**Assignment 8 (20 points), SE 421, 10/27/2021, due: Wednesday, 11/3/2021**

**Name (Last, First): Ogbondah, Chimzim**

**Electronic Copy Requirement**: (a) The answers should be typed. (b) The first page should include the top two lines with your last and the first name. (c) Include the question for each answer. (d) The file should be named HW6-lastname-firstname.

**Prerequisite:** Index XINU in Atlas.

**Problem 1 (3 points):** Answer the following questions for the function *dswrite.*

1. List all functions that get access to the memory allocated in *dswrite*? For each of those functions identify the mechanism used to provide the access. Choose from the following mechanisms: (a) access by passing parameter through a call chain, (b) passed as a return value, (c) global variable. Note: it can be multiple mechanisms in some cases. (2 points)
   1. **Dskqopt.c, dskstrt.c mechanism a**
   2. **Dskenq mechanism a and b**
   3. **Dsinit mechanism c**
2. Where is *dvioblk* declared, and where is it initialized? (1 point)
   1. It gets declared on line 23 of ***“conf.h”*** and then it is initialized on line 13 of “dsinit.c”

**Problem 2 (2 points):** Does the pointer *packet* in the function *sndrarp* have access to the memory allocated in the *dgwrte* function? Give precise reasoning. (2 points)

1. Yes the pointer packet will have access to memory allocated in dgwrite because if you follow the call chain of udpsend(), ipsend(), .. getaddr() -> sndrarp() you can see that freebuf(packet) isn’t called which would free the allocated memory.

**Problem 3 (5 points):** The following questions are for the function *dgwrite.*

Background: The memory is allocated for a structure of type *epacket,* and the pointer *packet* points to the allocated memory. The functions *getbuf* and *freebuf* respectively allocate and deallocate memory.

1. Show the call graph of *dgwrite*. (1 point)

Diagram

Description automatically generated

1. Show the truncated call graph of *dgwrite* to retain only the call chains to either *getbuf* or *freebuf*. Note: The truncated graph is the subgraph of the call graph with *getbuf* and freebuf as leaves. (2 points)

A picture containing text, sky, map

Description automatically generated

1. Find the truncated call graph of *dgwrite* to retain the functions that reference (read or write) to the structure of type *epacket*. (2 points)

Diagram

Description automatically generated

.

**Problem 4** **(5 Points):** Compute the number of interprocedural control flow paths. Assume that **f** has 2 paths, **g** has 3 paths, and **h** has 5 paths. Mark on each node the number of paths from the node to leaves. Algebraic expressions (e.g., 22 + 32 + 3\*5 + 1) are acceptable for the number of paths.

void foo(){

g();

Diagram

Description automatically generatedif(C1){

f();

}

if(C2){

if(C3){

h();

return;

}

if(C4){

f();

}

f();

}

h();

}

**Problem 5 (5 points):** Answer the following four questions for the function *dskqopt* based on the DAG given below.

1. On how many paths, *drptr* is passed as a parameter to callee functions. (1 point)
   1. 1
2. On how many paths, *drptr* is assigned to another pointer (pointer aliasing) (1 point)
   1. 4
3. How many paths are infeasible when *dskqopt* gets called from *dswrite*? Give for one path the precise reason why the path is infeasible. (3 points)
   1. 5 and on line 21 the conditional statement will only be taken if our starting point would be dsync but since we are starting from dswrite this path is infeasible since the preconditions for the node won’t ever bet met.

Chart

Description automatically generated

X