

# Προγραμματισμός Ι

## Εργαστήριο 1

Διδάσκων: Χρήστος Δίου

### 1 Τρόπος διεξαγωγής εργαστηρίων - προετοιμασία

Τα εργαστήρια του μαθήματος θα διεξάγονται στο εργαστήριο υπολογιστών του 4ου ορόφου. Ξεκινήστε τον υπολογιστή (αν χρειάζεται) και στον login manager επιλέξτε το περιβάλλον "Ubuntu Wayland". Εισάγετε το username σας (της μορφής it202XXXX) και τον κωδικό σας.

Αυτό θα σας εμφανίσει το γραφικό περιβάλλον Gnome. Πατώντας Alt+Ctrl+T ταυτόχρονα, ανοίγει ένα τερματικό (κονσόλα) με το κέλυφος (shell) bash (bourne again shell). Το bash και η γραμμή εντολών θα είναι η βασική μας διεπαφή με το λειτουργικό σύστημα στο μάθημα.

#### 1.1 Βασικές εντολές του Bash

- `man [command]`: Τυπώνει τις οδηγίες χρήσης μίας εντολής (αντικαταστήστε το `[command]` με την εντολή που σας ενδιαφέρει)
- `pwd`: Τυπώνει τον τρέχοντα κατάλογο
- `ls`: Παράθεση των περιεχομένων ενός καταλόγου
- `cd [dir]`: Αλλαγή καταλόγου
- `mkdir [dir]`: Δημιουργία καταλόγου
- `nano`: Εκτελεί τον nano editor
- `cp [source] [destination]`: Αντιγράφει ένα αρχείο
- `mv [source] [destination]`: Μεταφέρει/μετονομάζει ένα αρχείο

Για μία εκτενέστερη εισαγωγή στο linux terminal παρακαλώ δείτε το <https://ubuntu.com/tutorials/command-line-for-beginner>

Για παράδειγμα στο home directory σας μπορείτε να γράψετε

```
mkdir ~/programmingI
cd ~/programmingI
mkdir lab01
cd lab01
```

ώστε να δημιουργήσετε τον κατάλογο `programmingI/lab01` και να μεταβείτε σε αυτόν. Οι ακόλουθες εντολές είναι ισοδύναμες (και πιο συνοπτικές)

```
mkdir -p ~/programmingI/lab01
cd ~/programmingI/lab01
```

## 1.2 Επεξεργασία κειμένου και μεταγλωττιστές

### Βήμα 1: Συγγραφή προγράμματος

Για τη συγγραφή προγραμμάτων θα χρησιμοποιήσουμε δύο επεξεργαστές κειμένου (editors). Τον `nano`, ο οποίος εκτελείται στο τερματικό, και το `Visual Studio Code` που είναι ένα Ολοκληρωμένο Περιβάλλον Ανάπτυξης λογισμικού (Integrated Development Environment - IDE).

Για να καλέσουμε τον `nano` ώστε να ανοίξουμε ένα αρχείο `.c` (π.χ. το `hello.c`) γράφουμε στη γραμμή εντολών.

```
nano hello.c
```

Αφού γράψουμε το πρόγραμμά μας μπορούμε να σώσουμε το αρχείο πληκτρολογώντας `C-o` (ταυτόχρονα τα πλήκτρα `Control` και `o`). Μπορούμε είτε να βγούμε από το περιβάλλον του `nano` πληκτρολογώντας `C-x` ή να πληκτρολογήσουμε `C-z` το οποίο αναστέλλει τη λειτουργία του προγράμματος και μας επιστρέφει στη γραμμή εντολών. Μπορούμε όποτε θέλουμε να επαναφέρουμε το πρόγραμμα στο προσκήνιο με την εντολή `fg` (από το "foreground").

Για να γράψουμε ένα πρόγραμμα στο περιβάλλον `Visual Studio Code` γράφουμε

```
code &
```

στην οθόνη του τερματικού. Αυτή η εντολή τρέχει το πρόγραμμα `code` (που είναι το `Visual Studio Code`). Το ampersand "&" στέλνει την εκτέλεση του προγράμματος στο background και μας επιστρέφει τον έλεγχο της γραμμής εντολών. Αν δεν το έχουμε κάνει ήδη, τότε εγκαθιστούμε το C/C++ extension, πατώντας `Ctrl-Shift-X`. Έπειτα χρησιμοποιούμε το γραφικό περιβάλλον ώστε να γράψουμε και να σώσουμε το πρόγραμμά μας.

### Βήμα 2: Μεταγλώττιση προγράμματος

Αφού ετοιμάσουμε το πρόγραμμά μας μπορούμε να το μεταγλωττίσουμε χρησιμοποιώντας τον `compiler gcc` (GNU Compiler Collection) ή/και το πρόγραμμα `make`.

Αν δεν υπάρχουν τα προγράμματα αυτά στο σύστημά σας, τότε εγκαταστήστε τα πρώτα μέσω του συστήματος διαχείρισης πακέτων της διανομής σας. Για παράδειγμα στο `Debian` ή το `Ubuntu` θα χρησιμοποιήσετε

```
sudo apt install gcc make
```

```
gcc -o hello hello.c
```

ή

```
make hello
```

Μία καλή πρακτική είναι να χρησιμοποιούμε την επιλογή `-Wall` η οποία ενεργοποιεί τις προειδοποιήσεις (warnings) του `gcc` ώστε να εντοπίζουμε πιθανά λάθη

```
gcc -Wall -o hello hello.c
```

### Βήμα 3: Εκτέλεση προγράμματος

Μπορούμε πλέον να εκτελέσουμε το πρόγραμμά μας:

```
./hello
```

Η εντολή αυτή προϋποθέτει ότι βρισκόμαστε στην ίδια τοποθεσία με το εκτελέσιμο αρχείο `hello` (η τελεία `.` αντιστοιχεί στον τρέχοντα κατάλογο). Εναλλακτικά μπορούμε να προσδιορίσουμε την πλήρη διαδρομή

```
/home/[username]/programmingI/lab01/hello
```

Επιπλέον όσων θα δούμε στο εργαστήριο μαζί, πειραματιστείτε με όλα τα παραπάνω, διαβάστε τις οδηγίες χρήσης (μέσω της εντολής `man`) και φροντίστε να τα καταλάβετε όσο καλύτερα γίνεται. Αυτό θα σας βοηθήσει να εξοικειωθείτε με τα εργαλεία ανάπτυξης που έχετε στη διάθεσή σας.

## 2 Το πρώτο πρόγραμμα σε C

Γράψτε, μεταγλωττίστε και εκτελέστε το παρακάτω πρόγραμμα σε C:

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello, world!\n");
5
6     return 0;
7 }
```

Δοκιμάστε να χρησιμοποιήσετε το nano και το code για την ανάπτυξη του προγράμματος.

## 3 Μορφοποίηση Εξόδου με την printf

Η γενική μορφή της printf φαίνεται παρακάτω:

```
1 int printf(const char *format, ...);
```

Το αλφαριθμητικό format περιέχει το κείμενο που θα εκτυπωθεί μαζί με ειδικές ακολουθίες χαρακτήρων που βοηθούν στην εκτύπωση των μεταβλητών που ακολουθούν.

Οι ειδικές αυτές ακολουθίες έχουν την εξής γενική μορφή:

`%[flags][width][.precision][length]specifier`

όπου οι περισσότεροι specifiers φαίνονται στον παρακάτω πίνακα:

specifier	Έξοδος	Παράδειγμα
c	χαρακτήρας	a
d ή i	δεκαδικός αριθμός με πρόσημο	392
e	επιστημονικός συμβολισμός με e	3.9265e+2
E	επιστημονικός συμβολισμός με E	3.9265E+2
f	δεκαδικός αριθμός κινητής υποδιαστολής	392.65
o	οκταδικός με πρόσημο	610
s	αλφαριθμητικό	sample
u	δεκαδικός ακέραιος χωρίς πρόσημο	7235
x	δεκαεξαδικός αριθμός χωρίς πρόσημο	7fa
X	δεκαεξαδικός αριθμός χωρίς πρόσημο με κεφαλαία	7FA

όλα τα άλλα στοιχεία είναι προαιρετικά.

### 3.1 Άσκηση

Γράψτε ένα πρόγραμμα που να δηλώνει μία μεταβλητή τύπου `float`, να της δίνει την τιμή 3.141592 και να τυπώνει στην έξοδο:

PI is approximately 3.141592

χρησιμοποιώντας αυτήν την μεταβλητή.

### 3.2 Άσκηση

Γράψτε ένα πρόγραμμα που να δηλώνει μία μεταβλητή τύπου `int` να της δίνει την τιμή 325 και μετά να την εκτυπώνει 3 φορές, μία σε δεκαδικό, μία σε οκταδικό και μία σε δεκαεξαδικό.

### 3.3 Ειδικοί Χαρακτήρες

Οι ειδικοί χαρακτήρες στην C φαίνονται παρακάτω:

χαρακτήρας	κωδικός ASCII	ειδικός χαρακτήρας στην C
newline	10	'\n'
tab	9	'\t'
carriage return	13	'\r'
backspace	8	'\b'
form feed	12	'\f'
backslash	92	'\\'
single quotation mark	39	'\''
double quotation mark	34	'\"'
null character	0	'\0'

Γράψτε ένα πρόγραμμα σε C που να εκτυπώνει:

```
Hello, world      Hello, world
```

```
Hello, world      Hello, world
```

## 4 Χαρακτήρες και Κωδικοποίηση

Η γλώσσα C παρέχει τον τύπο **char** για την αποθήκευση χαρακτήρων. Οι χαρακτήρες αυτοί είναι ουσιαστικά αριθμοί οι οποίοι μετατρέπονται σε χαρακτήρες χρησιμοποιώντας την κωδικοποίηση ASCII. Αυτό μας επιτρέπει να κάνουμε αριθμητική με τους χαρακτήρες.

# ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Σχήμα 1: Οι χαρακτήρες από 0 έως 31 είναι ειδικοί χαρακτήρες για τον έλεγχο των συσκευών εξόδου, πχ το 8 είναι το BACKSPACE, το 10 είναι ο χαρακτήρας newline (νέα γραμμή), το 13 είναι το ENTER και το 27 το ESC.

Γράψτε και εκτελέστε τον παρακάτω κώδικα:

```

1  #include <stdio.h>
2
3  int main() {
4      char c = 'a';
5
6      printf("letter %c is represented as %d in ASCII code\n", c, c);
7      printf("letter %c is represented as %d in ASCII code\n", c+10, c+10);
8      printf("letter %c is represented as %d in ASCII code\n", c+25, c+25);
9
10     return 0;
11 }

```

Ποια είναι η κωδικοποίηση του χαρακτήρα a; Του χαρακτήρα z;

Ανάμεσα στις γραμμές 6 και 7 βάλτε ένα printf ακόμη που να τυπώνει τον χαρακτήρα  $c + 1$ . Ποιός είναι αυτός ο χαρακτήρας;

Στην συνέχεια αλλάξτε την τιμή της μεταβλητής c σε 'A'. Ποια είναι η κωδικοποίηση του 'B' με βάση το ASCII;