



# Proiect modul Algoritmica I

`get_next_line`

Staff Academy+Plus [contact@academyplus.ro](mailto:contact@academyplus.ro)

*Sumar: Acest proiect are ca obiectiv sa scrieti o functie ce returneaza o linie ce se termina cu un retur de linie de la un descriptor de fisier.*

# Cuprins

<b>I</b>	<b>Preambul</b>	<b>2</b>
<b>II</b>	<b>Subiect</b>	<b>4</b>
II.1	Parte obligatorie . . . . .	4
II.2	Livrare . . . . .	5
II.3	Considerente tehnice . . . . .	6
II.4	Functii autorizate . . . . .	8
II.5	Bonus . . . . .	8
<b>III</b>	<b>Instructiuni</b>	<b>9</b>
<b>IV</b>	<b>Notare</b>	<b>10</b>

# Capitolul I

## Preambul

James Deen, pe numele sau Bryan Matthew Seville, nascut in data de 7 februarie 1986, este un

actor america de filme pronografice. De asemenea director de productie, el si-a inceput devreme

cariera in industria porno, in 2004, la varsta de 18 ani.

James Deen s-a nascut in Los Angeles si a crescut in Pasadena. Crescut intr-o cultura evreiasca,

acesta afirma: „Ma identific in primul rand cu cultura iudaica. Nu as putea sa explic, dar de indata

ce intalnesc un evreu ma simt legat de el”.

James spune ca a vazut filme porno pentru prima data in scoala primara, si ca in acel moment i- a venit ideea de a face participa intr-un film. Devenit adolescent, el o alsculta pe Jenna Jameson,

un start de cinema porno, la radio (Loveline, emisiunea animata de aceasta), si cere informatii

despre cum sa devina un star chiar el. Aceasta i-a raspuns pur si simplu „sa se masturbeaza

in fata a douazeci dintre cei mai buni prieteni". Tanarul ia in serios sfatul si il pune in aplicare

in urmatoarele serate liceene. Astfel, el a luat numele de James Deen, care, precum celebrul

homophone, apare intunecat sumbru intr-o geaca de piele si fumand o tigara.

Pe baza acestor experimente initiale, James a decis sa intalneasca toate tinerele care au legaturi

cu industria porno si pe care le poate contacta; James este prezentat Pamelei Peaks, care ii

propune diverse nume de scene, inclusiv Clint Cullingus. Deen, care a uimit multi producatorii ani

de filme porno, a trebuit totusi sa isi dovedeasca talentele avand in vedere varsta sa tanara de si

lipsa de experienta pe ecran. Intr-un singur an de cariera, James a filmat 600 de filme, ceea ce

este un record.

In 2010, Deen a aparut in aproape 1000 de filme pornografice si atribuie succesul sau aspectului sau different fata de alti actori culturisti, imbracat de tatuaje si atragator din punct de vedere sexual.

James Deen este imaginea The Canyons 2013 al lui Paul Schrader. El nu a vazut nici o diferenta majora intre modul in care el a jucat la Hollywood si in filme porno, „sex mai putin”.



Figura I.1: James Deen

# Capitolul II

## Subiect

### II.1 Parte obligatorie

- Scrieti o functie ce returneaza o linie luata dintr-un descriptor de fisier.
- Numim “linie” o succesiune de caractere terminate printr-un ‘\n’ (cod ascii 0x0a) sau printr-un End Of File (EOF).



<https://latedev.wordpress.com/2012/12/04/all-about-eof/>: acest articol este relevant pentru cei ce stiu sa citeasca.

- Functia voastra va trebui sa aiba prototipul urmator:

```
int  get_next_line(int const fd, char ** line);
```

- Primul parametru este descriptorul de fisier din care se citeste.
- Al doilea parametru este adresa unui pointer spre caracterul ce serveste la stocarea linia luata din descriptorul de fisier.
- Valoarea returnata poate fi 1, 0 sau -1 daca o linie a fost citita, citirea s-a terminat, sau s-a produs o eroare.
- Functia `get_next_line` trebuie sa returneze rezultatul fara ‘\n’.
- Un apel in bucla a functiei `get_next_line` va permite deci, citirea textului disponibil in descriptorul de fisier, linie cu linie, pana la sfarsitul textului.
- Asigurati-va ca functia se comporta bine cand citeste dintr-un fisier, de la intrarea standard, dintr-o redirectare etc.

## II.2 Livrare

- Trebuie sa livrati in radacina directorului vostru de lucru/livrare, un fisier `auteur` continand login-ul vostru urmat de `'\n'`:

```
$>cat -e auteur  
xlogin$  
$>
```

- Trebuie sa livrati doua fisiere: `get_next_line.c` si `get_next_line.h`
- Nu trebuie sa aveti nici o functie `main` in directorul vostru de lucru/livrare.
- Nu livrati fisier `Makefile`.
- Continutul prezent in directorul vostru de lucru/livrare va fi evaluat la sustinere.

## II.3 Considerente tehnice

- Fisierul vostru `get_next_line.h` trebuie cel puțin să conțină prototipul funcției `get_next_line` și un macro ce permite dimensiunea bufferului de citire la fiecare apel `read`. Aceasta valoare va fi modificată la susținere pentru evaluarea calității livrării voastre. De exemplu:

```
#define BUFF_SIZE 32
```



Codul vostru funcționează întotdeauna când `BUFF_SIZE` ia valoarea 9999? Și când `BUFF_SIZE` ia valoarea 1? Și 10000000? Știți de ce?

- Se consideră că funcția `get_next_line` are un comportament nedeterminat între două apeluri, când un același descriptor de fișier desemnează două fișiere diferite în momentul în care citirea primului fișier nu e terminată.
- De asemenea, considerăm că un apel la funcția `lseek(2)` nu va avea loc niciodată între două apeluri ale `get_next_line` pe același descriptor de fișier.
- În fine, considerăm că `get_next_line` are un comportament nedeterminat în cazul în care se citește dintr-un fișier binar. Astfel, dacă doriți, puteți adăuga această funcționalitate.
- Este interzis să folosiți variabile globale pentru salvarea caracterelor ce au fost citite, dar care nu au fost returnate de funcție. Puteți folosi variabile statice.



Dacă știți ce este o variabilă statică este un bun început:  
[https://en.wikipedia.org/wiki/Static\\_variable](https://en.wikipedia.org/wiki/Static_variable)

- Dacă sunteți isteti și folosiți biblioteca `libft` pentru funcția `get_next_line`, trebuie să copiați sursele și `Makefile`-ul asociat într-un director numit `libft` ce va trebui să fie în rădăcina directorului vostru de lucru/livrare. La susținere, cel care va corectea va compila livrarea în felul următor (fișierul `main.c` va fi al celui care va corectea):

```
$>make -C libft/ fclean
$>make -C libft/
$>gcc -Wall -Wextra -Werror -I libft/includes/ -c get_next_line.c
$>gcc -Wall -Wextra -Werror -I libft/includes/ -c main.c
$>gcc -o test_gnl get_next_line.o main.o -L libft/ -lft
```

- Dacă dimpotriva, nu utilizați biblioteca `libft` la funcția `get_next_line`, cel care va corectea va compila livrarea în felul următor, la susținere (fișierul `main.c` va fi al celui ce va corectea):

```
$>gcc -Wall -Wextra -Werror -c get_next_line.c  
$>gcc -Wall -Wextra -Werror -c main.c  
$>gcc -o test_gnl get_next_line.o main.o
```



## II.4 Functii autorizate

- read
- malloc
- free

## II.5 Bonus

Proiectul `get_next_line` este simplu si lasa putine optiuni pentru adaugarea de bonu-suri, dar sunt convins ca aveti multa imaginatie. Daca veti reusi sa realizati perfect partea obligatorie, aceasta sectiune va propune cateva piste pentru a merge mai dearte. Atentie, niciun bonus nu va fi validat daca la partea obligatorie nu obtineti cel putin 18 din cele 20 puncte.

- Sa realizati functia `get_next_line` cu o singura variabila statica.
- Puteti folosi mai multi descriptori de fisier pentru `get_next_line`. De exemplu, daca descriptorii de fisier 3, 4 si 5 sunt accesibili la citire, atunci se poate apela `get_next_line` odata pentru 3, odata pentru 4 si din nou pentru 3, apoi odata pentru 5 etc, fara a pierde firul citirii pe fiecare din descriptori.

# Capitolul III

## Instructiuni

- Acest proiect nu va fi corectat doar de fiinte umane.
- Proiectul trebuie sa fie scris conform standardului de cod (Norme). Norminette va fi utilizata pentru verificarea Normei, care se aplica in ansamblu.
- Va trebui sa tratati erorile intro maniera adecvata. In niciun caz programul nu trebuie sa se opreasca intr-un mod neasteptat (Eroare de segmentare, eroare de magistrala etc).
- Toata memoria alocata va trebui eliberata in mod adecvat cand este necesar.
- Trebuie sa livrati in radacina directorului vostru de lucru/livrare, un fisier `auteur` continand login-ul vostru urmat de `'\n'`:

```
$>cat -e auteur  
xlogin$  
$>
```

- Daca decideti sa livrati acest proiect folosind biblioteca `libft`, va este formal interzis sa adaugati functii specifice la livrarea functiei `get_next_line` ca sa evitati constrangerile impuse de standardul de cod (Norme). Aceasta va fi considerata o tentativa de trisare la sustinere. Functia voastra `get_next_line` trebuie sa fie compusa din cel mult 5 functii de maxim 25 de linii. Respectarea acestei constrangeri va fi in mod minutios verificat la sustinere. Este inutil sa incercati sa convingeti membri staff-ului daca o anumita functie ar fi acceptabila si pe care vreti sa o adaugati bibliotecii voastre. Intrebati-va mai bine daca functia voastra incalca sau nu aceste constrangeri dupa constiinta voastra. Daca veti respecta aceasta regula veti fi incurajati sa va extindeti biblioteca cu functii generice pe a caror utilitate ati descoperit-o pe parcursul acestui proiect.

# Capitolul IV

## Notare

Notarea functiei `get_next_line` se face in doi timpi:

- In primul rand se va testa partea obligatorie. Aceasta va fi notata cu maxim 20 puncte.
- Apoi, vor fi evaluate bonusurile. Ele vor fi notate cu maxim 5 puncte.
- Bonusul va fi evaluat doar daca la partea obligatorie ati obtinut cel putin 18 din 20 puncte.

Succes tuturor!