**INTRODUCTION TO COMPUTER SCIENCE**

**HOMEWORK 5**

# Spring 2018

**Assigned**: Wednesday, April 11 **Due:** Wednesday, April 25 by 4pm



**ANSWER ALL QUESTIONS ON OTHER SHEETS OF PAPER. DO NOT SUBMIT ANY ANSWERS ON THESE SHEETS.**

**Part I. (8 points) General Operating Systems**

Provide a short answer for each of the following:

* 1. What is address binding, and when does it occur?

Address binding is the mapping (translation) of a logical address into a physical address. It occurs when a process’s instructions are placed into main memory.

* 1. Describe how a process might move through the various process states. Note specific reasons why a process moves form one state to another.

A new process begins in the **new** state. When the process has no bars to its execution, it moves into the **ready** state. It waits in the **ready** state until it gets time in the **running** state. It runs for a while and might issue a command for file or user input. The process is moved into the **waiting** state until the I/O has been completed, at which time it moves into the **ready** state to await another turn in the **running** state. Eventually it finishes, and moves into the **terminated** state.

* 1. What is demand paging?

Demand paging is the technique where pages are brought into memory only when they are referenced (needed). Demand paging allows programs of any size, thus giving the illusion of infinite memory.

* 1. What is the relationship between multiprogramming and timesharing?

Multiprogramming allows multiple processes to be active at once. Timesharing allows the multiple processes to be interactive ones.

**Part II. (18 points): Memory Management**

Assume the following arrangement of free blocks. The shaded blocks are used by other programs. The white blocks are available memory. Note, the blocks may not appear proportional to the amount of space they represent.

OS

0 20 32 58 66 96 118 156 180

For each memory management approach, draw (or copy/paste) a separate diagram showing the exact locations of the inserted processes and the remaining partitions (including starting and ending memory indices). Assume that **dynamic** partitions are being used.

Show the state of memory after the following processes are added (in the following order)

 **Process A (4 MB)**

 **Process B (10 MB)**

 **Process C (11 MB)**

using the following memory management approaches:

1. First Fit

A

B

OS

0 20 24 32 58 66 96 106 117118 156 180

C

1. Best Fit

B

C

OS

0 20 30 32 58 62 66 96 107 118 156 180

A

1. Worst Fit

B

OS

A

0 20 32 58 66 96 106 118 156 160 171 180

C

**Part III: (15 points): Processor Management**

For the set of processes and service times below, **draw** a separate Gantt Chart and **calculate** the average turnaround time for:

Process Service Time

 p1 45

 p2 20

 p3 70

 p4 35

1. First-come-first-serve CPU scheduling.

|  |  |  |  |
| --- | --- | --- | --- |
| P1 | P2 | P3 | P4 |

0 45 65 135 170

Turnaround Time = (45 + 65 + 135 + 170)/4 = 103.75

1. Round-robin CPU scheduling. For the round robin scheduling, use a time slice of 10 time units.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P1 | P2 | P3 | P4 | P1 | P3 | P4 | P1 | P3 | P4 | P1 | P3 |

0 10 20 30 40 50 **60** 70 80 90 100 110 120 130 **135 140 170**

Turnaround Time = (60 + 135 + 140 + 170)/4 = 126.25

c) Shortest job first

|  |  |  |  |
| --- | --- | --- | --- |
| P2 | P4 | P1 | P3 |

0 20 55 100 170

 Turnaround Time = (20 + 55 + 100 + 170)/4 = 86.25

**Part IV: (9 points) Artificial Intelligence**

1. For each of the following, identify whether a human or a computer can complete the task more easily: (1 pt each)
	1. Add up 42 ten-digit numbers. computer
	2. Paint a landscape. human
	3. Interpret a poem. human
	4. Produce an index of all of the words in a book. computer
2. (5 pts) If an artificial neuron accepted four input signals with values of 0, 1, 0, 1 and weights of 4, -3, 6, and ‑5, what is the output if the threshold is 5? What is the output if the input signals are 1, 0, 1, 0?

First output:

 0\*4 + 1\*-3 + 0\*6 + 1\*-5 = 0 + -3 + 0 + -5 = -8 < threshold, so output is 0.

Second output:

 1\*4 + 0\*-3 + 1\*6 + 0\*-5 = 4 + 0 + 6 + 0 = 10 > threshold, so output is 1.

**Part V: (50 points) Python Programming – Animations**

Your Python code must follow these guidelines (Be aware that if you don’t follow these guidelines, you will lose points, **even if your program runs perfectly**):

* *Always put your name, date, program name, and a general comment describing what your program does at the top of your code. This should be a general statement about how your program behaves.*
* *variable names should help describe the purpose of a variable.*
* *Function names should help describe the purpose of a function*
* *Every function should be preceded by a comment describing the function’s purpose, input, and output (if any).*

Your assignment is to create an animation in Python, with the following specifics:

* The animation must be at least 5 and at most 10 seconds long
* The animation’s frame rate must be 30 (so you’ll need to create at least 150 frames)
* The canvas for your animation should be 300-500 pixels in both height and width
* Your program should use exactly one “for” loop
* There must be at least 3 things in motion in your animation; they must all move differently in a substantial way.
	1. At least one “thing” must be a filled drawing object (i.e., use addArcFilled or addRectFilled or addOvalFilled)
	2. At least one “thing” must be an image.
	3. The other “thing” can be anything – other drawing objects, small pictures, whatever but it must be different than object 1 & 2 above. If using a filled object, it must be a different type of object and a different color than used for item 1.
* At least one of the objects must change its direction partway through the animation.
* You must include some text in the animation that changes in size and/or position multiple times as the animation progresses

**NOTE**: A perfectly working program as described above will earn 45 points. To earn the remaining 5 points, you must use your own creativity to go above and beyond those minimal requirements. Examples include making one of the objects have some aspect of its motion determined at random, wider variety of motions and acceleration, adding a background, etc.

**Tips**:

1. Inside hw/hw5, create an empty folder called “frames”. Don’t put anything in there (your program will put frames in there, though).
2. Build on what you’ve learned! If you don’t have your book beside you and your code from the labs and previous homeworks in front of you when you’re doing this, then you are going to waste a lot of time.
3. Use the “hw5.py” file as a starting point; it’s on the website!!!

**TURNING IT IN:**

1. Put a copy of your hw5.py, any other files your program needs, and your frames folder (it can be empty if you like) in your csis110/hw/hw5 folder. THIS IS REQUIRED IN ORDER FOR ME TO GRADE IT AND FOR YOU TO EARN FULL CREDIT.
2. Turn in a printout of your code (the hw5.py file) with the rest of your homework.