

Starfork

Stanley Lin and Alex Xu

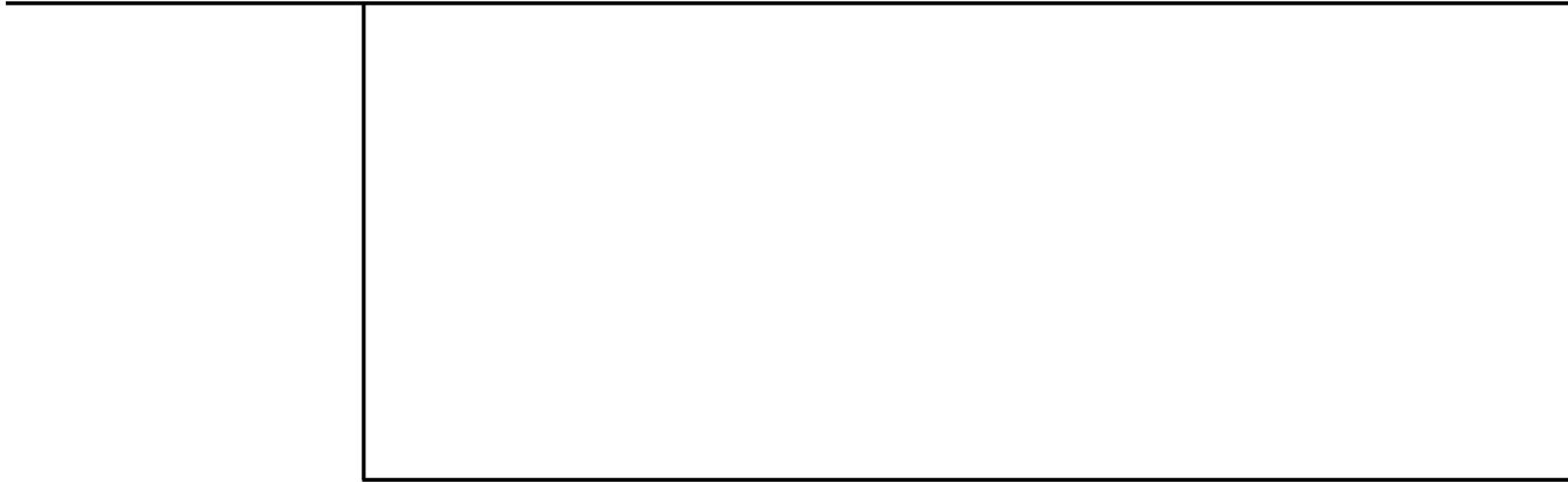
COMS 3157 SP23
Recitation 7
3/24/2023

./starfork-s2 1

```
$ ./starfork-s2 1
*
```

n=1 i=1

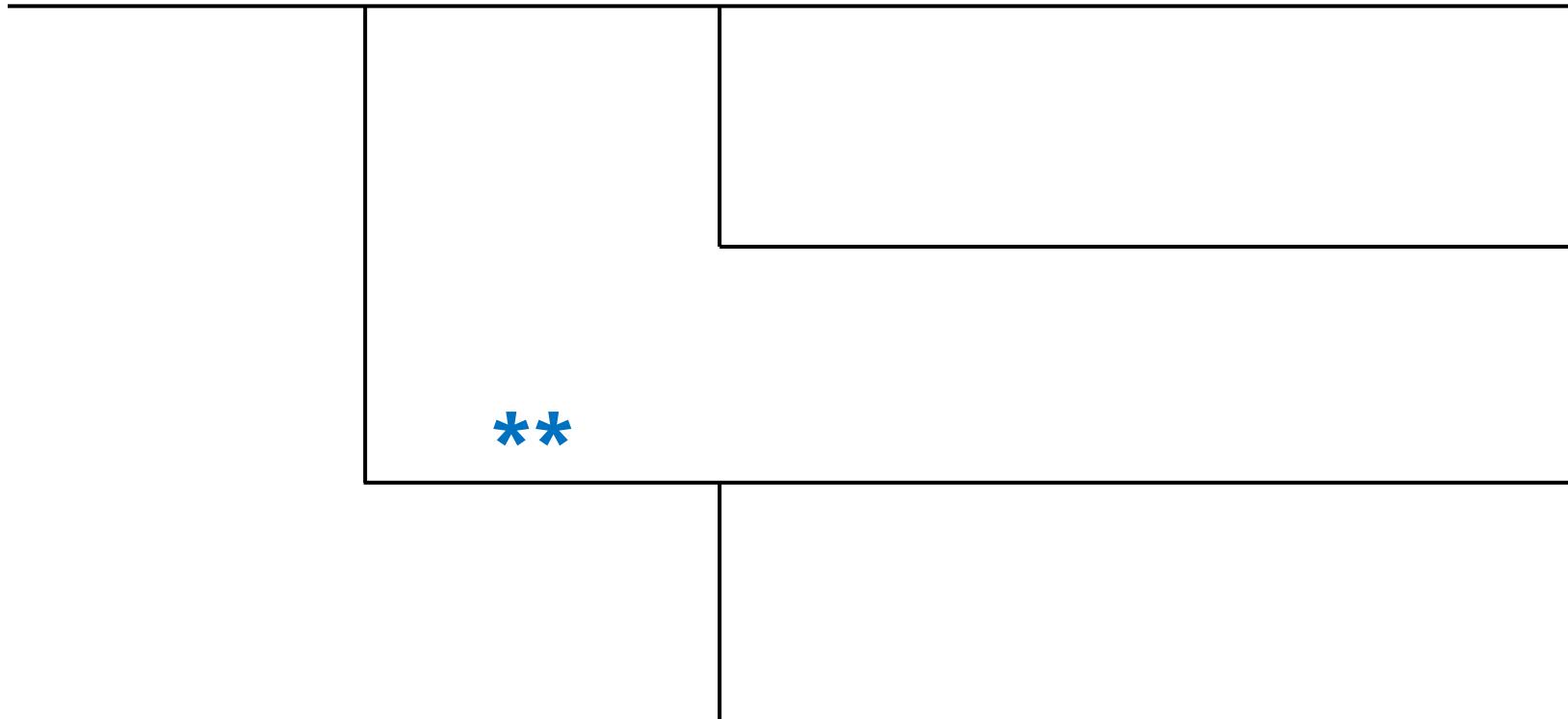
*



./starfork-s2 2

n=2 i=1 i=2

***** ******



```
$ ./starfork-s2 2
*
**
**
**
```

./starfork-s2 3

n=3 i=1 i=2 i=3

✓	✓	✗
*	*	*
**	**	***
**	***	**
***	**	***
***	***	**
***	***	***
***	***	***

Unpredictable, but not
totally random!

./starfork-s3 1

n=1

i=1

* *



./starfork-s3 3

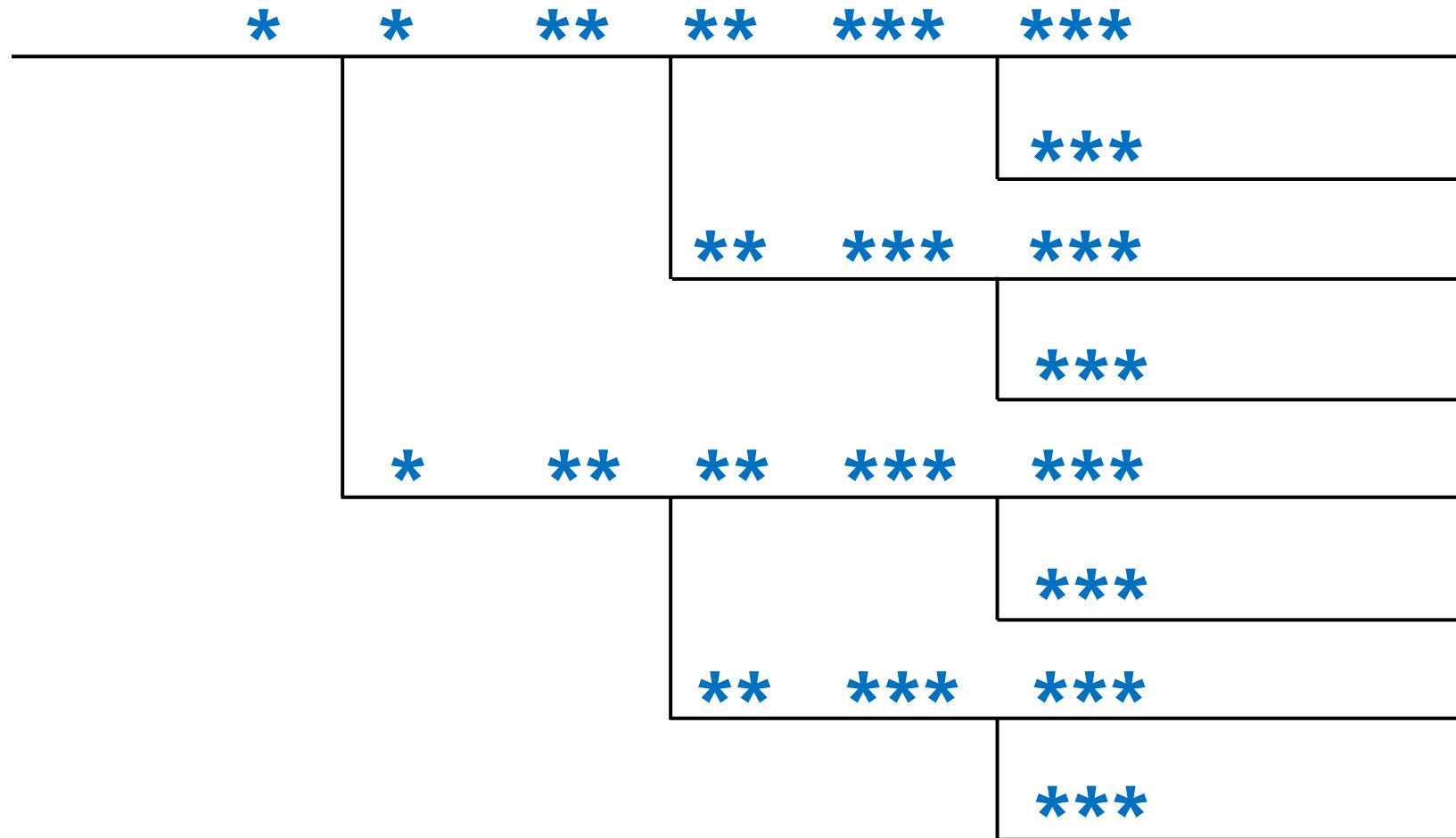
Unpredictable when n > 1

n=3

i=1

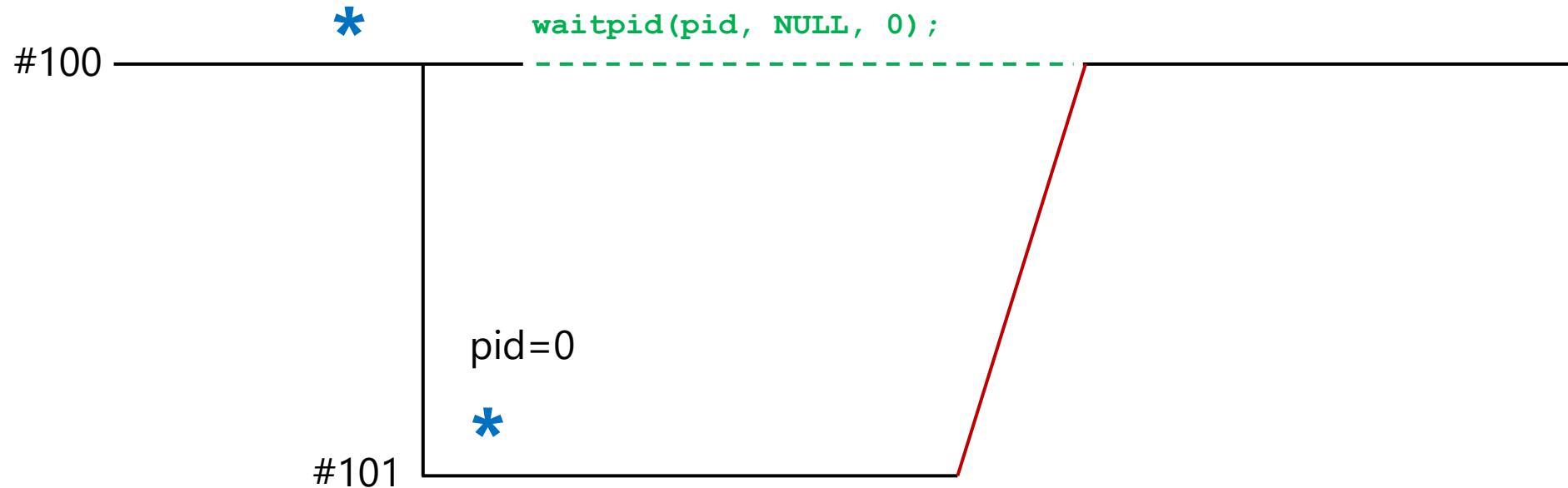
i=2

i=3

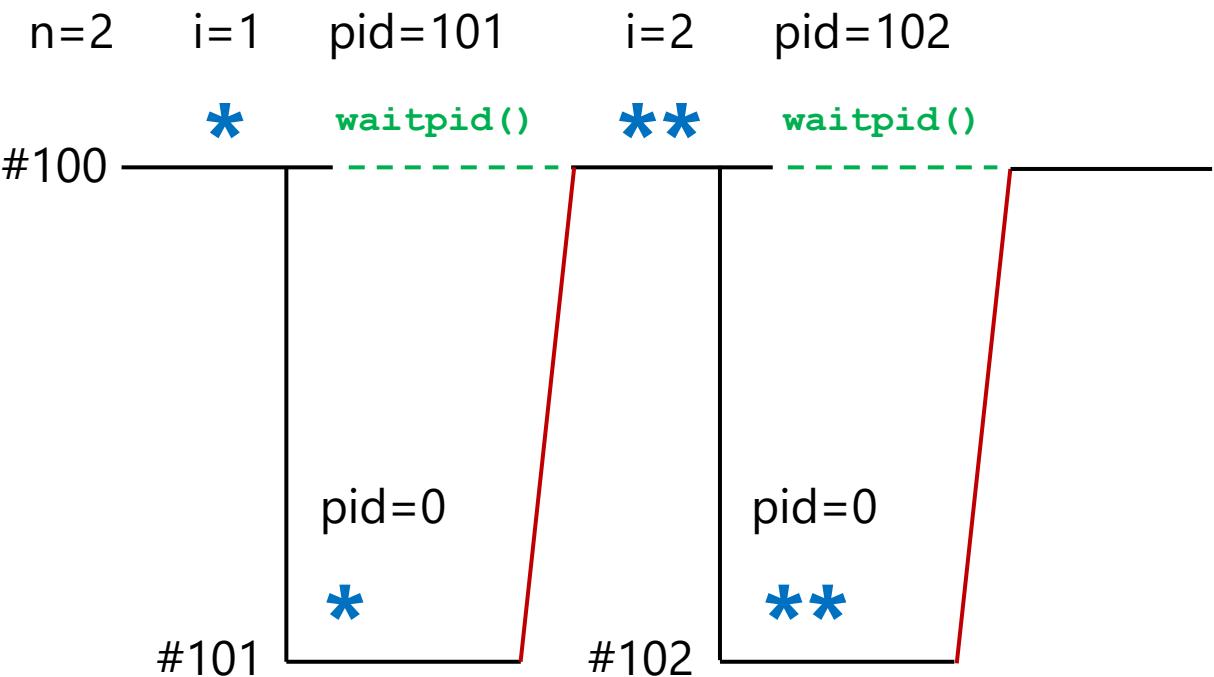


./starfork-s4 1

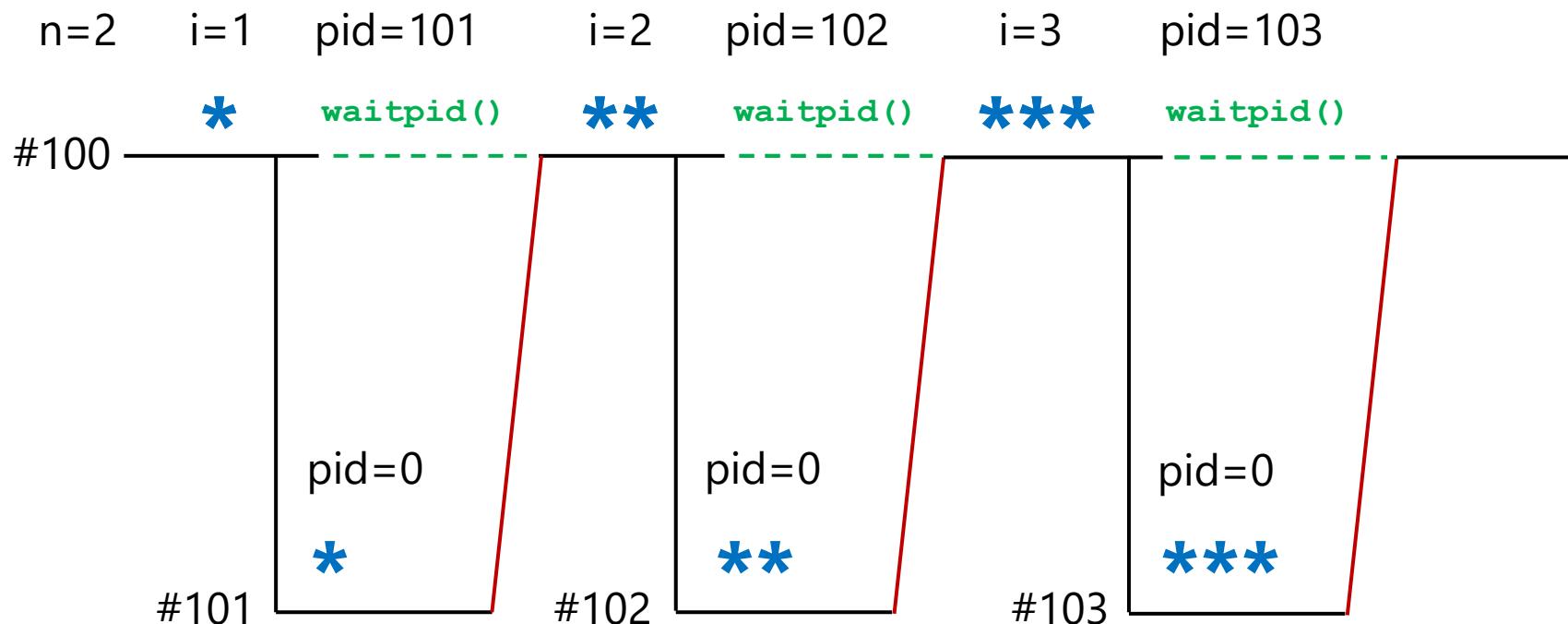
n=1 i=1 pid=101



./starfork-s4 2



./starfork-s4 3

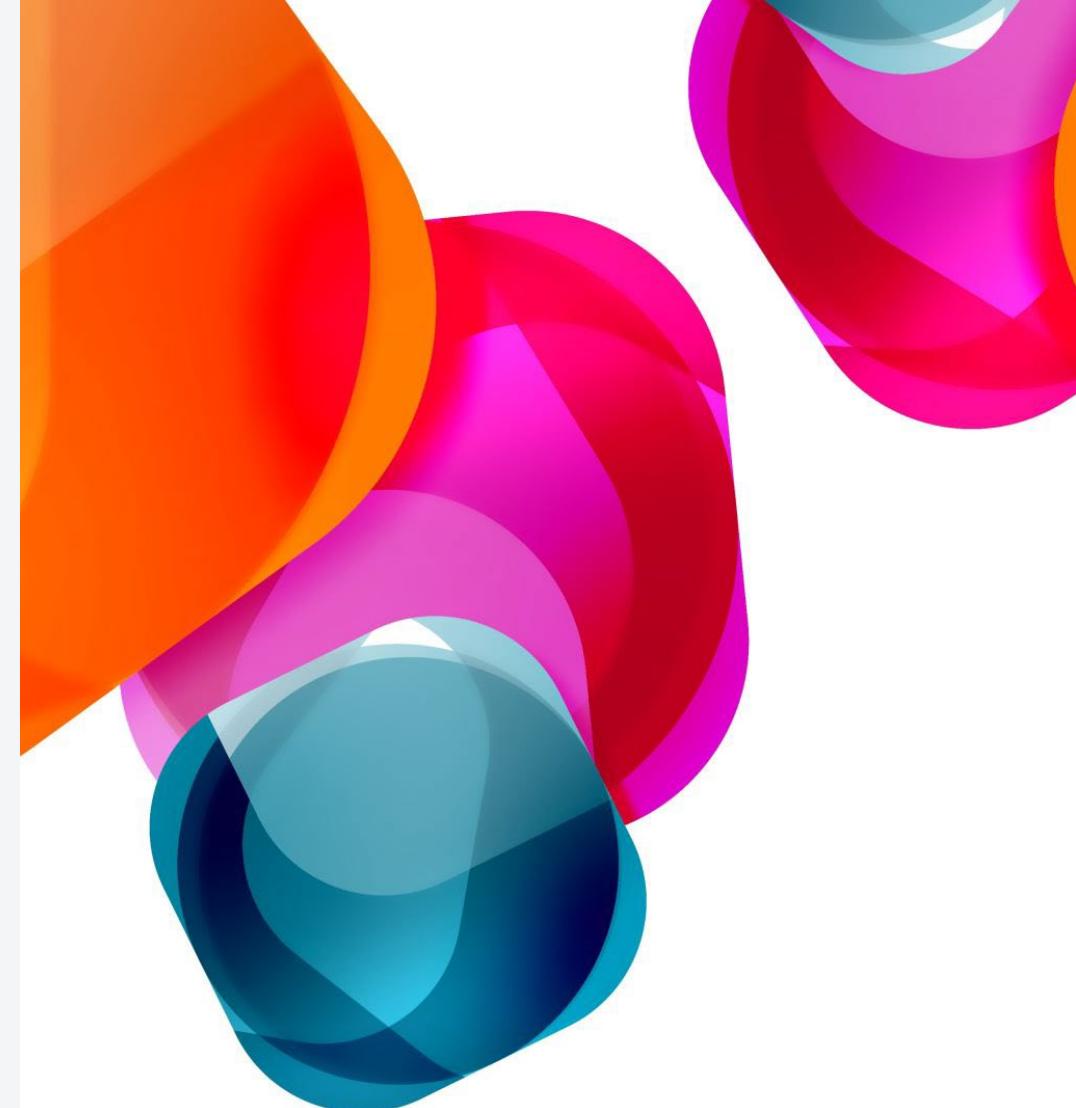


Recitation 7

fork, waitpid & exec

March 24, 2023

Alex XU



Part-5

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
  
        pid_t pid = fork();  
  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
  
            star(i);  
  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

How many lines are in the stdout outputs when running with arguments 1, 2 or 3?

Part-5

\$./starfork-s5 1 2>/dev/null

```
int main(int argc, char **argv) { n=1    i=1  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

stdout

Part-5

\$./starfork-s5 1 2>/dev/null

```
int main(int argc, char **argv) { n=1    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

stdout

*

Part-5

\$./starfork-s5 1 2>/dev/null

```
int main(int argc, char **argv) { n=1    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

stdout

*

Part-5

\$./starfork-s5 1 2>/dev/null

```
int main(int argc, char **argv) { n=1   i=1   *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

stdout

*

Part-5

\$./starfork-s5 1 2>/dev/null

```
int main(int argc, char **argv) { n=1   i=1   *  
    int n = atoi(argv[1]);  
  
    → for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

i=2

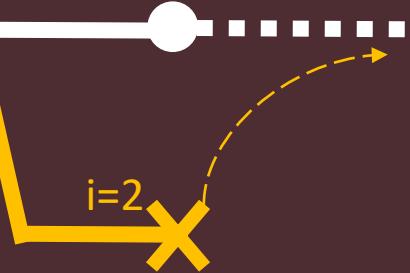
stdout

*

Part-5

\$./starfork-s5 1 2>/dev/null

```
int main(int argc, char **argv) { n=1   i=1   *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



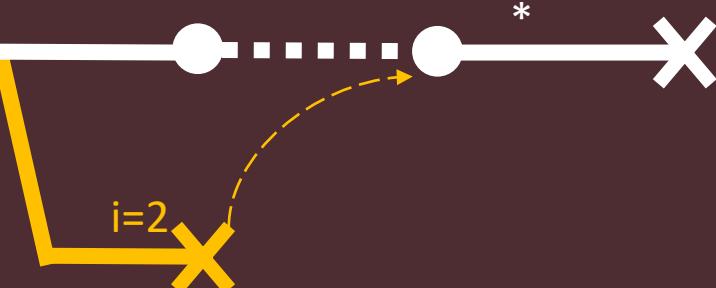
stdout
*

Part-5

\$./starfork-s5 1 2>/dev/null

```
int main(int argc, char **argv) {    n=1    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

stdout
*
*



Part-5

\$./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) { n=2    i=1  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

stdout

Part-5

\$./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) { n=2    i=1    *
int n = atoi(argv[1]);
for (int i = 1; i <= n; i++) {
    star(i);
    pid_t pid = fork();
    if (pid > 0) { // Parent
        waitpid(pid, NULL, 0);
        star(i);
        exit(EXIT_SUCCESS);
    }
}
```

stdout
*

Part-5

\$./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) { n=2    i=1    *
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

stdout
*

Part-5

\$./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) { n=2    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



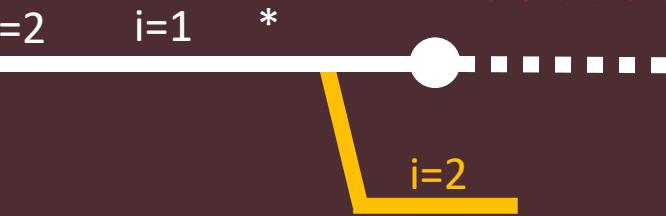
stdout

*

Part-5

\$./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) { n=2    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



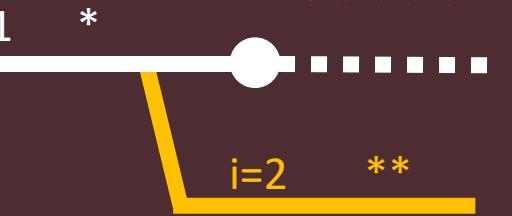
stdout

*

Part-5

\$./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) { n=2    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



stdout

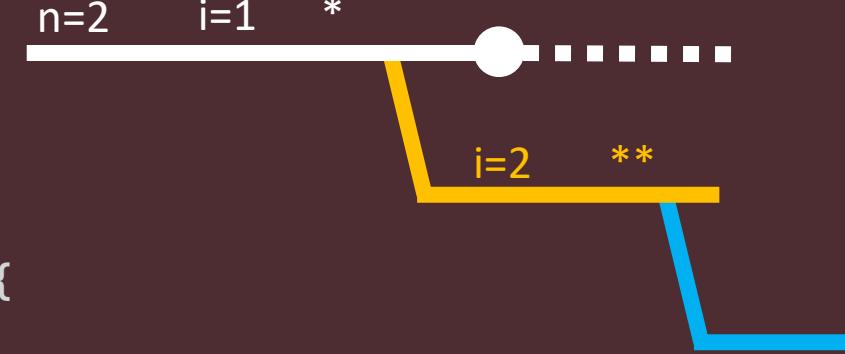
*

**

Part-5

\$./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) { n=2    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



stdout

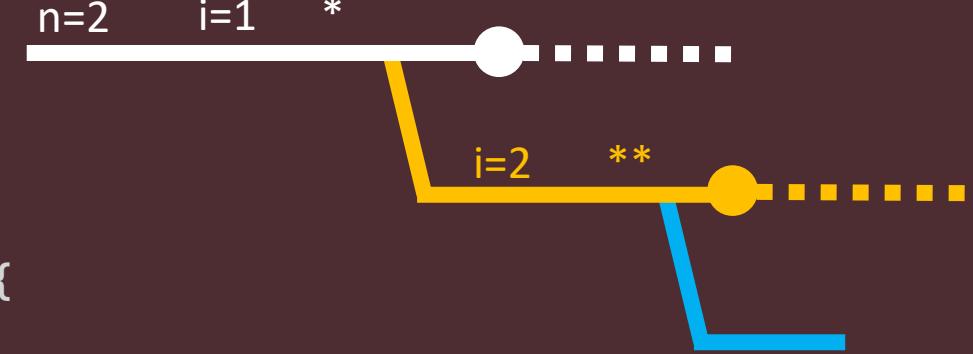
*

**

Part-5

\$./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) { n=2    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



stdout

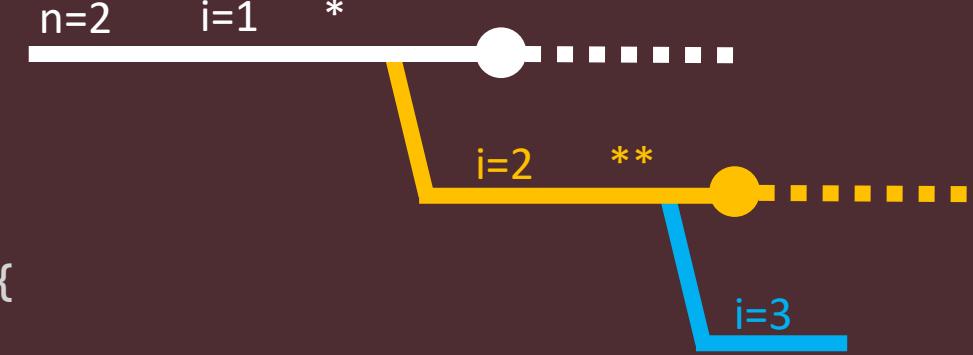
*

**

Part-5

\$./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) { n=2    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
  
        pid_t pid = fork();  
  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
  
            star(i);  
  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



stdout

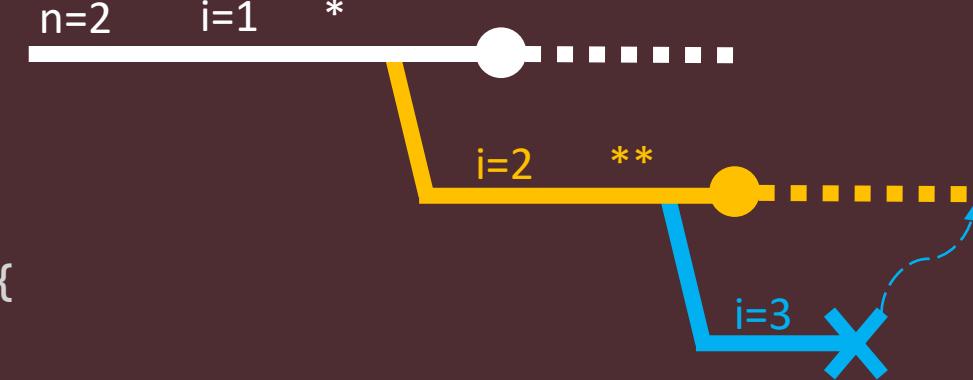
*

**

Part-5

\$./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) { n=2    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
  
        pid_t pid = fork();  
  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

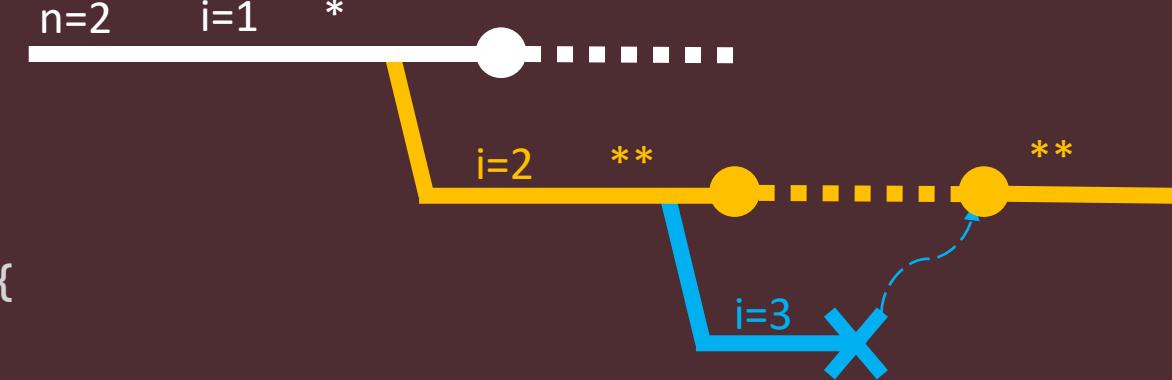


stdout
*
**

Part-5

\$./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) { n=2    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
  
        pid_t pid = fork();  
  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

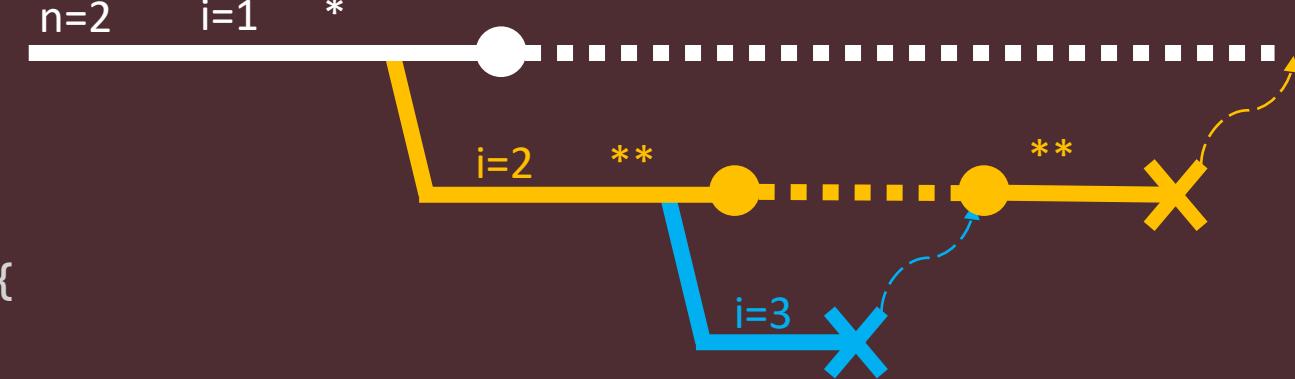


stdout
*
**
**

Part-5

\$./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) { n=2    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
  
        pid_t pid = fork();  
  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



stdout
*
**
**

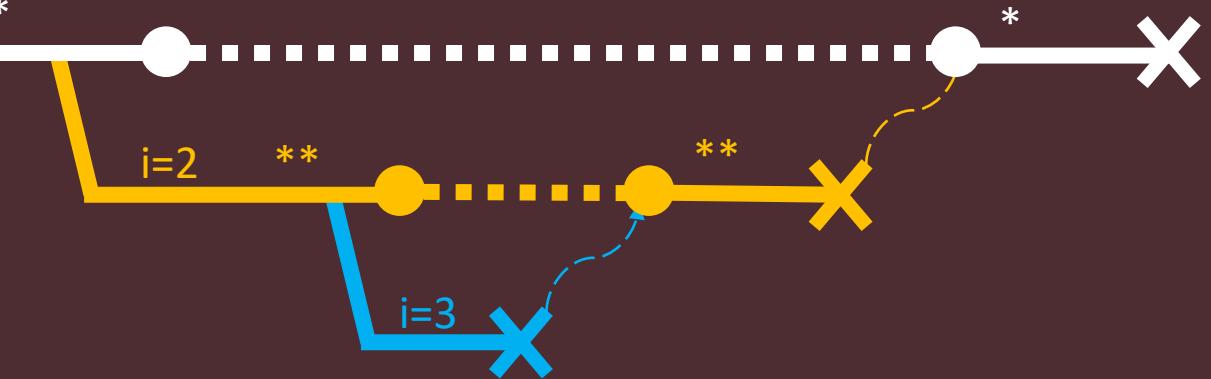
Part-5

\$./starfork-s5 2 2>/dev/null

```
int main(int argc, char **argv) { n=2 i=1 *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
  
        pid_t pid = fork();  
  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

stdout

```
*  
**  
**  
*
```



Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3    i=1  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

stdout

Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3    i=1    *
int n = atoi(argv[1]);
for (int i = 1; i <= n; i++) {
    star(i);
    pid_t pid = fork();
    if (pid > 0) { // Parent
        waitpid(pid, NULL, 0);
        star(i);
        exit(EXIT_SUCCESS);
    }
}
```

stdout

*

Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3   i=1   *
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

stdout
*

Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3   i=1   *
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

The diagram illustrates the state of variables at the beginning of the loop iteration. The variable `n` is set to 3, `i` is set to 1, and `*` is a pointer to a character. A yellow arrow points from the variable declarations to the first iteration of the loop. A yellow bracket highlights the value of `*`.

stdout

*

Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3   i=1   *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



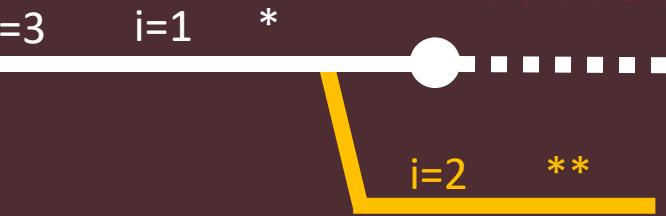
stdout

*

Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3   i=1   *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



The diagram illustrates the state of variables at the start of the loop iteration where *i* is 2. A horizontal bar represents the memory space. Above the bar, *n* is labeled as 3, *i* as 1, and '*' as the current character being processed. A yellow bracket below the bar indicates the range from *i*=1 to *i*=2, with the character '*' highlighted. A yellow arrow points to the character '*' in the code's assignment statement.

stdout

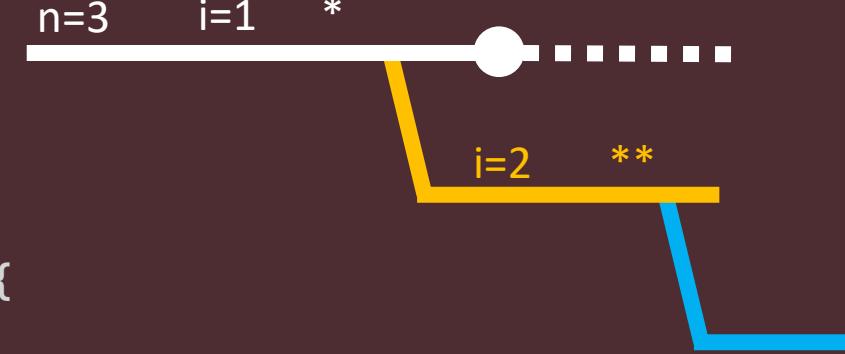
*

**

Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3   i=1   *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



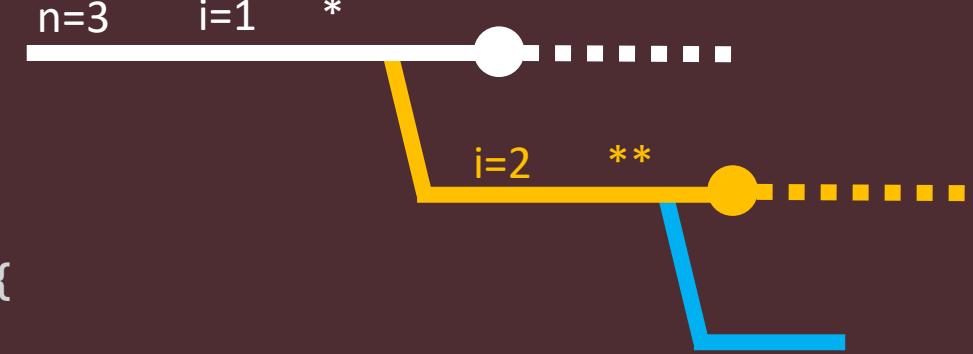
stdout

```
*  
**
```

Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3   i=1   *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



stdout

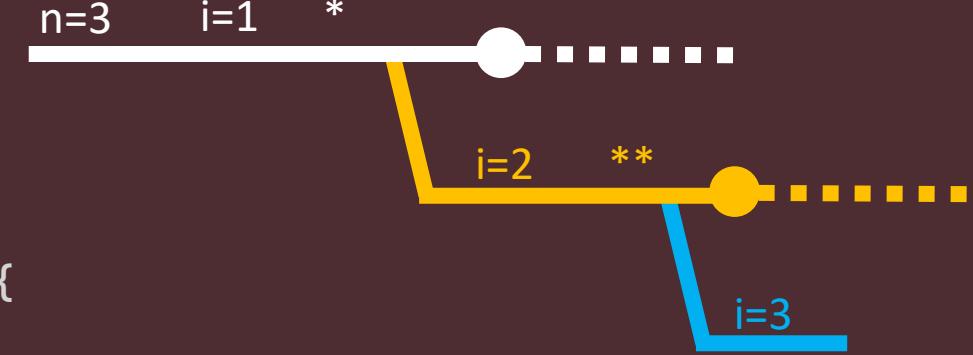
*

**

Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3 i=1 *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
  
        pid_t pid = fork();  
  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
  
            star(i);  
  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



stdout

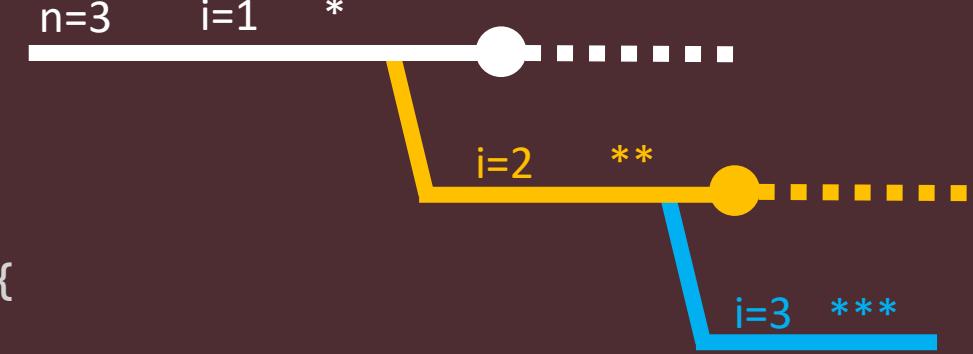
*

**

Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3   i=1   *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
  
        pid_t pid = fork();  
  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
  
            star(i);  
  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

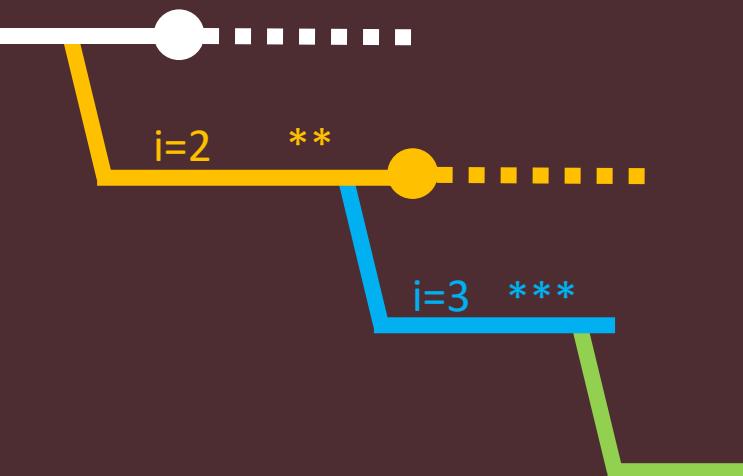


stdout
*
**

Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



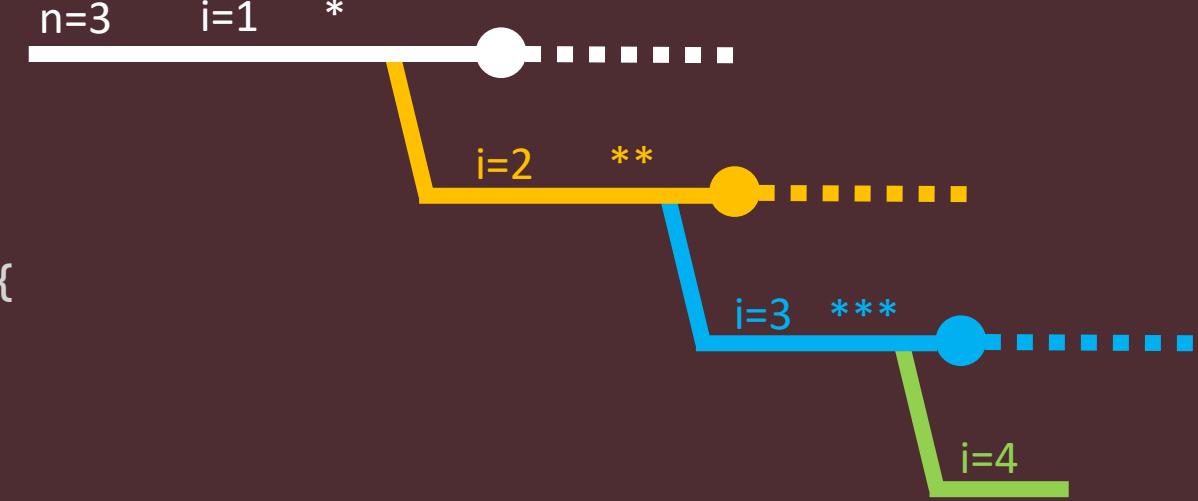
stdout

```
*  
**  
***
```

Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3 i=1 *
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(i);
        pid_t pid = fork();
        if (pid > 0) { // Parent
            waitpid(pid, NULL, 0);
            star(i);
            exit(EXIT_SUCCESS);
        }
    }
}
```

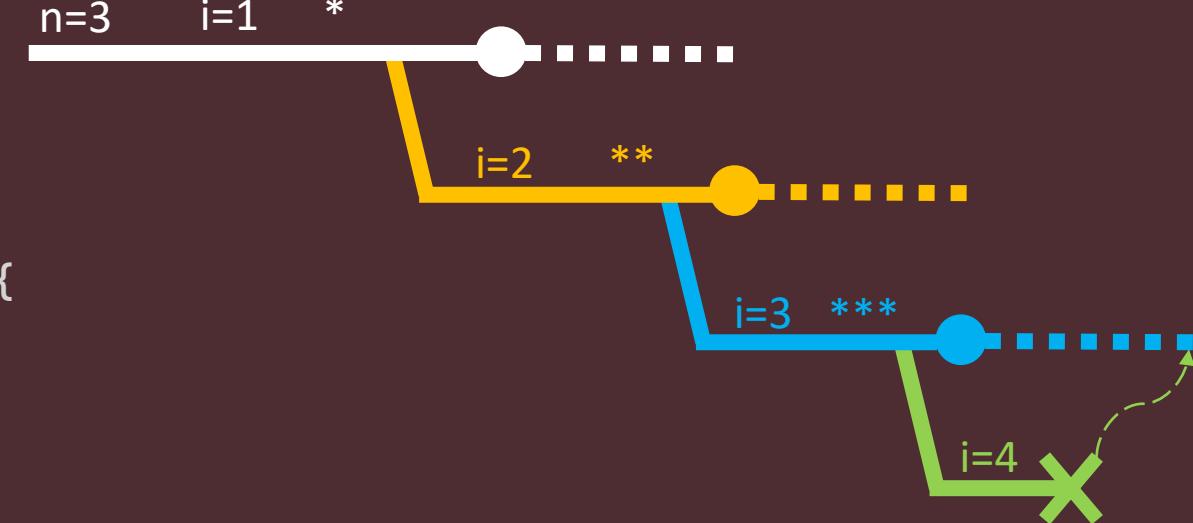


```
stdout
*
**
***
```

Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```



```
stdout  
*  
**  
***
```

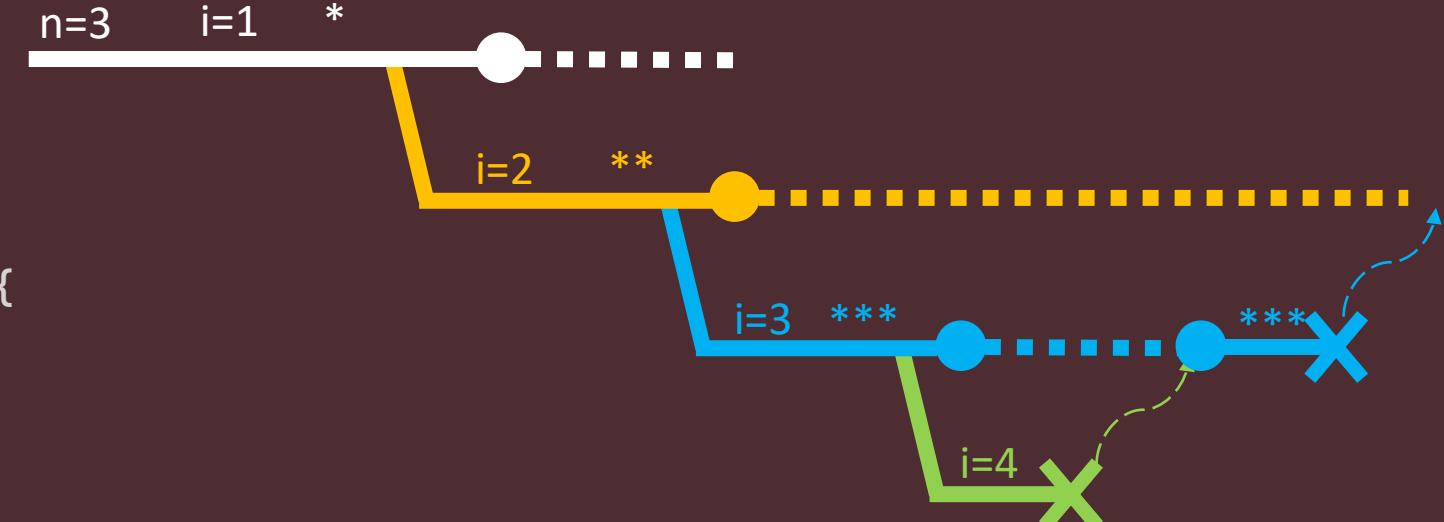
Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

stdout

```
*  
**  
***  
***
```



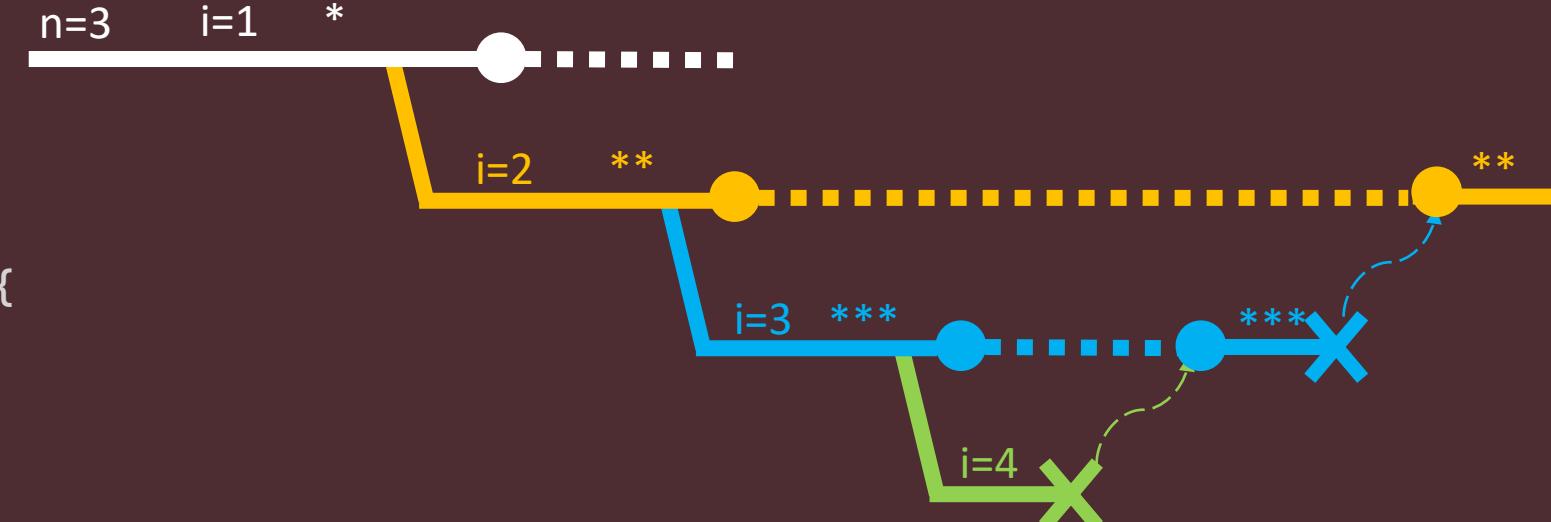
Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3 i=1 *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

stdout

```
*  
**  
***  
***  
**
```



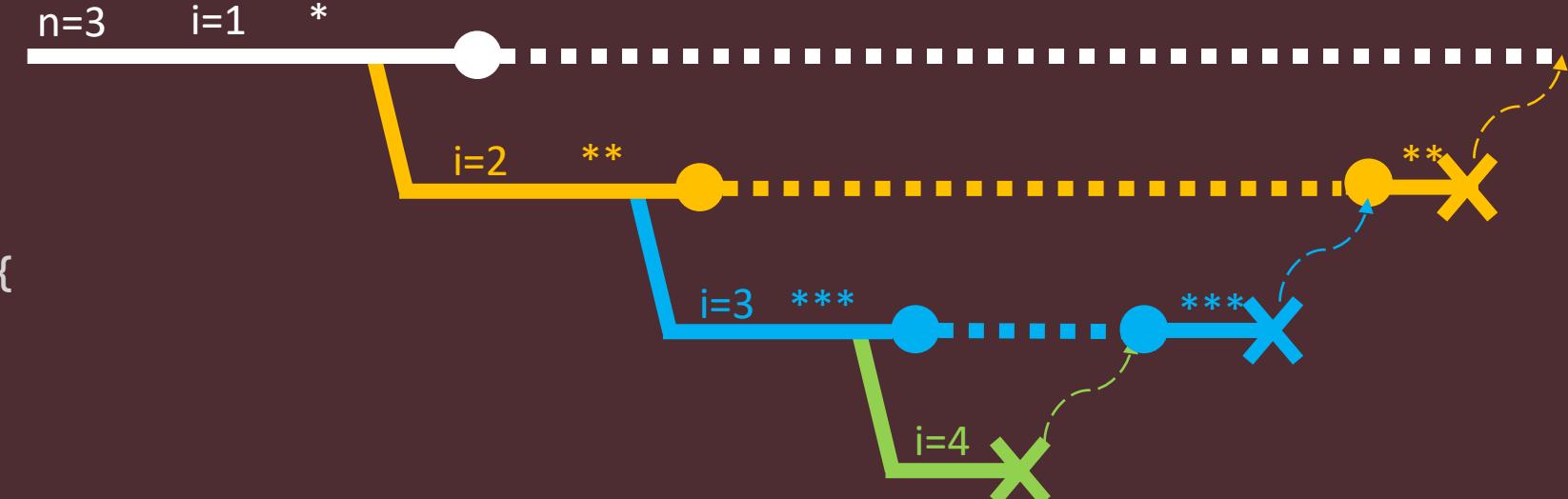
Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) { n=3    i=1    *  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        pid_t pid = fork();  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
            star(i);  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

stdout

```
*  
**  
***  
***  
**
```



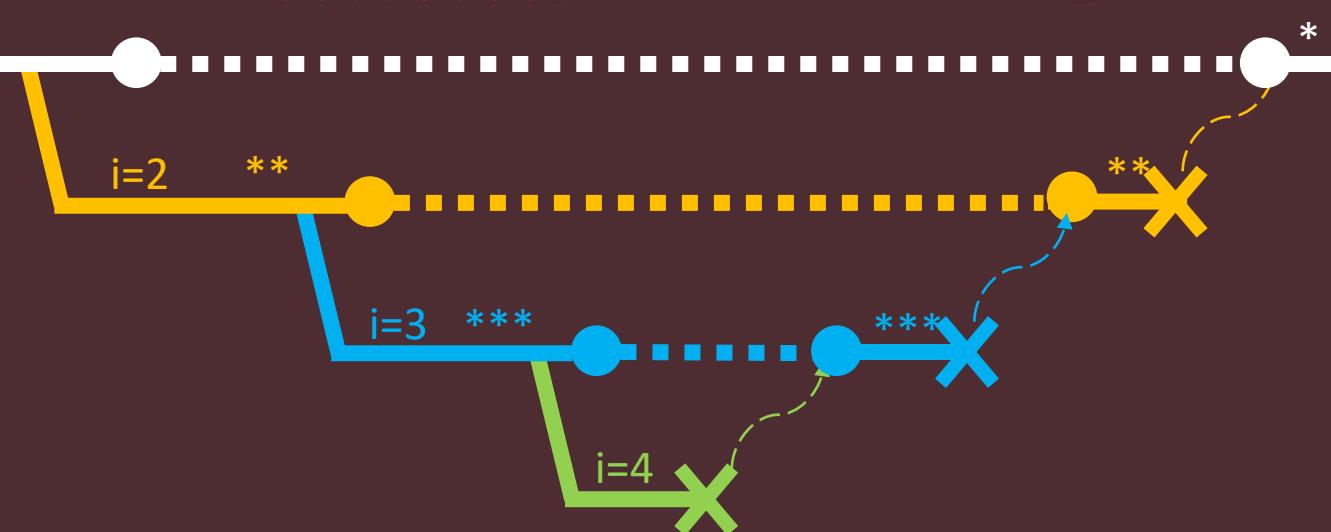
Part-5

\$./starfork-s5 3 2>/dev/null

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
  
        pid_t pid = fork();  
  
        if (pid > 0) { // Parent  
            waitpid(pid, NULL, 0);  
  
            star(i);  
  
            exit(EXIT_SUCCESS);  
        }  
    }  
}
```

stdout

```
*  
**  
***  
***  
**  
*
```



Part-6

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        sleep(1);  
        char *a[] = { argv[0], argv[1], NULL };  
        execv(*a, a);  
        printf("%s\n", "A STAR IS BORN");  
        exit(EXIT_SUCCESS);  
    }  
}
```

int execv (const char *path, char *const argv[]);
— Execute a file/program

What is the output when running with arguments 1, 2 or 3?
Is “A STAR IS BORN” ever printed?
Are any new processes ever created?

Part-6

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(i);
        sleep(1);
        char *a[] = { argv[0], argv[1], NULL };
        execv(*a, a);
        printf("%s\n", "A STAR IS BORN");
        exit(EXIT_SUCCESS);
    }
}
```

Part-6

\$./starfork-s6 1

n=1 i=1

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
  
    → for (int i = 1; i <= n; i++) {  
        star(i);  
        sleep(1);  
        char *a[] = { argv[0], argv[1], NULL };  
        execv(*a, a);  
        printf("%s\n", "A STAR IS BORN");  
        exit(EXIT_SUCCESS);  
    }  
}
```

stdout

Part-6

\$./starfork-s6 1

n=1 i=1 *

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        sleep(1);  
        char *a[] = { argv[0], argv[1], NULL };  
        execv(*a, a);  
        printf("%s\n", "A STAR IS BORN");  
        exit(EXIT_SUCCESS);  
    }  
}
```

stdout

*

Part-6

\$./starfork-s6 1

n=1 i=1 *
----- ?

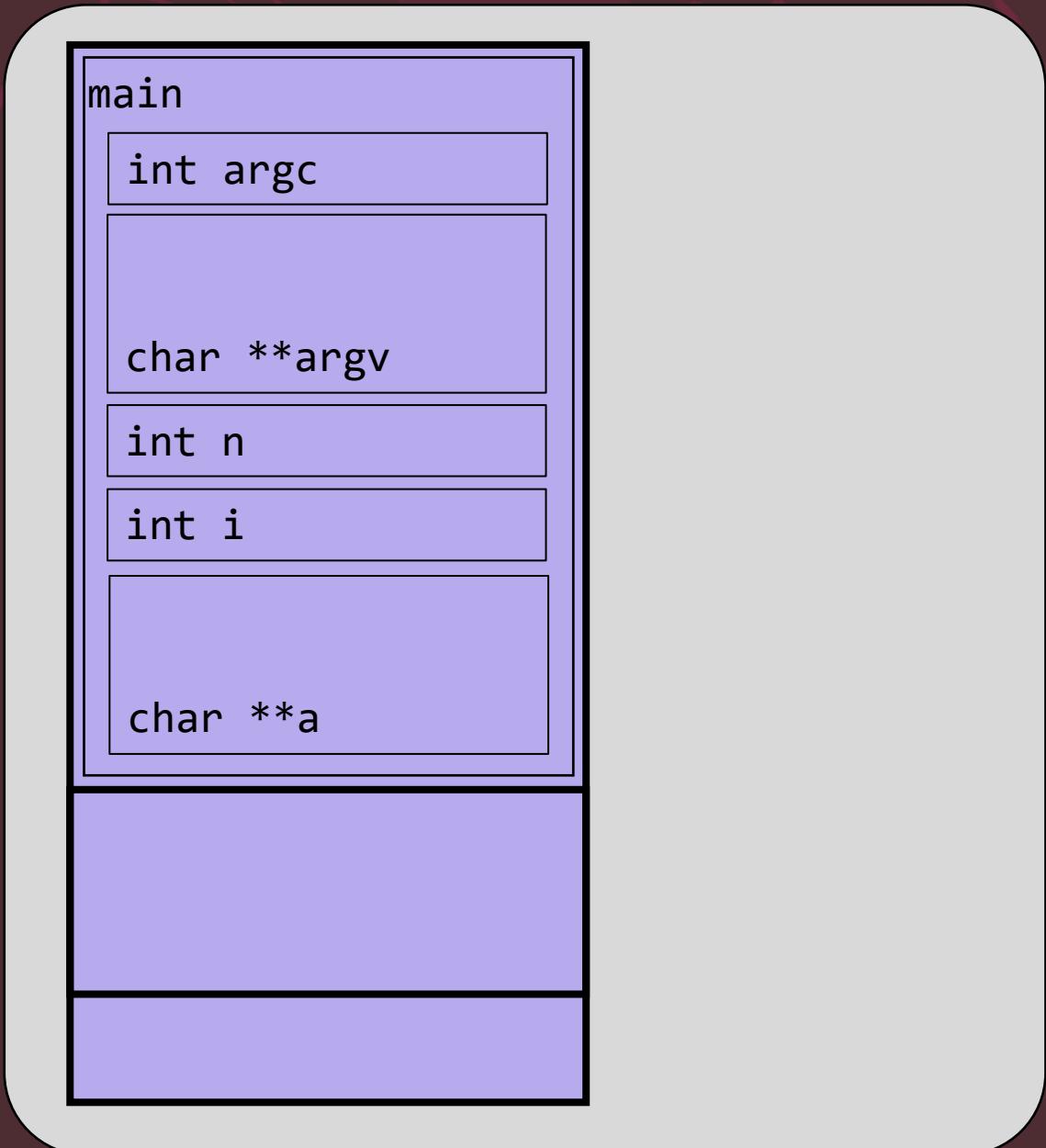
```
int main(int argc, char **argv) {  
  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
  
        star(i);  
  
        sleep(1);  
  
        char *a[] = { argv[0], argv[1], NULL };  
  
        execv(*a, a);  
  
        printf("%s\n", "A STAR IS BORN");  
  
        exit(EXIT_SUCCESS);  
  
    }  
}
```

stdout

*

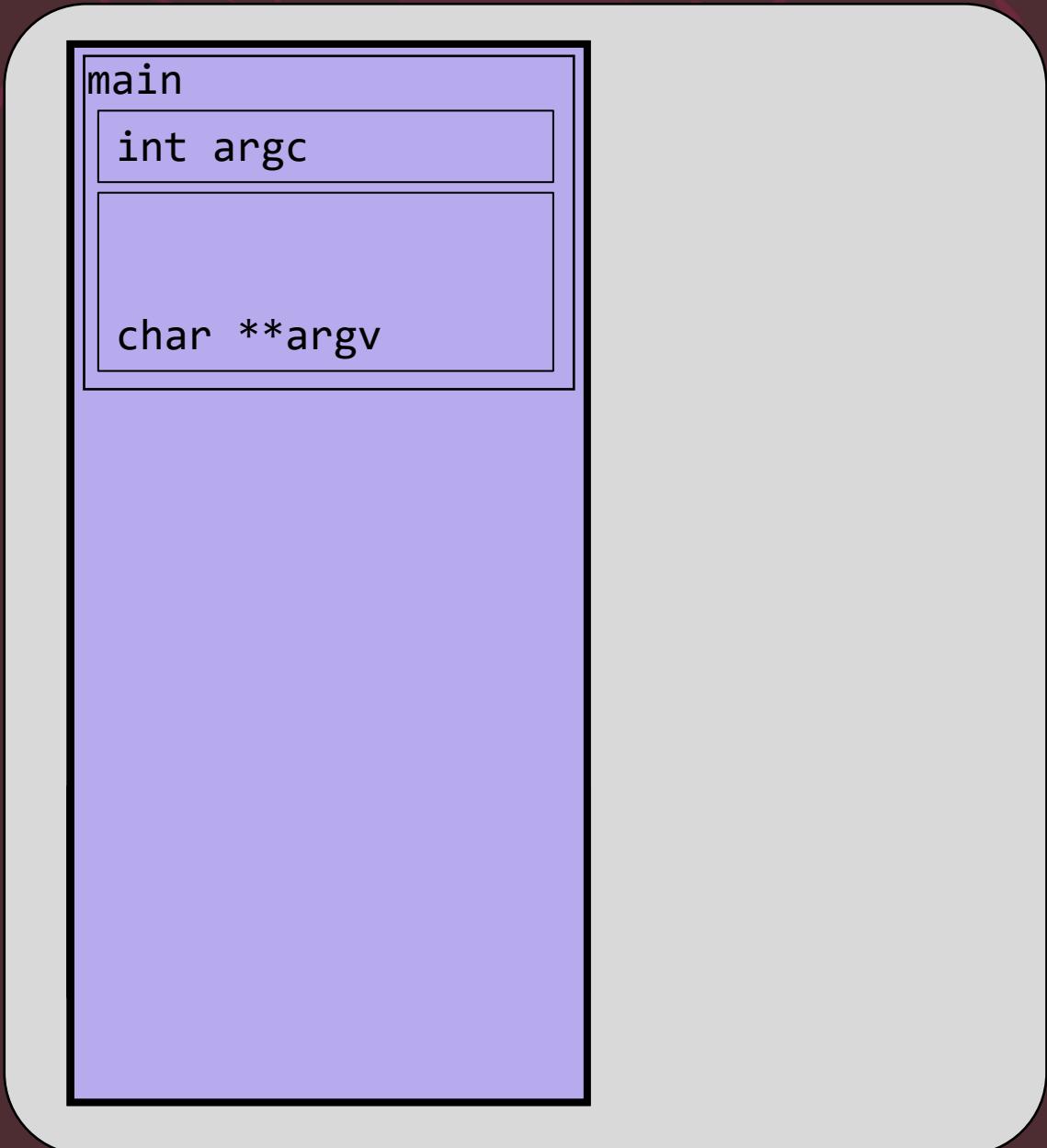
Part-6

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        sleep(1);  
        char *a[] = { argv[0], argv[1], NULL };  
        execv(*a, a);  
        printf("%s\n", "A STAR IS BORN");  
        exit(EXIT_SUCCESS);  
    }  
}
```



Part-6

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        sleep(1);  
        char *a[] = { argv[0], argv[1], NULL };  
        execv(*a, a);  
        printf("%s\n", "A STAR IS BORN");  
        exit(EXIT_SUCCESS);  
    }  
}
```



Part-6

\$./starfork-s6 1

n=1 i=1 *
----- ?

```
int main(int argc, char **argv) {  
  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
  
        star(i);  
  
        sleep(1);  
  
        char *a[] = { argv[0], argv[1], NULL };  
  
        execv(*a, a);  
  
        printf("%s\n", "A STAR IS BORN");  
  
        exit(EXIT_SUCCESS);  
  
    }  
}
```

stdout

*

Part-6

\$./starfork-s6 1

n=1 i=1 *

^ \$./starfork-s6 1

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        sleep(1);  
        char *a[] = { argv[0], argv[1], NULL };  
        execv(*a, a);  
        printf("%s\n", "A STAR IS BORN");  
        exit(EXIT_SUCCESS);  
    }  
}
```

stdout

*

Part-6

\$./starfork-s6 1

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
  
    for (int i = 1; i <= n; i++) {  
        star(i);  
        sleep(1);  
        char *a[] = { argv[0], argv[1], NULL };  
        execv(*a, a);  
        printf("%s\n", "A STAR IS BORN");  
        exit(EXIT_SUCCESS);  
    }  
}
```

n=1 i=1 * n=1 i=1 *
 ^ \$./starfork-s6 1 ...

stdout

*

*

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
  
        pid_t pid = fork();  
  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
  
        waitpid(pid, NULL, 0);  
    }  
    star(n);  
    exit(EXIT_SUCCESS);  
}
```

```
void star(int numstar) {  
    if (numstar >= 100)  
        exit(EXIT_FAILURE);  
    . . .  
}
```

int sprintf(char *str, const char *format, ...);
— Produce output according to a *format*
and write to the character string *str*.

What is the output when running with arguments 2, 10 or 50?

Part-7

\$./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {  
    → int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
  
        star(n);  
  
        pid_t pid = fork();  
  
        if (pid == 0) { // Child  
  
            char buf[100];  
  
            sprintf(buf, "%d", 2 * n);  
  
            char *a[] = { argv[0], buf, NULL };  
  
            execv(*a, a);  
  
        }  
  
        waitpid(pid, NULL, 0);  
  
        star(n);  
  
        exit(EXIT_SUCCESS);  
    }  
}
```

stdout

Part-7

```
$ ./starfork-s7 10 2>/dev/null
```

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

stdout

Part-7

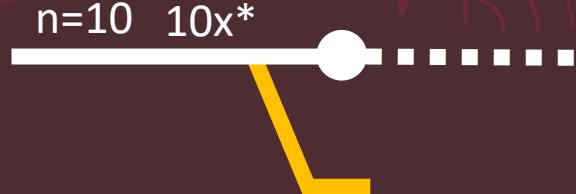
\$./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

stdout

Part-7

\$./starfork-s7 10 2>/dev/null



```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
  
        pid_t pid = fork();  
  
        if (pid == 0) { // Child  
  
            char buf[100];  
  
            sprintf(buf, "%d", 2 * n);  
  
            char *a[] = { argv[0], buf, NULL };  
  
            execv(*a, a);  
        }  
  
        waitpid(pid, NULL, 0);  
        star(n);  
  
        exit(EXIT_SUCCESS);  
    }
```

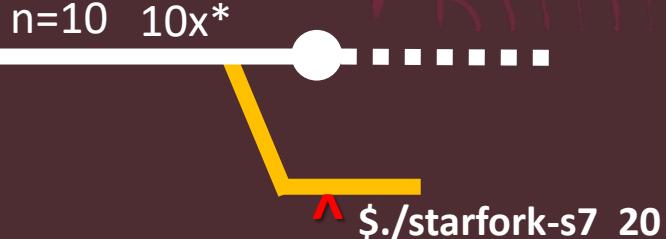


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

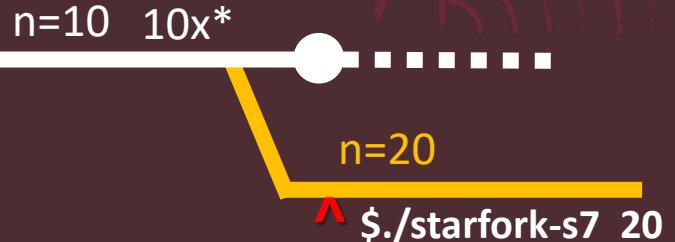


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

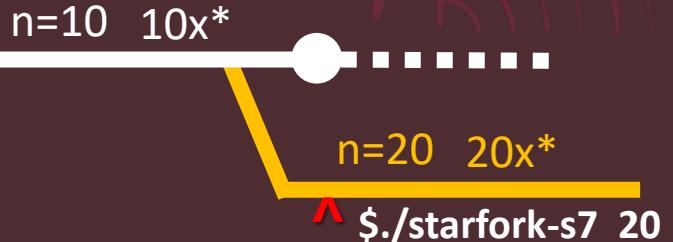


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

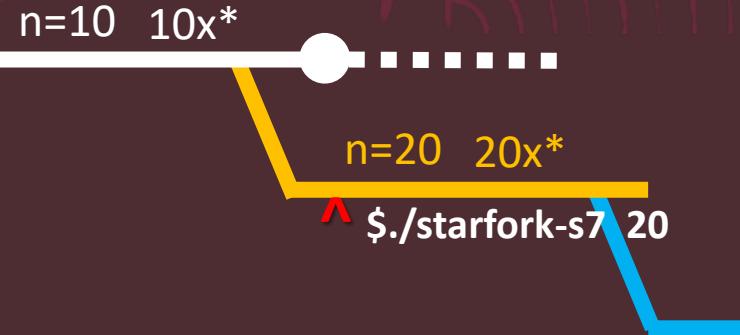


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

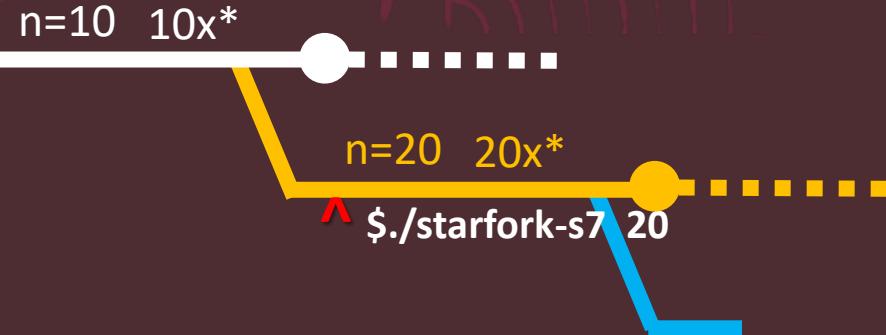


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

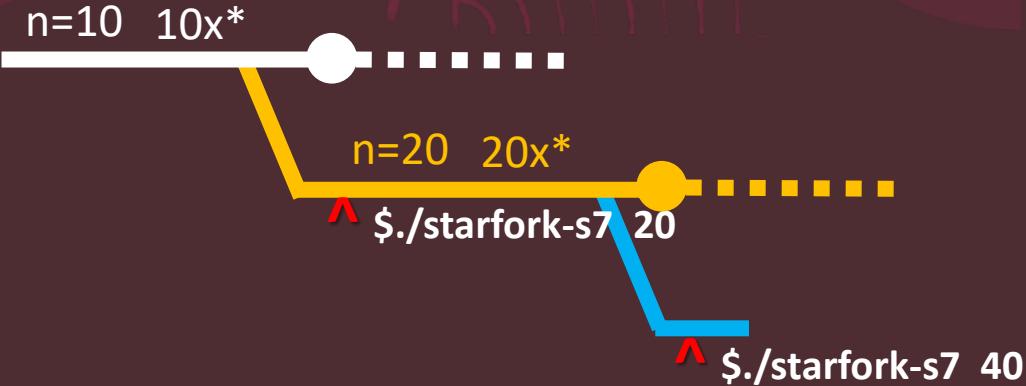


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

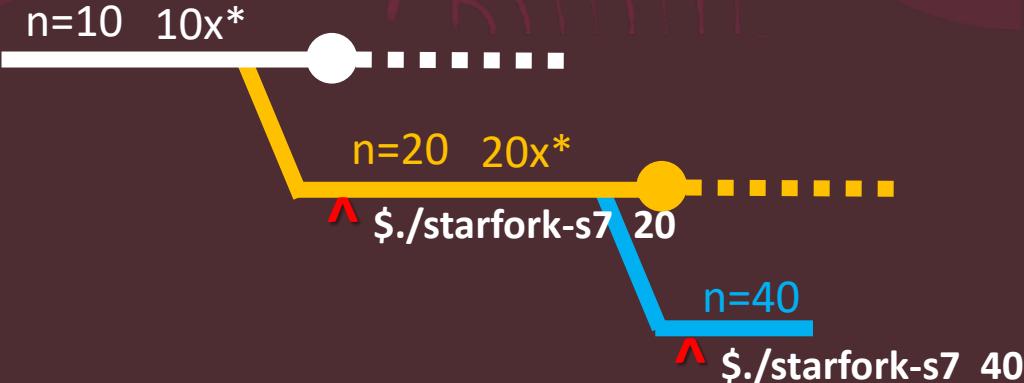


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

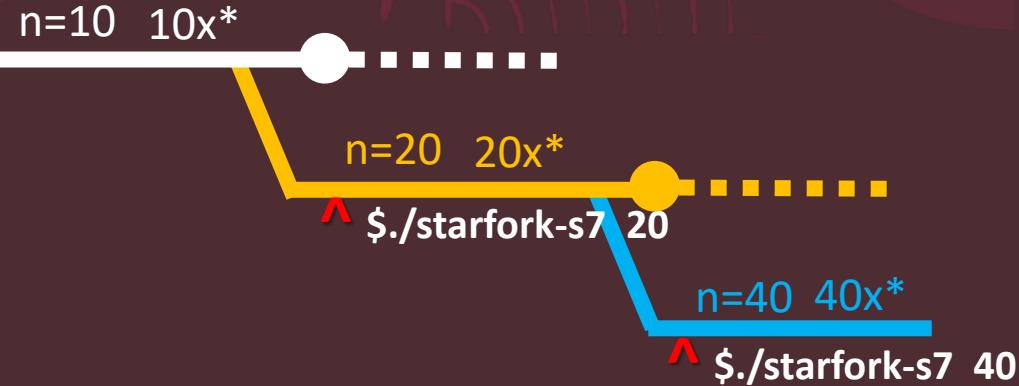


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

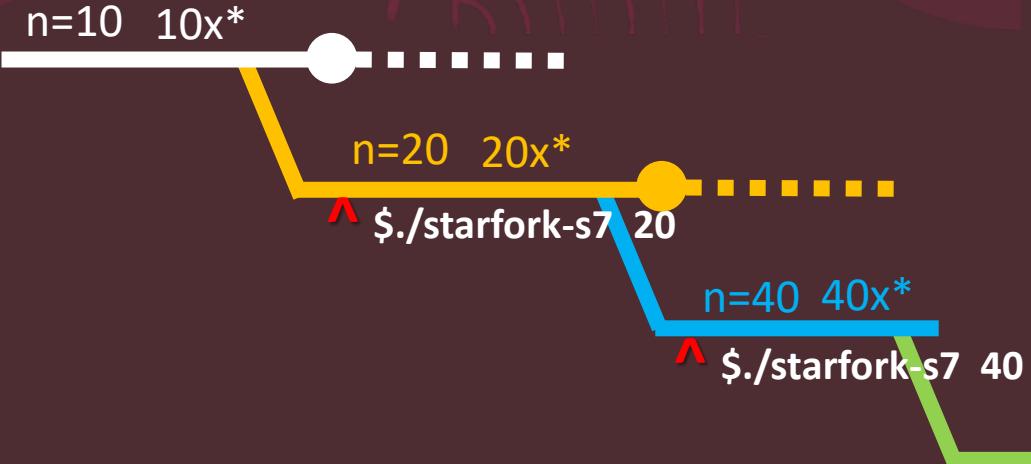


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

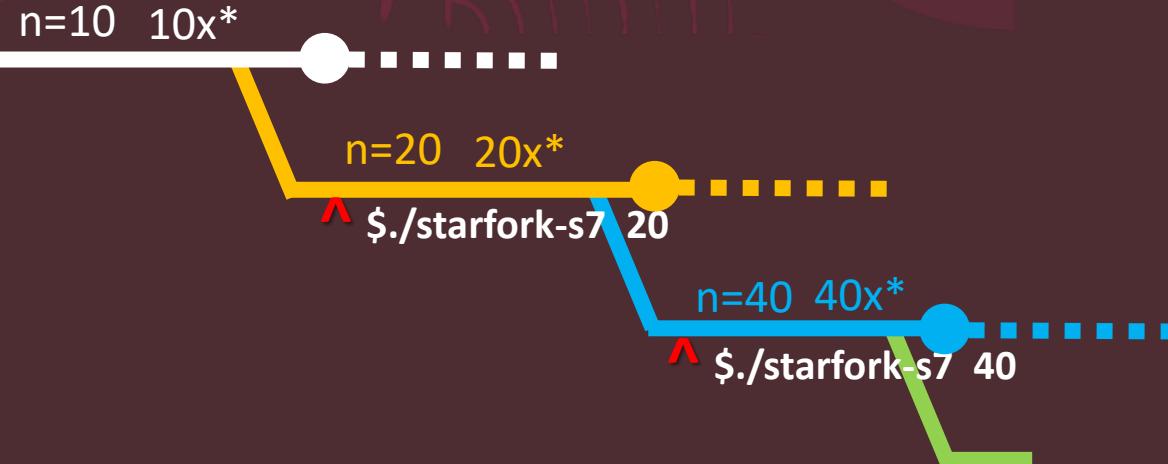


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

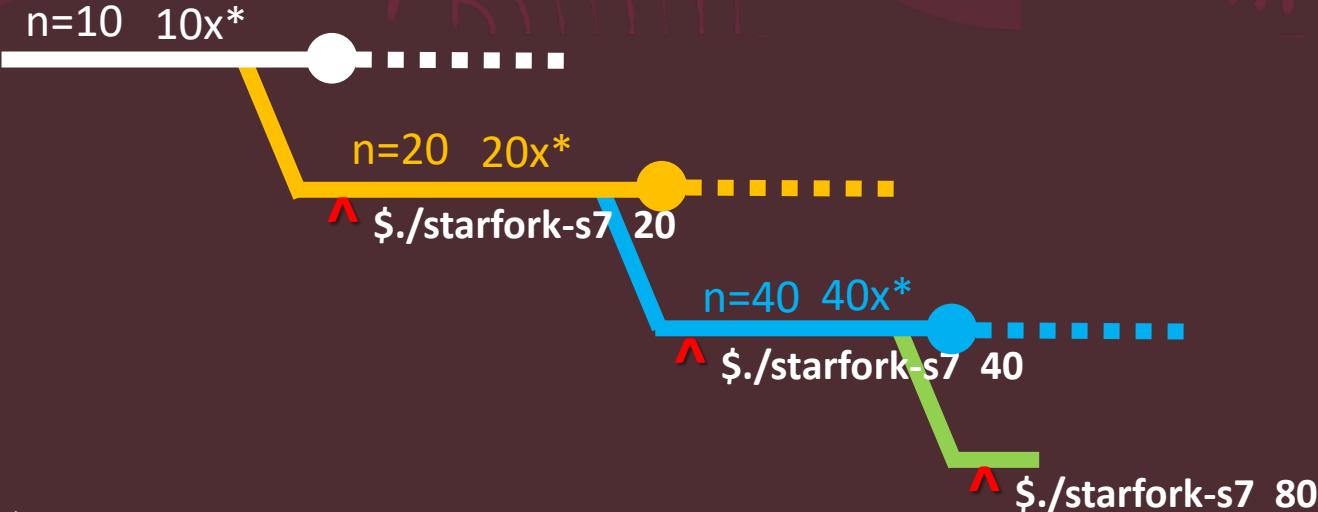


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

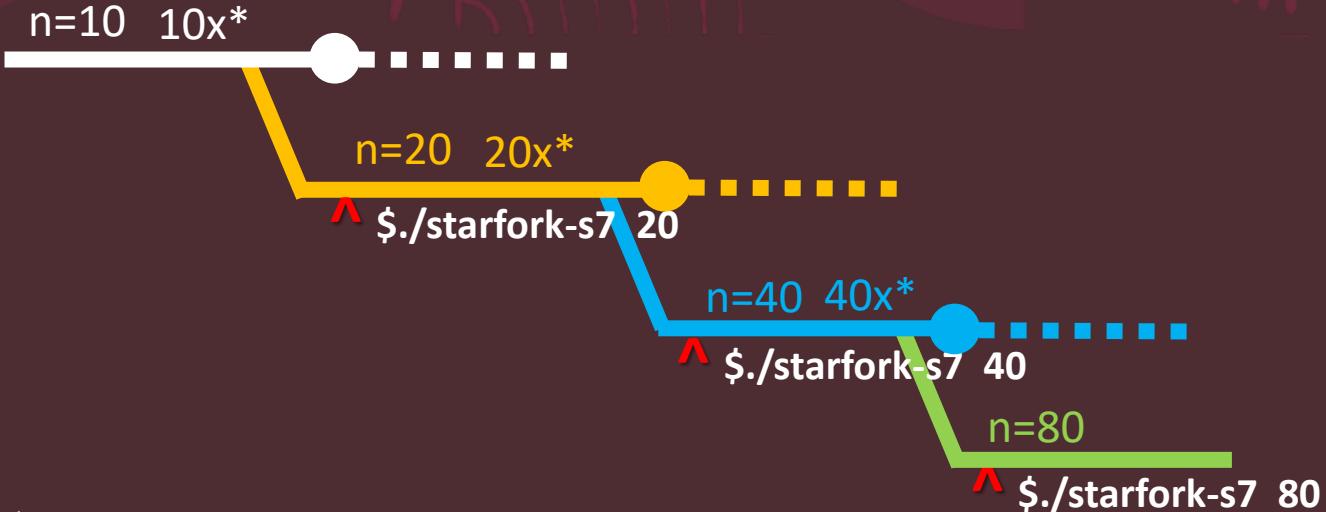


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
  
        pid_t pid = fork();  
  
        if (pid == 0) { // Child  
            char buf[100];  
  
            sprintf(buf, "%d", 2 * n);  
  
            char *a[] = { argv[0], buf, NULL };  
  
            execv(*a, a);  
        }  
  
        waitpid(pid, NULL, 0);  
        star(n);  
  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

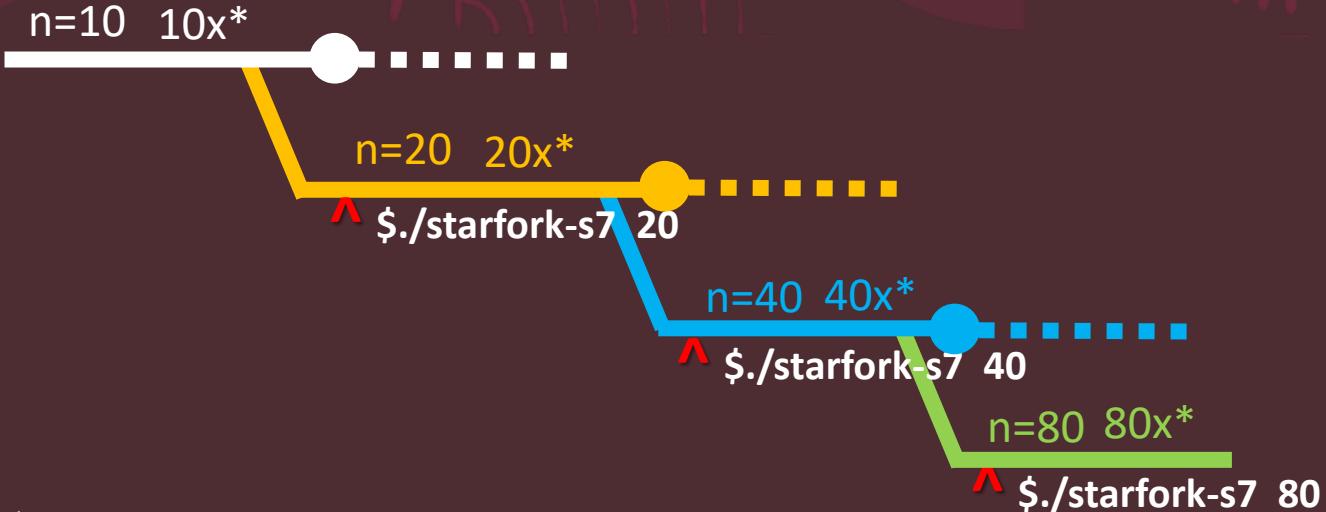


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

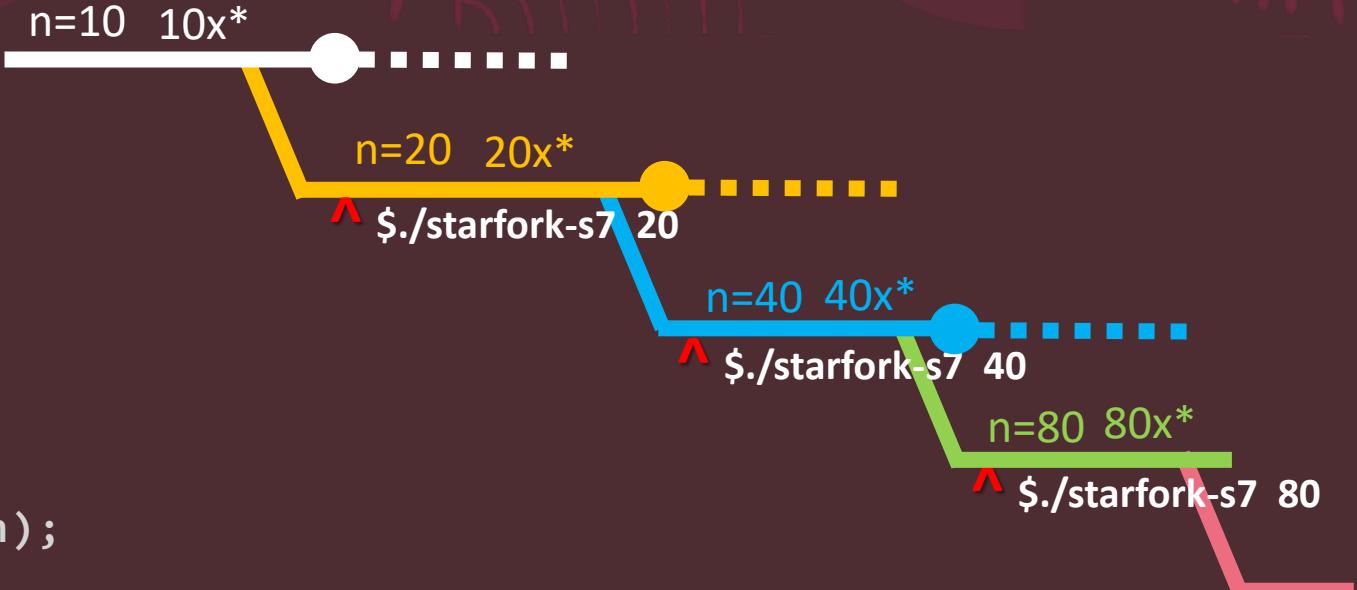


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

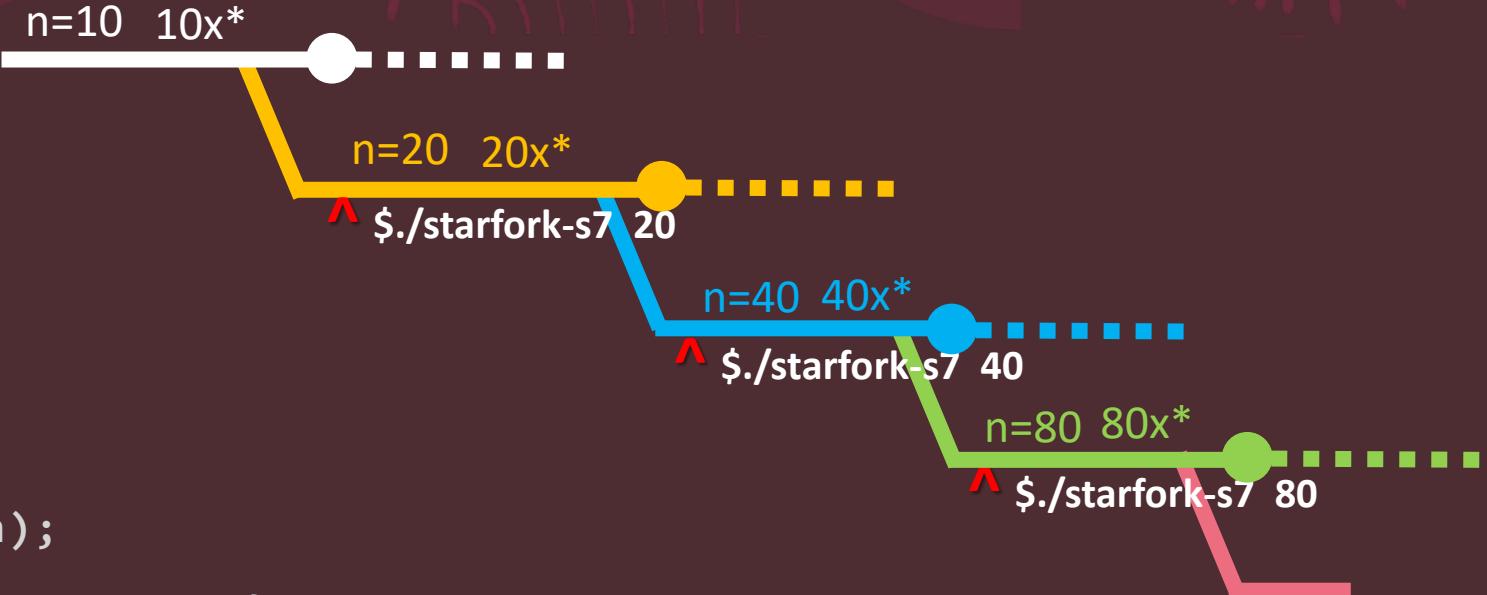


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

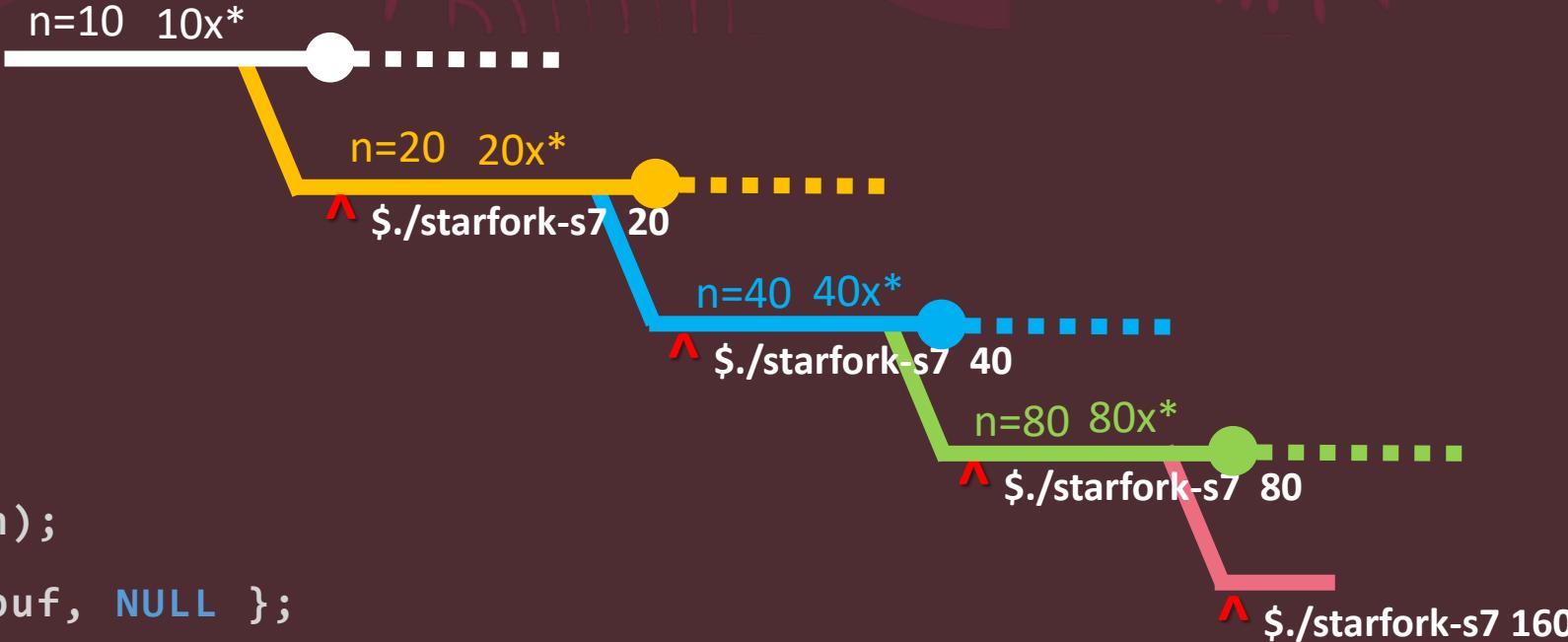


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

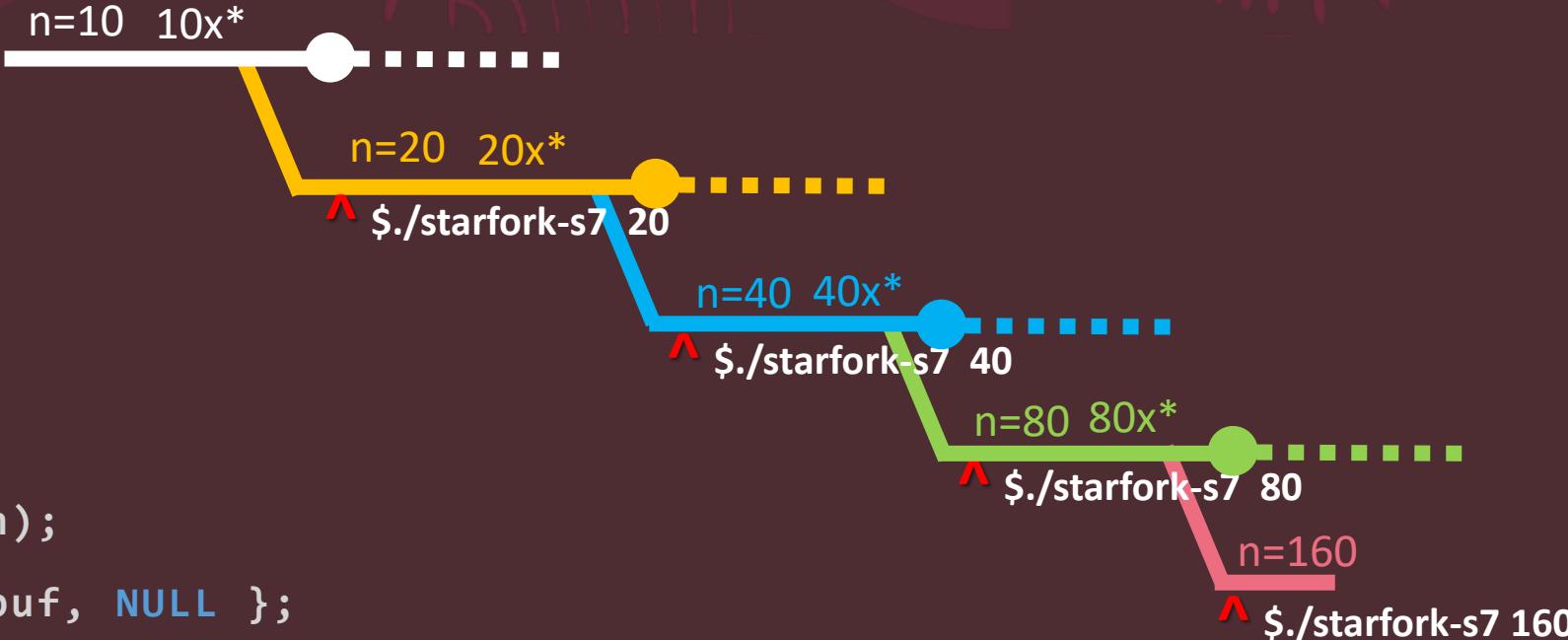


stdout

Part-7

```
int main(int argc, char **argv) {  
    → int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
  
        pid_t pid = fork();  
  
        if (pid == 0) { // Child  
            char buf[100];  
  
            sprintf(buf, "%d", 2 * n);  
  
            char *a[] = { argv[0], buf, NULL };  
  
            execv(*a, a);  
        }  
  
        waitpid(pid, NULL, 0);  
  
        star(n);  
  
        exit(EXIT_SUCCESS);  
    }
```

\$./starfork-s7 10 2>/dev/null

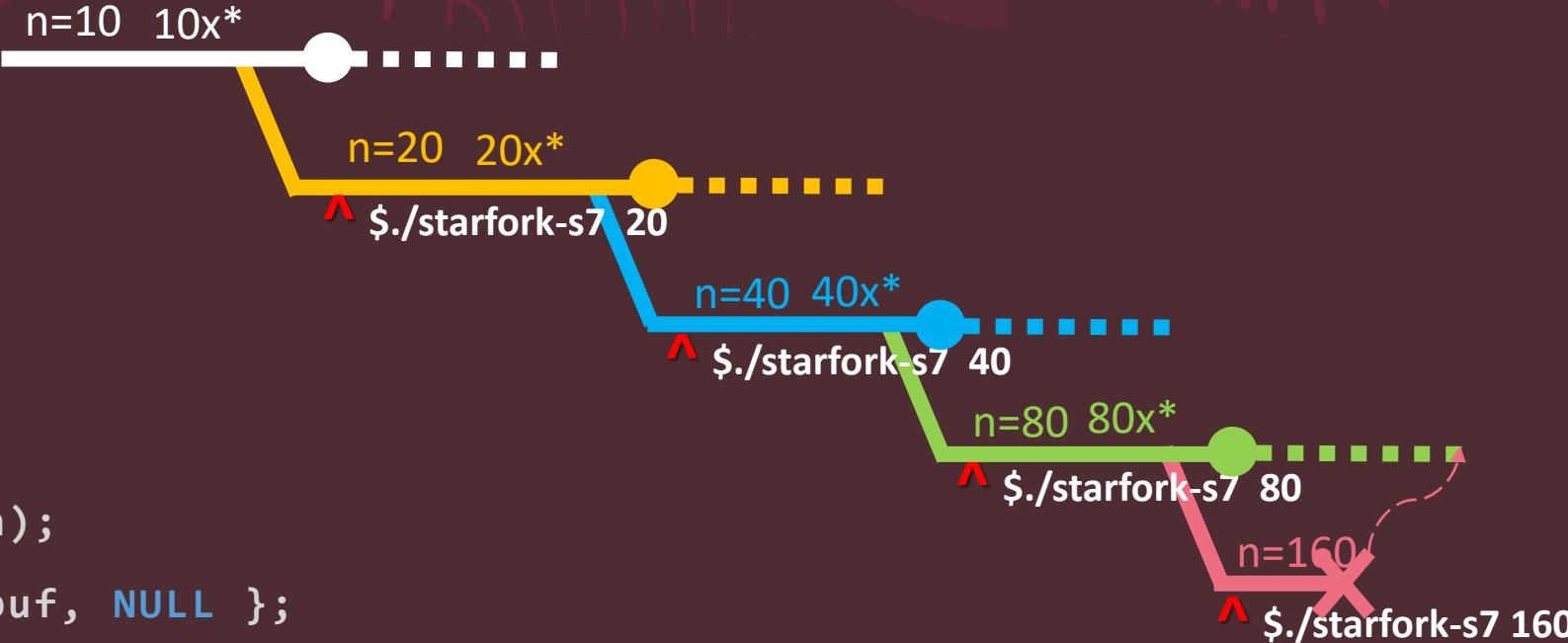


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
  
        void star(int numstar) {  
            if (numstar >= 100) child  
                exit(EXIT_FAILURE);  
            . . .  
            printf("%c", ' ', 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null

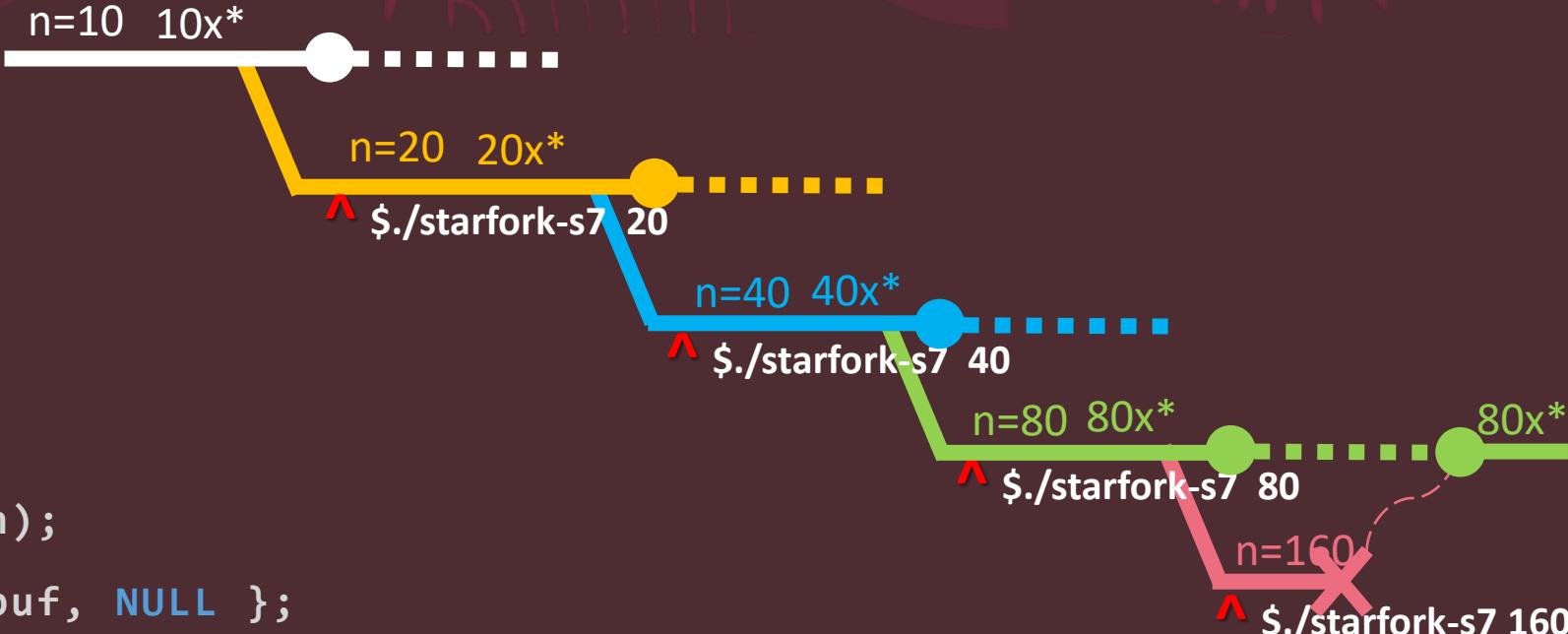


stdout

Part-7

```
int main(int argc, char **argv) {  
    int n = atoi(argv[1]);  
    for (int i = 1; i <= n; i++) {  
        star(n);  
        pid_t pid = fork();  
        if (pid == 0) { // Child  
            char buf[100];  
            sprintf(buf, "%d", 2 * n);  
            char *a[] = { argv[0], buf, NULL };  
            execv(*a, a);  
        }  
        waitpid(pid, NULL, 0);  
        star(n);  
        exit(EXIT_SUCCESS);  
    }  
}
```

\$./starfork-s7 10 2>/dev/null



stdout

Part-7

\$./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {
    n=10 10x*
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        n=20 20x*
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```

stdout



Part-7

\$./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {
    n=10 10x*
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        n=20 20x*
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```

stdout

```
*****
*****
*****
*****
```

Part-7

\$./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {
    n=10 10x*
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        n=20 20x*
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```

stdout

```
*****
*****
*****
*****
```

Part-7

\$./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {
    n=10 10x*
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```

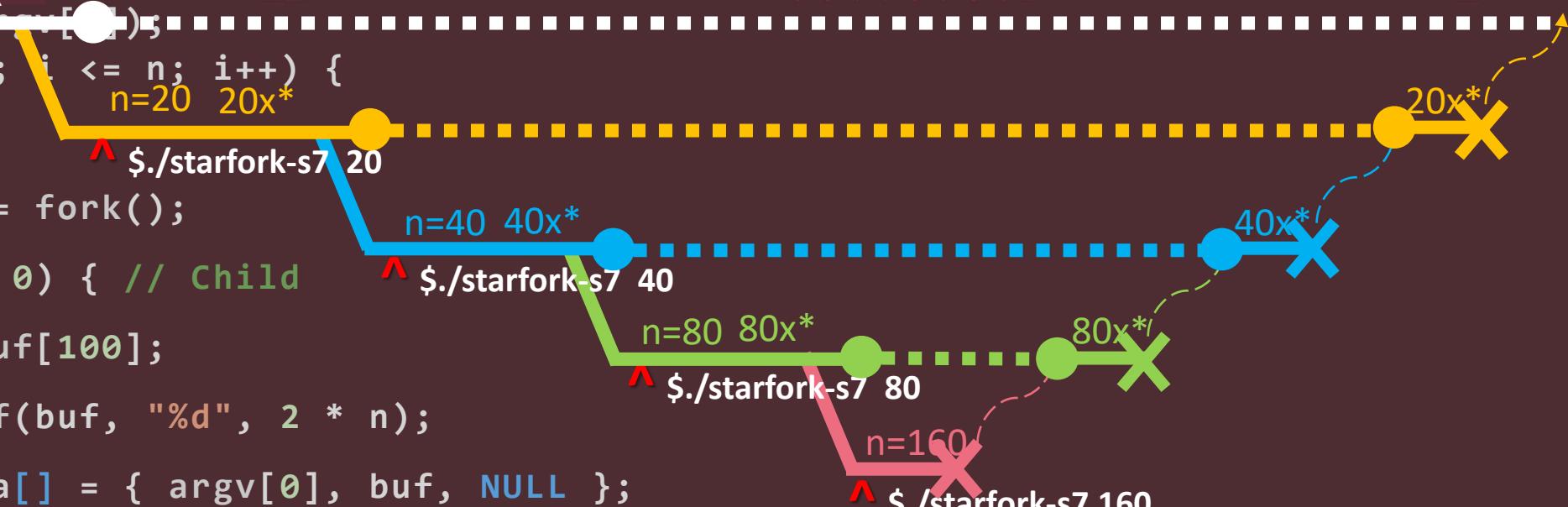
stdout

```
*****
*****  
*****  
*****  
*****
```

Part-7

\$./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {
    n=10 10x*
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



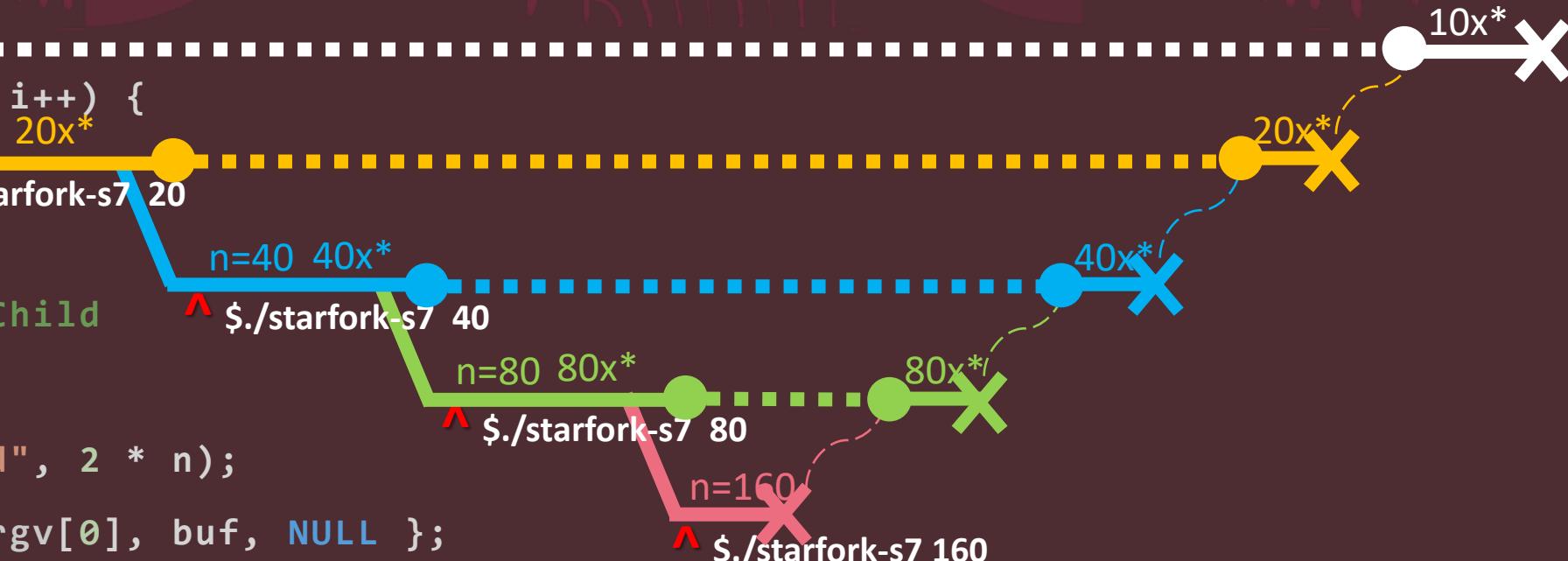
stdout

```
*****
*****  
*****  
*****  
*****
```

Part-7

\$./starfork-s7 10 2>/dev/null

```
int main(int argc, char **argv) {
    int n = atoi(argv[1]);
    for (int i = 1; i <= n; i++) {
        star(n);
        pid_t pid = fork();
        if (pid == 0) { // Child
            char buf[100];
            sprintf(buf, "%d", 2 * n);
            char *a[] = { argv[0], buf, NULL };
            execv(*a, a);
        }
        waitpid(pid, NULL, 0);
        star(n);
        exit(EXIT_SUCCESS);
    }
}
```



stdout

```
*****
*****  
*****  
*****  
*****
```

Part-7

\$./starfork-s7 10 2>/dev/null

