|  |  |
| --- | --- |
| **CS 341** | **Lecture Handout #4** |

**Q1 Thinking about pointers...**

|  |  |
| --- | --- |
| **1:** | **int \*\*\*\*\*\* ptr;** |

**Q2. Using read():**

|  |
| --- |
| **ssize\_t read(int fd, void \*buf, size\_t count);** |

…what type of call is read?

…how would we use it?

|  |  |
| --- | --- |
| **1:**  **2:**  **3:** |  |

**Q3. Using scanf():**

|  |
| --- |
| **int scanf(const char \* format, ...);** |

In **scanf**, the format string is the same as **printf** except that every type must be passed by reference to be written into by **scanf**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Specifier:** | **d i** | **u o x** | **f** | **c s** | **p** |
| **Type:** |  |  |  |  |  |

**scanf** return value (and why is it useful)?

Example:

|  |  |
| --- | --- |
| **1:**  **2:**  **3:**  **4:** | **int num; char c;**  **int result = scanf("%d %c", &num, &c);**  **printf("Values: %d %c\n", num, c);**  **printf("Return value: %d\n", result);** |

…what is the return value for the input: **7 hello**

…what is the return value for the input: **6** *(…followed by an EOF)*

|  |
| --- |
| **10,23 20,25 30,37 ...** |

**Q4 fscanf, scanf, sscanf ?**

How can I read and process my data?

|  |  |
| --- | --- |
| **1:**  **2:**  **3:** | **FILE \*file = fopen("mydata.csv","r");** |

**Q5. Using getline():**

|  |
| --- |
| **ssize\_t getline(char \*\*lineptr, size\_t \*n, FILE \*stream);** |

The C-string passed by reference as **lineptr** will store the line; the size of the memory allocated in **lineptr** must be stored in **n** (to avoid overflow). Additionally:

|  |
| --- |
| If **\*lineptr** is set to **NULL** and **\*n** is set **0** before the call, then **getline()** will allocate a buffer for storing the line. This buffer should be freed by the user program even if **getline()** failed. |

*…found in* ***man getline***

Example usage:

|  |  |
| --- | --- |
| **1:**  **2: 3:**  **4:**  **5:**  **...**  **n:** | **char \*s = NULL;**  **int n = 0;**  **getline(&s, &n, stdin); getline(&s, &n, stdin);**  **...**  **free(s);** |

**Q6. Processes: What are they? Can I have new one?**

A process is the base computation container on Linux; multiple processes allow for multiple separate (and parallel) execution.

Is there a system call to make a new process?

**Q7. Environmental Variables**

A process-specific dictionary that stores information about the execution environment:

* Command line:
* C programming: