frequency subtree on the **left**. Please make a series of drawings

by hand that show all your work from forest to complete tree. Please do not use a graphics program.

(b) Fill out the following table with the Huffman codes for each character. On the prefix code tree, a left branch is 0, and a right branch is 1. Assemble the code for each leaf node character by reading the branch numbers from top to leaf node.

char	0	1	2	3	4	5	6	7	8	9

You can draw the Huffman tree by hand, take a picture and include the image in the PDF file of your submission.

2. Chapter 1 metric prefixes and ASCII codes for project1.

(a) How many nanoseconds in one second? Write out all the digits.

(b) At how many bits per second is a 1 giga bit per second channel running? Write out all the digits.

(c) Kilo as a metric prefix means 1000 or 10^3 except if it is measuring memory in which case it means 1024 or 2^10. Suppose you have a sketch of one kilobyte of memory drawn as a rectangle that is 8 bytes wide. How high is the rectangle in bytes?

(d) Computer screens show letters and numbers but of course must use bits to store them. The ASCII code for a letter or a number or a non-printing character takes 8 bits or one byte. Look at:
https://www.eecis.udel.edu/~amer/CISC651/ASCII-Conversion-Chart.pdf . For each of three characters, ‘A’, ‘9’ and ‘?’, write the codes in decimal, binary and hexadecimal.

3. Scheduling Processes. Consider the following arrivals of processes in an interactive system.

(a) Draw three timelines that illustrate the execution of processes P1 - P6 described in the table below using these scheduling algorithms:

i. Shortest job first, SJF
ii. Non-preemptive priority, NPP – a smaller number has higher priority
iii. Round-robin, RR, with time slice 2:

Process	Arrival time	CPU time	Priority
P1	0	3	2
P2	0	3	1
P3	3	4	2
P4	4	2	3
P5	8	3	1
P6	8	1	2

If there is a tie (same CPU time for SJF or same priority for NPP), use first-in-first-out (FIFO) to break the tie. For RR scheduling, if P1 is removed from the CPU and put in the waiting queue just when P3 arrives, P3 will be ahead of P1 in the queue.

(b) Describe the priority inversion problem discussed in Chapter 2.