**77714490**

**segfault in an unexpected line**

https://stackoverflow.com/questions/77714490/segfault-in-an-unexpected-line

I need to understand why this code produces a segfault at line 15

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

void IncrementStringNum(char \*str){

size\_t len = 0;

//checking if the string is valid while also counting the length

for(size\_t i = 0 ; str[i] ; i++ , len ++){

if(str[i] < '0' || str[i] > '9'){

return;

}

}

//increments the last character if it is less than '9'

if(str[len - 1] < '9'){

str[len - 1] ++; // the segfault happens here

}

}

int main(){

char \*test = "000";

for(int i = 0 ; i < 9 ; i++){

IncrementStringNum(test);

printf("%s" , test);

}

return 0;

}

**77715668**

**scanf() is crashing my program upon reading a third dereferenced integer. How can I fix this?**

https://stackoverflow.com/questions/77715668/scanf-is-crashing-my-program-upon-reading-a-third-dereferenced-integer-how-ca

**1**

Okay so here is the relevant code:

int main(int argc, char \*argv[]) {

int \*v1, \*v2, \*v3;

printf("Enter three integers: ");

scanf("%d %d %d", v1, v2, v3);

printf("We made it this far?\n");

/\* Other stuff after \*/

return 0;

}

When an input of any three integers is used, it will successfully allocate those values to \*v1, \*v2, but the program will time out when attempting to assign the third integer to v3. This also occurs when I give more than three integers. I have segmented the program and scanned for integers one by one, in this process:K \*/

int main(int argc, char \*argv[]) {

int \*v1;

int \*v2;

int \*v3;

printf("Enter three integers: ");

scanf("%d ", v1);

printf("%d\n", \*v1);

scanf("%d ", v2);

printf("%d\n", \*v2);

scanf("%d\n", v3);

printf("%d\n", \*v3);

printf("We made it this far?\n");

return 0;

}

77724594

why is my malloc(1) segfaulting my program?

https://stackoverflow.com/questions/77724594/why-is-my-malloc1-segfaulting-my-program

**-1**

I have a problem in my C program that makes it so that it segfaults on malloc(1). I've spent many hours trying to find out why it segfaults but

I just can't figure it out. This portion of my program separates strings using identifiers. Example, string="12wo4ne53wone86wo99w5ne"; identifiers, start="wo", end="ne"; would return {"wo4ne", "wone", "wo99w5ne"}.

My code is as follows.

// basically like substring() in java

char\* stringAt(char\* str, int start, int end) {

int length = end - start + 1;

int temp = start;

char\* savedChar = NULL;

savedChar = malloc(length + 1);

for(int x = 0; x < length; x++) {

savedChar[x] = str[temp];

temp++;

}

savedChar[length] = '\0';

return savedChar;

}

// finds "identifier" in string after certain position in that string

int findIdentifier(char\* str, char\* identifier, int pos, int isEnd) {

// checks first character with first character in "identifier"

for(int x = pos; x < strlen(str) - 1; x++) {

if(str[x] == identifier[0]) {

// if first characters match then check rest

for(int i = 0; i < strlen(identifier); i++) {

if(str[x + i] != identifier[i]) {

i = strlen(identifier);

}

if(i == strlen(identifier) - 1) {

// isEnd checks if you should count the last position of the identifier...

// -like ident="1234" then "end" is the position of "4".

// -not end be the position of "1"

if(isEnd == 1) {

return x + strlen(identifier) - 1;

}

return x;

}

}

}

}

return -1;

}

char\*\* separateStrings(char\* str, char\* identifier, char\* lastIdentifier) {

char\*\* savedStr = NULL;

// segfault here!

savedStr = malloc(1);

// segfualt here!

int start = -2;

int end = 0;

int x = 0;

do {

// first run check/gets first and last positions of the identifier in the string

if(start == -2) {

start = findIdentifier(str, identifier, 0, 0);

} else {

start = findIdentifier(str, identifier, end + 1, 0);

}

end = findIdentifier(str, lastIdentifier, start + 1, 1);

if(start != -1) {

if(end == -1) {

end = strlen(str);

}

char\* newStr = stringAt(str, start, end);

// dynamically increase the size of savedStr (x is iterated every run through here)

realloc(savedStr, x + 1);

savedStr[x] = malloc(strlen(newStr) + 1);

strcpy(savedStr[x], newStr);

x++;

}

} while(start != -1 && end != -1);

return savedStr;

}

I have tried looking up why this could not be working and the only thing I can think of is a Buffer Overflow. So I then tried to run findIdentifier() many times but that works fine.

The program is also inconsistent and sometimes runs farther into the code than others but is accurate with its outcomes.

**77730288**

**passing double pointers as 2d arrays in C**

<https://stackoverflow.com/questions/77730288/passing-double-pointers-as-2d-arrays-in-c>

In my code, I am passing a double pointer to an integer as a representation of a 2D array but get segfaults - this problem gets fixed (oddly enough) when I pass the array as an array (with brackets instead)

This is the main function:

int main(void) {

int matrix[3][3] = {

{ 0, 1, 0 },

{ 0, 0, 1 },

{ 0, 1, 0 }

};

printf("matrix first value outside func %d", matrix[0][0]);

CSR \*csr\_val = convert\_to\_csr(matrix, 3, 3);

output(csr\_val, stdout, 1);

}

and this is the beginning of the function convert\_to\_csr:

CSR \*convert\_to\_csr(int \*\*matrix, size\_t row\_len, size\_t col\_len) {

printf("first value of array %d", \*\*matrix);

this essentially makes a seg fault when I run it but when I use the matrix form (ie matrix[][3]) it works fine. Does this have anything to do with the fact that this matrix isn't dynamically allocated and is rather allocated on the stack?

How would I go about implementing this if I cant guarantee the size of matrix at compile time though?

**77749455**

**Segmentation fault by realloc, while creating dynamic generic array**

<https://stackoverflow.com/questions/77749455/segmentation-fault-by-realloc-while-creating-dynamic-generic-array>

**2**

I tried to implement a dynamic generic array. However, when I tested my code, the result I got was "segmentation fault". I know that this error happens by realloc in a function ArrayListResize, but why?

Here is my implementation of a few methods:

typedef struct ArrayList {

void\*\* arr;

size\_t allocated, len;

} ArrayList;

int ArrayListInit(ArrayList \*list) {

list = (ArrayList \*)malloc(sizeof(ArrayList));

if (list == NULL) {

fprintf(stderr, FAILED\_ALLOCATION\_MSG);

return -1;

}

list->allocated = 0;

list->len = 0;

list->arr = (void \*\*)NULL;

return 0;

}

int ArrayListAppend(ArrayList \*list, void \*item) {

if (item == NULL) {

fprintf(stderr, INVALID\_ARG\_MSG);

return -1;

}

if (ArrayListResize(list, list->len + 1) == -1) {

return -1;

}

list->arr[list->len] = item;

return 0;

}

int ArrayListResize(ArrayList \*list, size\_t len) {

void \*\*arr;

size\_t allocated = list->allocated, new\_allocated;

if (allocated >= len && len >= (allocated >> 1)) {

assert(list->arr != NULL || len == 0);

list->len = len;

return 0;

}

if (len == 0)

new\_allocated = 0;

else

new\_allocated = len + (len >> 3) + (len < 9 ? 3 : 6);

arr = (void\*\*)realloc(list->arr, sizeof(void \*) \* new\_allocated); // Here I get the segmentation fault

if (arr == NULL) {

fprintf(stderr, FAILED\_REALLOCATION\_MSG);

return -1;

}

list->arr = arr;

list->allocated = new\_allocated;

list->len = len;

return 0;

}

And this is the test code:

int\* GenerateIntPointer(int n) {

int\* ptr\_int = (int\*)malloc(sizeof(int));

\*ptr\_int = n;

return ptr\_int;

}

int main() {

ArrayList list;

ArrayListInit(&list);

for (size\_t i = 0; i < 10; i++) {

ArrayListAppend(&list, (void\*)GenerateIntPointer((int)i));

}

ArrayListDelete(&list, free);

return 0;

}

**77765295**

**Does anyone know why I&#39;m getting a segmentation fault after the returning value of a RPC function?**

https://stackoverflow.com/questions/77765295/does-anyone-know-why-im-getting-a-segmentation-fault-after-the-returning-value

I have a RPC program which implements two method:

1): int \* delete\_occurrences\_1(char \*\*, CLIENT \*);

2): ListDir \* subdirectory\_list\_1(char \*\*, CLIENT \*);

with ListDir defined as:

#include <linux/limits.h>

#define N 6

struct DirectoryName{

char nameD[PATH\_MAX]; //PATH\_MAX=4096

};

struct ListDir{

DirectoryName directories[N];

int cont;

};

in the client, when the function subdirectory\_list\_1 is called i get segmentation fault in the same line, even though the server side sends it correctly or at least set the structure with the correct value.

Here's the code of the client RPC (line 65 gives me seg fault):

#include "RPC\_xFile.h"

#include <rpc/rpc.h>

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#define BUFF 256

int main(int argc,char \*\* argv){

if(argc!=2){

printf("Name Program , ipAddress");

exit(EXIT\_FAILURE);

}

CLIENT \* cl;

char \* host;

int \* result;

ListDir \* out =(ListDir \*) malloc (sizeof(ListDir));

if (out == NULL) {

perror("Errore durante l'allocazione di memoria per ListDir");

exit(EXIT\_FAILURE);

}

out->cont=0;

char choiche[BUFF];

host = argv[1];

char \* path = (char\*) malloc(PATH\_MAX);

cl = clnt\_create(host, RPC\_XFILE, RPC\_XFILEVERS, "udp"); //fondamentale FATTOREX E FATTOREXVERS STESSI DEL FILE.X

if (cl == NULL) {

clnt\_pcreateerror(host);

exit(1);

}

char \* pos;

printf("Avviato servizio RPC\n");

printf("Inserire:\n");

printf("E - per eliminare le occorrenze numeriche da un file di testo specificato presente lato server\n");

printf("L - per avere la lista dei sottodirettori di primo di livello contenenti almeno 5 file di testo dato un direttorio\n");

printf("ctrl^D per terminare\n");

while(fgets(choice,sizeof(choice),stdin)!=NULL){

if ((pos=strchr(choice, '\n')) != NULL)

\*pos = '\0';

switch(choice[0]){

case 'E':

printf("Dimmi il nome del file da esaminare\n");

fgets(path,PATH\_MAX,stdin);

if ((pos=strchr(path, '\n')) != NULL)

\*pos = '\0';

result = delete\_occurences\_1(&path, cl);

if(\*result == -1){

printf("File inesistente oppure problemi di creazione/modifica file\n");

}

else{

if(\*result == 0){

printf("Nel file non sono state trovate occorrenze numeriche\n");

}

else{

printf("Nel file sono state rimosse %d occorrenze numeriche\n",\*result);

}

}

break;

case 'L':

printf("Dimmi il nome del direttorio\n");

fgets(path,PATH\_MAX,stdin);

if ((pos=strchr(path, '\n')) != NULL)

\*pos = '\0';

out = subdirectory\_list\_1(&path,cl);

printf("Ricevuto risultato");

if(out->cont == -1){

printf("Directory non trovata oppure problemi gestione file lato server\n");

}

else if(out->cont == 0){

printf("Non sono state trovate sottocartelle di primo livello che soddisfano tali specifiche\n");

}

else{

printf("Sottocartelle di primo livello trovate:\n");

for(int i=0;i<out->cont;i++){

printf("%s\n",out->directories[i].nameD);

}

}

break;

default:

printf("Servizio non disponibile, inserire E o L\n");

}

printf("Inserire:\n");

printf("E - per eliminare le occorrenze numeriche da un file di testo specificato presente lato server\n");

printf("L - per avere la lista dei sottodirettori di primo di livello contenenti almeno 5 file di testo dato un direttorio\n");

printf("ctrl^D per terminare\n");

}

free(out);

free(path);

clnt\_destroy(cl);

printf("Programma terminato\n");

exit(0);

}

for completeness, wishing that anyone could solve my problem, I paste also The server the .x file and the .h file.

Server:

#include "RPC\_xFile.h"

#include <rpc/rpc.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <dirent.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <fcntl.h>

#include <dirent.h>

#include <unistd.h>

#define DIM 256

int \* delete\_occurences\_1\_svc(char \*\* nameFile, struct svc\_req \* rq){

static int result=0;

int fd=open(\*nomeFile,O\_RDONLY);

if(fd<0){

printf("Errore apertura file %s\n",\*nomeFile);

result=-1;

return &result;

}

char tempFile[DIM];

int cont=0;

for(int i=0;i<strlen(\*nameFile);i++){

if((\*(\*nameFile)+i)=='.')

break;

else

cont++;

}

if(cont==(strlen(\*nameFile))){

strcpy(tempFile,"temp.txt");

}

else{

memcpy(tempFile,\*nameFile,cont);

strcat(tempFile,"\_temp.txt");

}

printf("Creo file temporaneo %s\n",tempFile);

int tempFd=creat(tempFile,O\_CREAT | O\_TRUNC | S\_IRUSR | S\_IWUSR);

if(tempFd<0){

printf("Errore creazione file temporaneo %s\n",tempFile);

close(fd);

result=-1;

return &result;

}

char temp;

while(read(fd,&temp,1)>0){

if(temp<'0' || temp>'9'){

write(tempFd,&temp,1);

}

else{

result++;

}

}

close(fd);

if(rename(tempFile,\*nameFile)<0){

printf("Modifica avvenuta con successo, ma errore avvenuto nella modifica del nome file temporaneo\n");

close(tempFd);

result=-1;

return &result;

}

printf("Modifica avvenuta correttamente, eliminati %d numeri\n",result);

return &result;

}

int EndsWith(char \*str, char \*suffix)

{

if(strlen(str) >= strlen(suffix))

{

if(!strcmp(str + strlen(str) - strlen(suffix), suffix))

{

return 1;

}

}

return 0;

}

ListDir \* subdirectory\_list\_1\_svc(char \*\* nameDir, struct svc\_req \* rq){

static ListDir out;

out.cont=0;

char pwd[PATH\_MAX];

char finallyDir[PATH\_MAX];

if(strlen(\*nameDir)<=0){

printf("Errore input nome direttorio\n");

out.cont=-1;

return &out;

}

if((\*nameDir[0])=='/'){

strcpy(finallyDir,\*nomeDir);

}

else{

if(getcwd(pwd,sizeof(pwd))==0){

printf("Errore getcwd\n"); //restituisce pwd

out.cont=-1;

return &out;

}

snprintf(finallyDir,PATH\_MAX,"%s/%s",pwd,\*nameDir);

}

DIR \* dir;

out.cont=0;

dir=opendir(finallyDir);

if(dir==NULL){

printf("Directory non trovata\n");

out.cont=-1;

return &out;

}

struct stat statFile;

struct dirent \* dr;

char filePath[PATH\_MAX];

char filePath2[PATH\_MAX];

DIR \* dir2;

struct dirent \* dr2;

int contAllow=0;

while((dr=readdir(dir))!=NULL && out.cont<6){

snprintf(filePath,PATH\_MAX,"%s/%s",finallyDir,dr->d\_name);

if(stat(filePath,&statFile)<0){

printf("Errore con stat file\n");

closedir(dir);

out.cont=-1;

return &out;

}

else if(S\_ISDIR(statFile.st\_mode)){

dir2 = opendir(filePath);

if(dir2==NULL){

printf("Apertura directory %s null\n",filePath);

out.cont=-1;

return &out;

}

else{

while((dr2=readdir(dir2))!=NULL && contAllow<5){

snprintf(filePath2,PATH\_MAX,"%s/%s",filePath,dr2->d\_name);

if(stat(filePath2,&statFile)<0){

printf("Errore con stat file\n");

closedir(dir);

closedir(dir2);

out.cont=-1;

return &out;

}

else if(S\_ISREG(statFile.st\_mode)){

if(EndsWith(dr2->d\_name,".txt")==1){

contAllow++;

}

}

}

if(!(contAllow<5)){

strncpy(out.directories[out.cont].nameD,filePath,PATH\_MAX);

out.cont++;

}

contAllow=0;

closedir(dir2);

}

}

}

closedir(dir);

return &out;

}

#include <linux/limits.h>

#define N 6

struct DirectoryName{

char nameD[PATH\_MAX];

};

struct ListDir{

DirectoryName directories[N];

int cont;

};

program RPC\_XFILE {

version RPC\_XFILEVERS{

int DELETE\_OCCURENCES(string) = 1;

ListDir SUBDIRECTORY\_LIST(string) = 2;

} = 1;

} = 0x20000013;

File.h

#define RPC\_XFILE 0x20000013

#define RPC\_XFILEVERS 1

#if defined(\_\_STDC\_\_) || defined(\_\_cplusplus)

#define DELETE\_OCCURENCES 1

extern int \* delete\_occurences\_1(char \*\*, CLIENT \*);

extern int \* delete\_occurences\_1\_svc(char \*\*, struct svc\_req \*);

#define SUBDIRECTORY\_LIST 2

extern ListDir \* subdirectory\_list\_1(char \*\*, CLIENT \*);

extern ListDir \* subdirectory\_list\_1\_svc(char \*\*, struct svc\_req \*);

extern int rpc\_xfile\_1\_freeresult (SVCXPRT \*, xdrproc\_t, caddr\_t);

#else /\* K&R C \*/

#define DELETE\_OCCURRENCES 1

extern int \* delete\_occurrences\_1();

extern int \* delete\_occurrences\_1\_svc();

#define SUBDIRECTORY\_LIST 2

extern ListDir \* subdirectory\_list\_1();

extern ListDir \* subdirectory\_list\_1\_svc();

extern int rpc\_xfile\_1\_freeresult ();

#endif /\* K&R C \*/

/\* the xdr functions \*/

#if defined(\_\_STDC\_\_) || defined(\_\_cplusplus)

extern bool\_t xdr\_DirectoryName (XDR \*, DirectoryName\*);

extern bool\_t xdr\_ListDir (XDR \*, ListDir\*);

#else /\* K&R C \*/

extern bool\_t xdr\_DirectoryName ();

extern bool\_t xdr\_ListDir ();

#endif /\* K&R C \*/

#ifdef \_\_cplusplus

}

#endif

#endif /\* !\_RPC\_XFILE\_H\_RPCGEN \*/

**77765445**

**cannot find map.png when it is in the same place**

<https://stackoverflow.com/questions/77765445/cannot-find-map-png-when-it-is-in-the-same-place>

For a school project i have to create what they call is a my radar it so i chose to use a map that represents the world and it does not charge ans it segfault, it has segfaulted when i tried to had tower so the future planes will go to there tower.

#include <SFML/Graphics.h>

#include "../include/struct.h"

#include "../include/my.h"

#include <stdlib.h>

#include <stdio.h>

#define MAX\_TOWERS 10 //

sfSprite\* create\_background(sfRenderWindow\* window, sfVideoMode mode)

{

sfTexture\* background\_texture;

sfSprite\* background;

sfVector2u textureSize;

float scaleX;

float scaleY;

background\_texture = sfTexture\_createFromFile("map.png", NULL);

background = sfSprite\_create();

sfSprite\_setTexture(background, background\_texture, sfTrue);

textureSize = sfTexture\_getSize(background\_texture);

scaleX = (float)mode.width / textureSize.x;

scaleY = (float)mode.height / textureSize.y;

sfSprite\_setScale(background, (sfVector2f){scaleX, scaleY});

return background;

}

void initializing\_game(GameData\* game, sfVideoMode mode) {

game->window = sfRenderWindow\_create(mode, "my\_radar", sfResize | sfClose, NULL);

if (!game->window) {

fprintf(stderr, "Failed to create window\n");

exit(EXIT\_FAILURE); // Exit if there is an error

}

game->background = create\_background(game->window, mode);

if (!game->background) {

fprintf(stderr, "Failed to create background\n");

exit(EXIT\_FAILURE); // Exit if there is an error

}

// Read tower positions from file

FILE\* file = fopen("towers\_pos", "r");

if (!file) {

fprintf(stderr, "Failed to open tower\_positions.txt\n");

exit(EXIT\_FAILURE); // Exit if there is an error

}

// Create towers

for (int i = 0; i < NUM\_TOWERS; i++) {

float x, y;

if (fscanf(file, "%f %f", &x, &y) != 2) {

fprintf(stderr, "Error reading tower position from file\n");

exit(EXIT\_FAILURE); // Exit if there is an error

}

create\_tower(&game->towers[i], x, y, game->window);

}

fclose(file);

}

void processing\_events(GameData\* game)

{

sfEvent event;

while (sfRenderWindow\_pollEvent(game->window, &event)) {

if (event.type == sfEvtClosed)

sfRenderWindow\_close(game->window);

}

}

int main(void)

{

sfVideoMode mode = {800, 600, 32};

GameData game;

initializing\_game(&game, mode);

while (sfRenderWindow\_isOpen(game.window)) {

processing\_events(&game);

sfRenderWindow\_clear(game.window, sfBlack);

sfRenderWindow\_drawSprite(game.window, game.background, NULL);

draw\_towers(&game);

sfRenderWindow\_display(game.window);

}

cleaning\_game(&game);

return 0;

}  
  
here is my code i can had some images i have debugged it said that there is a problem with sfspirte get\_size here is the error message i have

Failed to load image "map.png". Reason: Unable to open file

Segmentation fault (core dumped)

0x00007ffff7f993b4 in sfTexture\_getSize () from /usr/local/lib/libcsfml-graphics.so.2.5

**77769764**

**fwrite causes segmentation fault**

https://stackoverflow.com/questions/77769764/fwrite-causes-segmentation-fault

please point out the mistake I could have made. I am trying to implement a simple file system based on the guide provided [here](https://azrael.digipen.edu/%7Emmead/www/Courses/CS180/Simple-FS.html). I have decided to implement the file system interface as read and writes to a file. The code still needs refactoring but here is the code:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define TOTAL\_FILE\_SIZE 4227104

#define TOTAL\_DIR\_ENTRIES 1024

#define NUM\_FAT\_ENTRIES 8192

typedef struct super\_block {

unsigned short total\_sectors; /\* max value 8192 \*/

unsigned short bytes\_per\_sector; /\* max value 512 \*/

unsigned short available\_sectors; /\* max value 8192 \*/

unsigned short total\_dir\_entries; /\* fixing it to 1024 \*/

unsigned short available\_dir\_entries; /\* max value 1024 \*/

unsigned char label[8]; /\* label for the filesystem \*/

/\* These are not used \*/

unsigned short sectors\_per\_cluster; /\* We always set it to 1\*/

unsigned char reserved[12]; /\* Just making it 32 bytes \*/

} super\_block;

/\* Each dir entry is 16 bytes \*/

typedef struct dir\_entry {

unsigned char file\_name[10];

unsigned short fat\_entry;

unsigned int size;

} dir\_entry;

/\* Each FAT entry is 2 bytes; times 8192 entries => 16 KiB

\* size of data block (sector) can be varied

\* number of dir\_entries can be varied

\* the number of fat entries is same as the number of data blocks

\* the number of data blocks can be varied total\_sectors

\*

\* the total size of our filesystem is => +

\* size of super block = 32 bytes

\* size of dir entry = 16 KiB

\* size of fat = 16 KiB

\* total size of data block = 8192 \* 512 bytes => 4 MiB

\* Total bytes = 4227104

\* End: sFAT.c

\*/

/\* mkfs.c

\*

\*/

/\* Later: integrate label of FS to argument \*/

void mkfs(const char \*filename) {

/\* Write super block, dir entries and fat table and data blocks to the file \*/

super\_block sb = {

.total\_sectors = 8192,

.bytes\_per\_sector = 512,

.available\_sectors = 512,

.total\_dir\_entries = 1024,

.available\_dir\_entries = 1024,

.sectors\_per\_cluster = 1,

.label = "Demo FS",

.reserved = ""

};

dir\_entry default\_dir\_entry = {

.file\_name = "",

.fat\_entry = 0,

.size = 0,

};

unsigned short default\_fat\_entry = 0;

/\* Write sb, 1024 dir\_entry and 8192 default\_fat\_entry and 4 MiB data to create our filesystem \*/

FILE \*f = fopen(filename, "wb");

if (f == NULL) {

fprintf(stderr, "Error opening file: %s\n", filename);

}

/\* Write super block \*/

/\* I seem to encounter SEGMENTATION FAULT in this line \*/

fwrite(&sb, sizeof (super\_block), 1, f);

/\* Write dir entries \*/

fwrite(&default\_dir\_entry, sizeof(dir\_entry), TOTAL\_DIR\_ENTRIES, f);

fwrite(&default\_fat\_entry, sizeof(unsigned short), NUM\_FAT\_ENTRIES, f);

char \*data;

data = malloc(TOTAL\_FILE\_SIZE - ftell(f));

fwrite(data, TOTAL\_FILE\_SIZE - ftell(f), 1, f);

free(data);

fclose(f);

fprintf(stdout, "Filesystem created successfully!\n");

}

/\* End: mkfs.c \*/

/\* main.c

\*

\*/

void display\_help(char \*program\_name) {

fprintf(stdout, "Usage: %s [function name]\n", program\_name);

fprintf(stdout, "Functions:\n");

fprintf(stdout," mkfs\n");

}

int main(int argc, char \*argv[]) {

if (argc < 2) {

display\_help(argv[0]);

return 1;

}

if (strcmp(argv[1], "mkfs") == 0) {

mkfs(argv[2]);

}

else {

display\_help(argv[0]);

return 1;

}

return 0;

}

/\*

\*

\* End: main.c

\*/

I encounter segmentation fault on the fwrite statment of the super block structure. However, if I comment out that line, the file runs successfully. What mistake could I have possibly made?

**77777506**

**The segmentation fault in malloc\_trim(0) when run ASAN**

https://stackoverflow.com/questions/77777506/the-segmentation-fault-in-malloc-trim0-when-run-asan

The code 1.c :

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <malloc.h>

void\* thread\_function(void\* arg) {

char\* ptr = (char \*)malloc(10);

printf("%p\n", ptr);

free(ptr);

malloc\_trim(0);

pthread\_exit(NULL);

}

int main() {

pthread\_t threads[2];

for (long i = 0; i < 2; ++i) {

pthread\_create(&threads[i], NULL, thread\_function, (void\*)i);

}

for (int i = 0; i < 2; ++i) {

pthread\_join(threads[i], NULL);

}

return 0;

}

compile use -fsanitize=address gcc -pthread -o ASANtest -fsanitize=address -O1 1.c And then run ASANtest multiple times. An error occured.

**The error information**:

bash-4.4$ ./ASANtest

ASAN:DEADLYSIGNAL

=================================================================

==7076==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000008 (pc 0x7f95439d87d9 bp 0x000000000fff sp 0x7f953f8fde00 T2)

==7076==The signal is caused by a READ memory access.

==7076==Hint: address points to the zero page.

#0 0x7f95439d87d8 in malloc\_trim (/lib64/libc.so.6+0x9e7d8)

#1 0x400a8f in thread\_function (/workspace/git/ehaauwn/PCPB-22713/2024/eric-pc-routing-engine/raas/product/docker/build/ASANtest+0x400a8f)

#2 0x7f9543d396e9 in start\_thread (/lib64/libpthread.so.0+0xa6e9)

#3 0x7f9543a51a8e in clone (/lib64/libc.so.6+0x117a8e)

AddressSanitizer can not provide additional info.

SUMMARY: AddressSanitizer: SEGV (/lib64/libc.so.6+0x9e7d8) in malloc\_trim

Thread T2 created by T0 here:

#0 0x7f9543f8bc80 in pthread\_create (/usr/lib64/libasan.so.4+0x39c80)

#1 0x400b32 in main (/workspace/git/ehaauwn/PCPB-22713/2024/eric-pc-routing-engine/raas/product/docker/build/ASANtest+0x400b32)

#2 0x400d13 in \_IO\_stdin\_used (/workspace/git/ehaauwn/PCPB-22713/2024/eric-pc-routing-engine/raas/product/docker/build/ASANtest+0x400d13)

==7076==ABORTING

If I compile without -fsanitize=address , the program runs normally.

I want to know if the malloc\_trim function has any security risks, or if there are any bugs in ASAN

**77791078**

**Why am I getting segmentation fault for the pointer here?**

https://stackoverflow.com/questions/77791078/why-am-i-getting-segmentation-fault-for-the-pointer-here

#include <stdio.h>

int main() {

printf("enter the first number: ");

int a;

scanf("%d\n",a);

int \*pa = &a;

printf("enter the second number: ");

int b;

scanf("%d\n",b);

int \*pb = &b;

int \*sum = \*pa + \*pb;

printf("the sum of the given numbers is %d\n", sum);

return 0;

}

**77796077**

**fclose() throws segfault in Docker container**

https://stackoverflow.com/questions/77796077/fclose-throws-segfault-in-docker-container

#include <stdio.h>

#include <string.h>

#include <errno.h>

size\_t create\_empty\_file (char \* out\_fname, size\_t file\_size)

{

FILE \*outfile;

char buf[1024];

outfile = fopen(out\_fname, "wb+");

if (outfile == NULL) {

strcpy(buf, strerror(errno));

printf("%d --> %s\n", errno, buf); }

fseek(outfile, file\_size, SEEK\_SET);

if (errno != 0) {

strcpy(buf, strerror(errno));

printf("%d --> %s\n", errno, buf); }

fputc('\0', outfile);

if (errno != 0) {

strcpy(buf, strerror(errno));

printf("%d --> %s\n", errno, buf); }

fclose( outfile );

return 0;

}

77808359

Setting LD\_PRELOAD path saves the process from getting Segmentation fault

https://stackoverflow.com/questions/77808359/setting-ld-preload-path-saves-the-process-from-getting-segmentation-fault

My application obuspa is crashing when it is executing the standard OpenWRT library call uci\_lookup\_ptr of the libuci.so linked library. Loader is showing this library in the list of linked libraries as below

# /lib/ld-linux.so.3 --list /sbin/obuspa

libmosquitto.so.1 => /usr/lib/libmosquitto.so.1 (0xb68de000)

libubus.so => /usr/lib/libubus.so (0xb68cf000)

libubox.so => /usr/lib/libubox.so (0xb68bb000)

libblobmsg\_json.so => /usr/lib/libblobmsg\_json.so (0xb68b0000)

libuci.so => /lib/libuci.so (0xb689e000)

libc.so.6 => /lib/libc.so.6 (0xb6766000)

/lib/ld-linux.so.3 (0xb6f2d000)

If the same library is set in shell environment using ld\_preload like below, and application is restarted, the process is running fine and not recieving any SIGSEGV signal at all. Hence, no SEGFAULT.

export LD\_PRELOAD=/lib/libuci.so

Below is the code:  
  
static int get\_value\_from\_uci(char \*path, char \*value, size\_t max\_value\_len)

{

struct uci\_context \*uci\_ctx;

struct uci\_ptr ptr;

int ret = 0;

if (!path || !value || max\_value\_len == 0)

return -1;

uci\_ctx = uci\_alloc\_context();

if (!uci\_ctx)

return -1;

printf("%s, uci ctx : %p created, lookup for %s\n", \_\_FUNCTION\_\_, uci\_ctx, path);

if (uci\_lookup\_ptr(uci\_ctx, &ptr, path, true) != UCI\_OK) {

ret = -1;

uci\_free\_context(uci\_ctx);

printf("%s, uci lookup failed\n", \_\_FUNCTION\_\_);

return ret;

}

printf("%s, uci ptr flag:%d\n", \_\_FUNCTION\_\_, ptr.flags);

if ((ptr.flags & UCI\_LOOKUP\_COMPLETE)

&& (ptr.o != NULL)

&& (ptr.o->v.string!=NULL)) {

ret = 0;

USP\_STRNCPY(value, ptr.o->v.string, max\_value\_len);

printf("%s, uci lookup completed, value:%s\n", \_\_FUNCTION\_\_, value);

}

return ret;

}

int main()

{

char cached\_json\_file[256] = {0};

char uci\_role\_path[] = "obuspa.global.dm\_caching\_exclude";

// Read rest of the roles from JSON, if defined

printf("%s, GET UCI Role\n", \_\_FUNCTION\_\_);

if (get\_value\_from\_uci(uci\_role\_path, cached\_json\_file, 256) != 0)

goto exit;

....

return 0;

}

**77812920**

**Aho-Corasick algorithm with C language**

https://stackoverflow.com/questions/77812920/aho-corasick-algorithm-with-c-language

I have programmed an [Aho-Corasick algorithm](https://en.wikipedia.org/wiki/Aho%E2%80%93Corasick_algorithm) with a transition table that searches for a set of words in a text and displays the number of occurrences by using malloc(), but I am encountering this error:

Segmentation fault (core dumped)

Here is the algorithm that I have programmed

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define ALPHABET\_SIZE 70

#define MAX\_WORD\_LENGTH 100000

typedef struct Element Element;

struct Element {

int nombre;

Element \*suivant;

};

typedef struct File File;

struct File {

Element \*premier;

Element \*dernier;

};

File \*creerfile() {

File \*file = (File \*)malloc(sizeof(File));

if (file == NULL) {

perror("Erreur d'allocation pour la file");

exit(EXIT\_FAILURE);

}

file->premier = NULL;

file->dernier = NULL;

return file;

}

int estVide(File \*file) {

return file->premier == NULL;

}

void enfiler(File \*file, int element) {

Element \*nouveau = malloc(sizeof(\*nouveau));

if (nouveau == NULL) {

perror("Erreur d'allocation pour un élément de la file");

exit(EXIT\_FAILURE);

}

nouveau->nombre = element;

nouveau->suivant = NULL;

if (file->premier != NULL) {

file->dernier->suivant = nouveau;

} else {

file->premier = nouveau;

}

file->dernier = nouveau;

}

int defiler(File \*file) {

if (file == NULL || file->premier == NULL) {

perror("Erreur de défiler : file vide");

exit(EXIT\_FAILURE);

}

int nombreDefile = file->premier->nombre;

Element \*elementDefile = file->premier;

file->premier = elementDefile->suivant;

free(elementDefile);

if (file->premier == NULL) {

file->dernier = NULL;

}

return nombreDefile;

}

struct \_trie {

int maxNode;

int nextNode;

int \*\*transition;

char \*finale;

int \*supp;

};

typedef struct \_trie Trie;

void initialisationTrie(Trie \*trie, int maxNode) {

trie->maxNode = maxNode;

trie->nextNode = 1;

trie->transition = (int \*\*)malloc(maxNode \* sizeof(int \*));

if (trie->transition == NULL) {

perror("Erreur d'allocation pour le tableau de transitions");

exit(EXIT\_FAILURE);

}

for (int i = 0; i < maxNode; ++i) {

trie->transition[i] = (int \*)malloc(ALPHABET\_SIZE \* sizeof(int));

if (trie->transition[i] == NULL) {

perror("Erreur d'allocation pour une ligne du tableau de transitions");

exit(EXIT\_FAILURE);

}

for (int j = 0; j < ALPHABET\_SIZE; ++j) {

trie->transition[i][j] = 0;

}

}

trie->finale = (char \*)malloc(maxNode \* sizeof(char));

if (trie->finale == NULL) {

perror("Erreur d'allocation pour le tableau finale");

exit(EXIT\_FAILURE);

}

for (int i = 0; i < maxNode; ++i) {

trie->finale[i] = 0;

}

trie->supp = (int \*)malloc(maxNode \* sizeof(int));

if (trie->supp == NULL) {

perror("Erreur d'allocation pour le tableau supp");

exit(EXIT\_FAILURE);

}

for (int i = 0; i < maxNode; ++i) {

trie->supp[i] = 0;

}

}

void ajoutmot(Trie \*trie, char mot[MAX\_WORD\_LENGTH]) {

int courant = 0;

for (int i = 0; i < strlen(mot); ++i) {

char c = mot[i];

int index;

if (c >= 'a' && c <= 'z') {

index = c - 'a';

} else {

// Caractère spécial, traitez-le comme une lettre

index = c - 'A' + 26; // Ajoutez 26 pour l'ajustement

}

if (trie->transition[courant][index] == 0) {

trie->transition[courant][index] = trie->nextNode;

trie->nextNode++;

}

courant = trie->transition[courant][index];

}

trie->finale[courant] = 1;

}

void constructionsuppleants(Trie \*trie) {

File \*file = creerfile();

for (int i = 0; i < ALPHABET\_SIZE; ++i) {

if (trie->transition[0][i] != 0) {

enfiler(file, trie->transition[0][i]);

trie->supp[trie->transition[0][i]] = 0;

}

}

while (!estVide(file)) {

int r = defiler(file);

for (int i = 0; i < ALPHABET\_SIZE; ++i) {

int s = trie->transition[r][i];

if (s != 0) {

enfiler(file, s);

int etat = trie->supp[r];

while (trie->transition[etat][i] == 0 && etat != 0) {

etat = trie->supp[etat];

}

trie->supp[s] = trie->transition[etat][i] != 0 ? trie->transition[etat][i] : 0;

}

}

}

free(file);

}

int AhoCorrasick(Trie \*trie, char \*text) {

int cmpt = 0;

int etat = 0;

for (int i = 0; text[i] != '\0'; ++i) {

char c = text[i];

int index;

if (c >= 'a' && c <= 'z') {

index = c - 'a';

} else {

// Traitez le caractère spécial comme une lettre

index = c - 'a' + 26;

}

while (trie->transition[etat][index] == 0 && etat != 0) {

etat = trie->supp[etat];

}

etat = trie->transition[etat][index] != 0 ? trie->transition[etat][index] : 0;

int etatTemporaire = etat;

while (etatTemporaire != 0) {

if (trie->finale[etatTemporaire] == 1) {

cmpt++;

}

etatTemporaire = trie->supp[etatTemporaire];

}

}

return cmpt;

}

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "aho\_corrasick\_matrice.c"

#include "aho\_corrasick\_matrice.h"

#define ALPHABET\_SIZE 70

#define BUFFER\_SIZE 4096

#define MAX\_WORD\_LENGTH 100000

int main(int argc, char \*argv[]) {

if (argc != 3) {

fprintf(stderr, "Usage: %s mots.txt texte.txt\n", argv[0]);

exit(EXIT\_FAILURE);

}

Trie trie;

long int maxNode = 500000;

initialisationTrie(&trie, maxNode);

FILE \*motsFile = fopen(argv[1], "r");

if (motsFile == NULL) {

fprintf(stderr, "Erreur lors de l'ouverture du fichier %s\n", argv[1]);

exit(EXIT\_FAILURE);

}

char mot[MAX\_WORD\_LENGTH];

while (fscanf(motsFile, "%s", mot) == 1) {

ajoutmot(&trie, mot);

}

fclose(motsFile);

constructionsuppleants(&trie);

FILE \*texteFile = fopen(argv[2], "r");

if (texteFile == NULL) {

fprintf(stderr, "Erreur lors de l'ouverture du fichier %s\n", argv[2]);

exit(EXIT\_FAILURE);

}

char buffer[BUFFER\_SIZE];

int occurrences = 0;

while (fgets(buffer, sizeof(buffer), texteFile) != NULL) {

occurrences += AhoCorrasick(&trie, buffer);

}

printf("Nombre d'occurrences est : %d\n", occurrences);

fclose(texteFile);

free(trie.finale);

for (int i = 0; i < maxNode; ++i) {

free(trie.transition[i]);

}

free(trie.transition);

free(trie.supp);

return 0;

}

77821810

Segmentation Fault with structs c

https://stackoverflow.com/questions/77821810/segmentation-fault-with-structs-c

#include "stdio.h"

typedef struct

{

char name[50];

int money;

char IBAN[50];

char passoword[20];

char email[30];

}account;

void create\_account(account \*ptr)

{

printf("\*\*\*\*Welcome to account resgister\*\*\*\*\n");

printf("Whats your name?: \n");

scanf(" %49s",ptr->name);

printf("Whats your email? : ");

scanf(" %29s",ptr->email);

printf("Type you password (from 4 to 20): ");

scanf(" %19s",ptr->passoword);

}

int menu()

{

char option = 0;

while(option != '0' && option != '1' && option != '2')

{

printf("\*\*\*\*MENU\*\*\*\*\n");

printf("0 - Exit\n");

printf("1 - Create Account\n");

printf("2 - Log in\n");

printf("What you want to do: \n");

scanf(" %c", &option);

}

return option;

}

int main (void)

{

char option;

account \*ptr;

while(option != '0')

{

option = menu();

switch (option)

{

case '0':

break;

case '1':

create\_account(ptr);

break;

}

}

}

Can someone tell me when i choose my option 1 in my menu and then when im asked to write my name i write it and them gives me a segmentation fault in the line : scanf(" %49", ptr->name). I dont know why this is happening like before i had %50s and i look it up and changed to %49s to match my array size but the error persists. Thx

**77826658**

**Segfault in C while Implementing a Simple Dictionary**

https://stackoverflow.com/questions/77826658/segfault-in-c-while-implementing-a-simple-dictionary

So I tried implementing a simple dictionary as shown in this [post](https://stackoverflow.com/questions/4384359/quick-way-to-implement-dictionary-in-c). However, I did modify the code:

a) The dictionary:

/\* String utility for duplication \*/

char \*strdup(char \*s) /\* make a duplicate of s \*/

{

char \*p;

p = (char \*) malloc(strlen(s)+1); /\* +1 for ’\0’ \*/

if (p != NULL)

strcpy(p, s);

return p;

}

struct nlist {

struct nlist \*next;

char \*name;

char \*defn;

};

#define HASHSIZE 101

struct Dict {

struct nlist \*hashtab[HASHSIZE];

};

/\* hash: form hash value for string s \*/

unsigned hash(char \*s)

{

unsigned hashval;

for (hashval = 0; \*s != '\0'; s++)

hashval = \*s + 31 \* hashval;

return hashval % HASHSIZE;

}

/\* lookup: look for s in hashtab \*/

struct nlist \*lookup(struct Dict \*d, char \*s)

{

struct nlist \*np;

for (np = d->hashtab[hash(s)]; np != NULL; np = np->next)

if (strcmp(s, np->name) == 0)

return np; /\* found \*/

return NULL; /\* not found \*/

}

struct nlist \*insert(struct Dict \*d, char \*name, char \*defn)

{

struct nlist \*np;

unsigned hashval;

if ((np = lookup(d, name)) == NULL) { /\* not found \*/

np = (struct nlist \*) malloc(sizeof(\*np));

if (np == NULL || (np->name = strdup(name)) == NULL)

return NULL;

hashval = hash(name);

np->next = d->hashtab[hashval];

d->hashtab[hashval] = np;

} else /\* already there \*/

free((void \*) np->defn); /\*free previous defn \*/

if ((np->defn = strdup(defn)) == NULL)

return NULL;

return np;

}

b) Example:

int main() {

struct Dict \*dtest = (struct Dict\*)malloc(sizeof(struct Dict));

insert(dtest, "hello", "world");

// Look up a definition by name

struct nlist \*result = lookup(dtest, "hello");

if (result != NULL) {

printf("Definition for 'hello': %s\n", result->defn);

} else {

printf("'hello' is not found in the dictionary.\n");

}

return 0;

}

**77828583**

**Why can’t I just add an int to a memory address to address any array?**

<https://stackoverflow.com/questions/77828583/why-cant-i-just-add-an-int-to-a-memory-address-to-address-any-array>

I am trying to override part of my first string with my second one with an offset of 2.

Here is my second attempt that works:

#include <stdio.h>

#include <string.h>

int main() {

char s[] = "Hello, World!";

char a[] = "aaa";

printf("%d\n", &s);

printf("%d\n", &s[2]);

memcpy(&s[2], a, sizeof(a)-1);

printf("%s\n", s);

return 0;

}

$ gcc -o test test2.c && ./test

520783106

520783108

Heaaa, World!

And here is my first attempt, that does not work, raises a warning at compilation and segfaults:

#include <stdio.h>

#include <string.h>

int main() {

char s[] = "Hello, World!";

char a[] = "aaa";

printf("%d\n", &s);

printf("%d\n", sizeof(char)\*2);

printf("%d\n", &s + (sizeof(char)\*2));

memcpy(&s + (sizeof(char)\*2), a, sizeof(a)-1);

printf("%s\n", s);

return 0;

}

$ gcc -o test test1.c && ./test

test1.c: In function ‘main’:

test1.c:12:4: warning: ‘memcpy’ writing 3 bytes into a region of size 0 overflows the destination [-Wstringop-overflow=]

12 | memcpy(&s + (sizeof(char)\*2), a, sizeof(a)-1);

| ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

test1.c:5:9: note: at offset 28 into destination object ‘s’ of size 14

5 | char s[] = "Hello, World!";

| ^

1547566642

2

1547566670

Hello, World!

Segmentation fault

Can someone explain to me why test1 is not working ?

Why is the sum of 1547566642 and 2 equal 1547566670 and not 1547566644 ?

I assume it's a type problem. The return of sizeof is size\_t but what is the type of &s ?

Thank you in advance for your help. ;)

**77842946**

**Program received signal SIGSEGV, Segmentation fault - No such file or directory**

<https://stackoverflow.com/questions/77842946/program-received-signal-sigsegv-segmentation-fault-no-such-file-or-directory>

I am running gdb to debug a program with a segmentation fault error. When I build my app and run

gdb ./app

I get the following output:

Program received signal SIGSEGV, Segmentation fault.

\_\_strcpy\_avx2 () at ../sysdeps/x86\_64/multiarch/strcpy-avx2.S:611

611 ../sysdeps/x86\_64/multiarch/strcpy-avx2.S: No such file or directory.

Running bt gives,

#0 \_\_strcpy\_avx2 () at ../sysdeps/x86\_64/multiarch/strcpy-avx2.S:611