Exam Two Master Sheet To Take Over The Word

IO and Files:

Files:

F family of functions: fopen(), fclose(), fread(), fwrite(), fgets(), fscanf()

Fopen():

Returns a pointer to a file object that you can now use the other f family functions on. Takes in a path to the file in the form of a string and a char for the opening mode for example, r or w. If fopen() fails it will return NULL.

Fclose():

Returns 0 on successful completion and will take in that file object that fopen() produces.

Fread():

Returns the number of items read on completion if successful and will return a short item count or 0 on a failure. Takes in an array to store what it is reading, the length of each item to read, the, number of elements, and the file object.

Fwrite():

Returns the number of items written on completion if successful and will return a short item count or 0 on a failure. Takes in an array to write to the file, the length of each element, number of elements and the file object we are operating on.

Fgets():

Will get a line from the file. It will return the string on success and will return NULL on failure. It also reads the string into the buffer. This function takes in a string to store the line, the size of the maximum number of characters to be read, the file object we created when we opened with fopen().

Fscanf():

It will scan input and look for the format that is inputted to the function. It will return the number of items successfully matched on success and on failure it is 0. It takes in file object, the format we are looking for.

Redirection:

Dup2():

Redirects from right to left so it will change what was supposed to go to the right parameter to now go to the left parameter and then it will auto close the right parameter.

Built in file descriptors:

Stdint: 0

Stdout: 1

Stderr: 2

Remember to close all file descriptors that are not in use. If you dup2 you will at some point need to close that new file descriptor.

Forking:

fork():

Will output the PID of the child process created. Use an if statement to check if this PID is less than 0 if so then fork() failed. Else if PID = 0 you are now in the child function. And Else now you are in the parent function. Everything open and in the function before the fork will be open and accessible in both the parent and the child after the fork. However, once after the fork things opened, closed and created in the parent or child will not be shared. Also key thing is to make sure you use waitpid(PID, &exitStatus, 0) in the parent otherwise you will have zombie processes and they take points off for that. And finally remember to exit the child process or else it will run forever.

Return will end the process

Pipes:

1. Step is to create an int array of length 2
2. Then call the pipe function on that array
3. Pipe() will return -1 on failure
4. Now you can use the array as a pipe, the read end will be the 0 index will the write end will be the 1 end.

Remember pipes are only one directional so if you need two-way communication then you need two pipes. Always remember to close the ends you are not using at the beginning and then close the ones you used after you are done with them. So you could kind of treat them as file descriptors and you can also use them in dup2() allowing you to write outputs to the pipe and read from a pipe instead of stdin.

String, char, and useful Functions:

Strchr()

Strlen()

Isdigit() will check if a character is a digit or not

Isalpha() will check if a character is a letter or not

ATOI() will convert a string to an integer

Strstr()

Sizeof() will return the size of the object you pass in don’t use this for strings though use strlen() for that

Strcat()

Strcmp()

Strcpy()

Strlwr()

Strupr()