



دانشگاه علم و صنعت

دانشکده مهندسی کامپیوتر

درجه تحصیلی: کارشناسی

## گزارشکار تکلیف OS 2

گردآورنده:

پرnian شاكریان - 99400064

استاد:

دکتر انتظاری

سال تحصیلی: اسفند ۱۴۰۱

تمام کتابخانه ها و headerهای مورد نیاز ما برای تمرین ۲ به طور کلی و برای آسانی کار اینجا نوشته و اشاره شده است.

```
#include <assert.h>
```

```
#include <stdio.h>
```

```
#include <pthread.h>
```

```
#include <stdlib.h>
```

```
#include <unistd.h>
```

```
#include <sys/types.h>
```

```
#include <sys/stat.h>
```

```
#include <sys/wait.h>
```

```
#include <string.h>
```

برای ساخت makefile ها از لینک های زیر کمک گرفته ام.

<https://youtu.be/bPMDEdjKBWA>

[https://youtu.be/\\_r7i5X0rXJk](https://youtu.be/_r7i5X0rXJk)

سوال ۳: یه برنامه بنویسید که روت یک درخت را داریم و به دنبال عدد  $X$  میگردیم. با استفاده از multi-process مسیر root تا node که حاوی عدد  $X$  هست را چاپ کنید مسئله را به صورت divide and conquer حل کنید.

به طور کلی این برنامه یک الگوریتم جستجو را برای ساختار داده درختی پیاده سازی می کند. درخت به عنوان یک گره ساختاری نشان داده می شود که حاوی یک مقدار و آرایه ای از اشاره گرها به گره های فرزند خود است. این برنامه یک درخت با یک گره ریشه و سه سطح از گره های فرزند ایجاد می کند و سپس با استفاده از تابع `search_tree` به دنبال یک گره با یک مقدار خاص می شود. تابع جستجو دارای چهار آرگومان است: یک اشاره گر به گره فعلی مورد جستجو، مقدار مورد جستجو، یک آرایه برای ذخیره مسیر از ریشه به گره با مقدار، و سطح فعلی درخت مورد جستجو.

تابع `search_tree` به صورت بازگشتی درخت را با بررسی هر گره فرزند از گره فعلی برای مقدار جستجو می کند. اگر مقدار در یکی از گره های فرزند یافت شود، تابع مسیر آن گره را در آرایه مسیر ذخیره می کند و `true` را برمی گرداند. اگر مقدار در هیچ یک از گره های فرزند یافت نشد، تابع `false` را برمی گرداند.

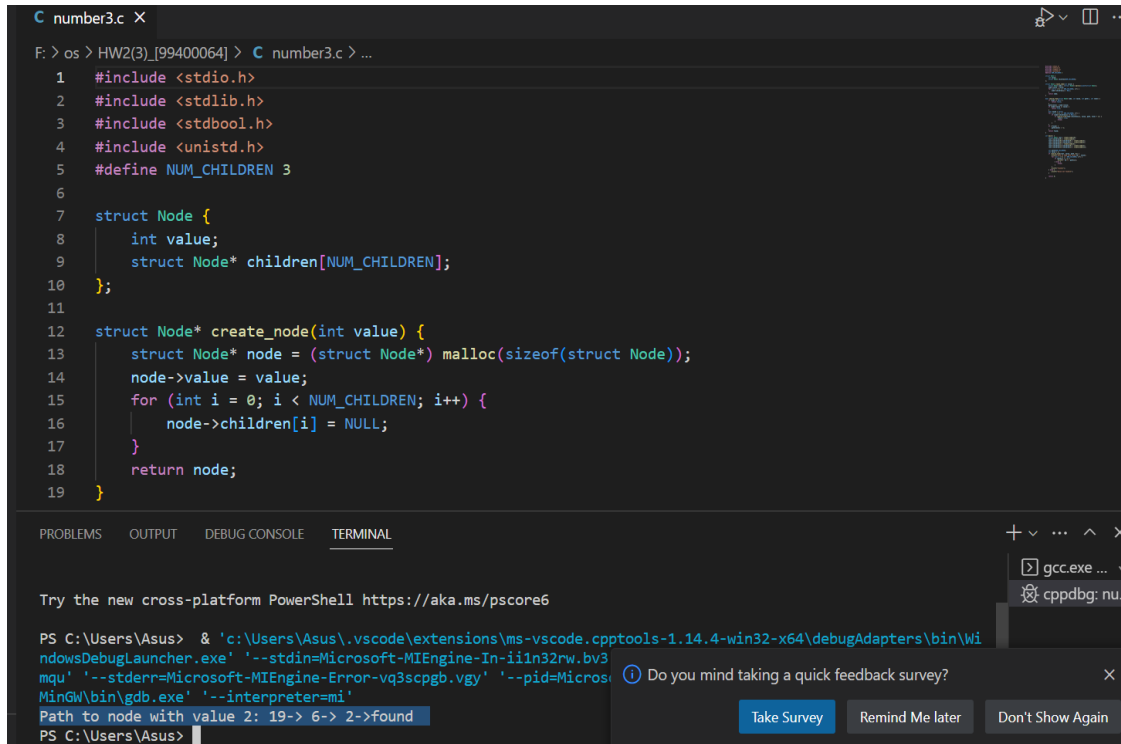
در تابع `main`، برنامه یک درخت ایجاد می کند، مقداری را برای جستجو تعیین می کند و تابع `search_tree` را برای جستجوی آن مقدار اجرا می کند. اگر مقدار پیدا شد، برنامه مسیر را از ریشه به گره با مقدار چاپ می کند. اگر مقدار پیدا نشد، برنامه پیامی را چاپ می کند که نشان می دهد مقدار پیدا نشد.

اگر بخواهیم خط به خط کد را توضیح دهیم تابع اول یک مقدار صحیح را به عنوان ورودی می گیرد و یک اشاره گر را به ساختار `Node` جدید برمی گرداند. این تابع یک نمونه `Node` جدید با استفاده از `malloc` ایجاد کرده تا حافظه را به صورت پویا به آن اختصاص دهد. این تابع، فیلد مقدار گره تازه ایجاد شده را روی مقدار ورودی تنظیم می کند و تمام اشاره گرهای فرزند خود را به `NULL` مقداردهی اولیه می کند.

سپس از خط ۲۱ تابع از طریق درخت یک مقدار صحیح داده شده را جستجو می کند. چهار آرگومان نیاز دارد که به آن اشاره شد این تابع با بررسی NULL بودن گره فعلی شروع می شود. در این صورت، تابع بلافاصله false را برمی گرداند سپس، تابع مقدار گره فعلی را در آرایه مسیر در سطح فعلی ذخیره می کند. سپس بررسی می کند که آیا مقدار گره فعلی با مقدار ورودی مطابقت دارد یا خیر. اگر چنین است، تابع true را برمی گرداند. اگر مقدار ورودی در گره فعلی یافت نشد، تابع یک متغیر بولین یافت شده را روی false تنظیم می کند و روی هر فرزند گره فعلی حلقه می زند. اگر یک گره فرزند داده شده NULL نباشد، تابع به صورت بازگشتی جستجو را از طریق فرزند فراخوانی می کند. اگر جستجو از طریق فرزند true را برگرداند، این نشان می دهد که مقدار ورودی در جایی در زیر درخت آن پیدا شده است، بنابراین پرچم یافت شده روی true تنظیم می شود و حلقه روی فرزندان دیگر با استفاده از break خاتمه می یابد. if نهایی بررسی می کند که آیا مقدار پیدا شده است، همانطور که با بولی پیدا شده نشان داده شده است. اگر پیدا نشد، ورودی مسیر [سطح] روی ۱- تنظیم می شود. سپس تابع پیدا شده را برمی گرداند. این تابع یک درخت ساده تنظیم می کند و سپس تابع search\_tree را با این درخت به عنوان ورودی فراخوانی می کند تا مشخص کند که آیا مقدار ۲ در درخت یافت می شود یا خیر. برنامه با فراخوانی create\_node برای تخصیص حافظه برای گره های جدید و تنظیم مقادیر آنها شروع می شود. پس از این، برنامه یک مسیر آرایه عدد صحیح را با NUM\_CHILDREN (که در حال حاضر به عنوان ۳ تعریف می شود) مقداردهی می کند تا مسیری را که از طریق درخت طی شده ذخیره کند. برنامه مقدار ۲ را به مقدار متغیر صحیح اختصاص می دهد و تابع search\_tree را برای ذخیره مسیر پیمایش فراخوانی می کند و با تمام ۰ها مقدار دهی اولیه می شود. (عدد صحیح ۰ به عنوان شاخص مسیر شروع)

دستور if در کد بررسی می کند که آیا تابع search\_tree() true برمی گرداند یا خیر. اگر مقدار را بیابد، برنامه مسیری را که در درخت از ریشه تا گره هدف طی شده است چاپ می کند. اگر مقدار پیدا نشد، برنامه پیامی مبنی بر یافت نشدن آن خروجی می دهد. پس از اتمام، برنامه ۰ را برمی گرداند.

خروجی مثال من:



The image shows a Visual Studio Code editor window with a C program named 'number3.c'. The code defines a linked list structure with a 'Node' struct containing an 'int value' and a pointer to the next 'Node'. A 'create\_node' function is implemented, which allocates memory for a new node and initializes its children array. The terminal output shows the execution of the program, which prints the path to a node with value 2: 19-> 6-> 2->found. A feedback survey prompt is also visible in the terminal.

```
C number3.c X
F: > os > HW2(3)_[99400064] > C number3.c > ...
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4  #include <unistd.h>
5  #define NUM_CHILDREN 3
6
7  struct Node {
8      int value;
9      struct Node* children[NUM_CHILDREN];
10 };
11
12 struct Node* create_node(int value) {
13     struct Node* node = (struct Node*) malloc(sizeof(struct Node));
14     node->value = value;
15     for (int i = 0; i < NUM_CHILDREN; i++) {
16         node->children[i] = NULL;
17     }
18     return node;
19 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS C:\Users\Asus> & 'c:\Users\Asus\.vscode\extensions\ms-vscode.cpptools-1.14.4-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-ii1n32rw.bv3mqu' '--stderr=Microsoft-MIEngine-Error-vq3scpgb.vgy' '--pid=Micros: MinGW\bin\gdb.exe' '--interpreter=mi'

Path to node with value 2: 19-> 6-> 2->found

PS C:\Users\Asus>

Do you mind taking a quick feedback survey?

Take Survey Remind Me later Don't Show Again