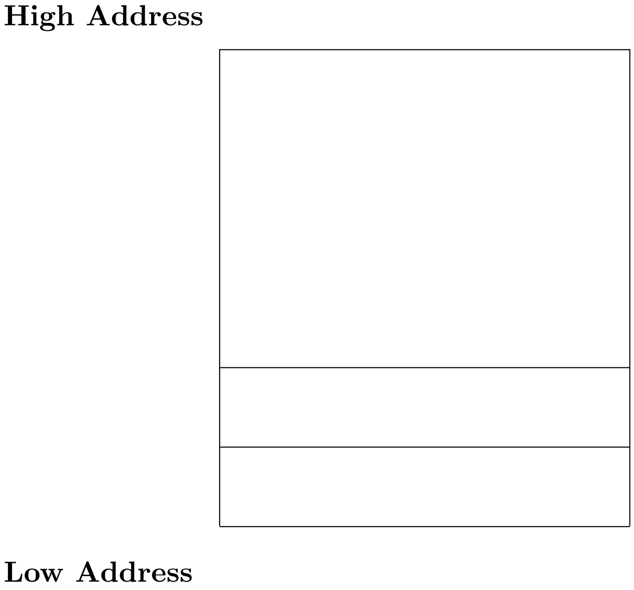
CSSE 332 -- OPERATING SYSTEMS

The Process Abstraction

Name: ________________________________

Question 1. (5 points) The figure below represents the address space of a process `P`. Label each part of the process address space with its corresponding section content (i.e., stack, heap, etc.).



Question 2. (5 points) Describe how processes are related to each other in a Unix-like operating system.

Question 3. (5 points) How does a process keep track of who its direct parent is?

Question 4. (5 points) In the standard C library, the ______________ system call is used to create a new process by ______________ the calling process. The new process is called a ______________ of the calling process. Finally, a process can use the ______________ system call to obtain its process id.

Question 5. (a) (5 points) Where can you find the documentation for the `fork` system call? What is the command you can use to bring it up?

(b) (5 points) From the documentation page, which header file should you include to use `fork`?

Question 6. (5 points) Consider the code snippet below.

```c
pid_t pid = fork();
if(pid == 0) {
  printf("Hello from the child process %d\n", getpid());
  exit(0);
} else {
  printf("Hello from the parent process %d\n", getpid());
  exit(0);
}
```

Which of the print statements will show up on the console first?

Question 7. (5 points) Consider the code snippet below.

```
pid_t my_pid = getpid();
if(fork() == 0) {
  printf("My pid is %d\n", my_pid);
  exit(0);
} else {
  printf("My pid is %d\n", my_pid);
  exit(0);
}
```

Which of the following statements is **True**?

A. Each process will print its own process id.

B. Both processes will print the same value, which is the process id of the parent.

C. Both processes will print the same value, which is the process id of the child.

D. We cannot know what values will be printed in each case.

E. None of the above.

Question 8. (10 points) Consider the following snippet of code.

```
for(int i = 0; i < 3; i++)
  fork();
```

How many process will we end when this loop runs? Draw the corresponding tree of these processes.