Curs 1 - Sisteme de operare

Putty

```
ls /bin /usr/bin | grep "^[a-z]*sh$" | sort
```

→ nu există ~~driv~~ C, D ...

/ → root ♥

| → pipe

→ Ctrl+C nu copiază în linia de comandă
Ctrl+S oprește consola (ctrl+q ca să își revină)
dacă se selectează ceva - se copiază în clipboard

ls -a (arată și fișiere ascunse : încep cu .)

ls -l (arată detalii)

ls --color=none (auto ...)

licență → tematică

space ↑
B ↓ } MAN
Q ieșe

apropos sort

Ctrl + L → curăță ecranul

whatis open

man 2 open → secțiunea 2 a manualului

editor : - vi
- emacs
- nano
- joe

compilare - gcc

mkdir → make director

vi hello-world.c

i → mod editare

TAB = autocomplete

```c
#include <stdio.h>
int main (int argc, char **argv) {
    return 0;
}
```

```
gcc -Wall -g -o prog fis.c
```

-Wall: adaugă toate warning
-g: pt. debug
-o prog: fișier obj

./prog
↓
din directorul curent

CTRL-U undo

VALGRIND → DETECTEAZĂ PROBLEME DE MEMORIE

```
#include <stdio.h>
int main (int argc, char *argv) {
        printf ("Salutare lume!\n");
        return 0;
}
```

`cp hello-world.c hello-name.c`

```
#include <stdio.h>
int main (int argc, char **argv) {
        char name[64];
        while ( scanf ("%s", name) == 1 ) {
        }
        return 0;
}
```

`printf ("Salut %s!\n", name);`

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

CTRL-D → sfârșit de fișier, gata input

CTRL-C → omoară programul

`cp hello-world.c hello-remember.c`

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
struct node {
        char *name;
        struct node* next;
};
struct node* add (struct node* head, char *name) {
        struct node *n;
        struct node *p;
        n = (struct node *) malloc (sizeof (struct node));
        n->name = (char *) malloc (strlen (name) + 1);
        strcpy (n->name, name);
        n->next = NULL;
        if (head == NULL) {
                return n;
        }
        p = head;
        while (p->next != NULL) {
                p = p->next;
        }
        p->next = n;
        return head;
}
```

TL CONVERTIM → RETURNEAZĂ POINTER

```
struct
int known (struct node * head , char * name ) {
        struct  node * p;
        if (head == NULL) {
                return 0;
        }
        p = head ;
        while (p != NULL && strcmp (p -> name, name ) != 0 ){
                p = p -> next ;
        }
        return p != NULL;
}

int main ( int argc , char ** argv ) {
        char name [64] ;
        Struct  node * head = NULL ;
        while (scanf ("%s",  name ) == 1 ) {
                if (known (head, name )) {
                        printf ("Tot pe aici, %s?", name);
                else {
                        printf ( "salut, %s!", name );
                        add (head, name);
        }
        return 0;
}

valgrid    ./name-file

void clear (struct node * head ) {
        if (head == NULL) {
                return;
        }
        clear (head -> next);
        free (head -> name);
        free (head);
}
```