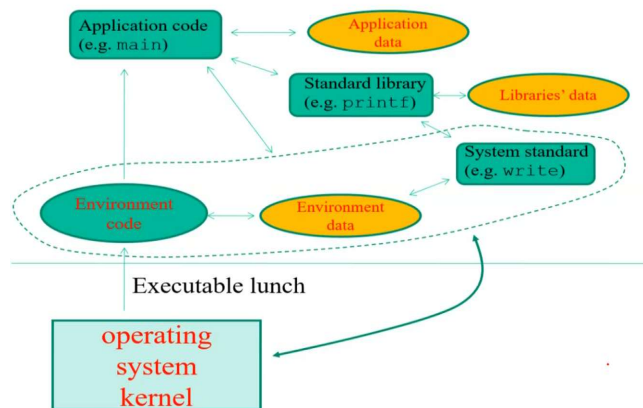


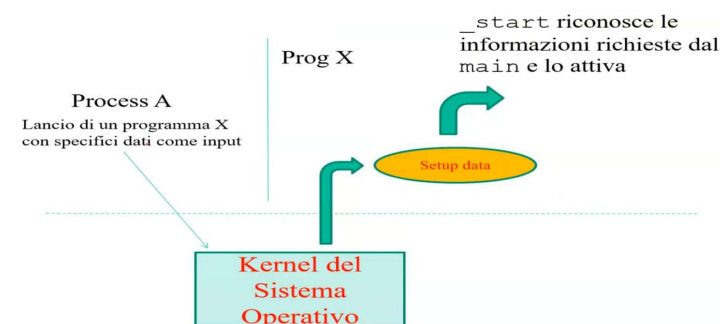
# Introduzione10 Ambiente

venerdì 10 ottobre 2025 17:45

In generale un qualunque programma compilato si ha il codice applicativo vero e proprio e si ha anche un insieme di moduli software (eventuali dati) chiamati **ambiente di esecuzione**. In generale il *main* non è il primo programma/funzione che viene invocato: è lo *\_start*. Quindi l'ambiente serve per raccordare il SO ed il codice applicativo, inoltre serve come mezzo per poter pilotare esecuzione di specifiche funzioni di libreria (i dati possono essere importanti per il software). In dettaglio:



Mentre se lancio un programma la situazione è la seguente:



Nota: setup data è address space, tra cui eventuali parametri da passare al main. In generale si può avere una situazione del genere: vogliamo compilare un programma a prescindere dall'ambiente, quindi possiamo sviluppare applicazioni solamente usando i dati ambientali (ignorando tutto il resto) e impedendo accesso a moduli già scritti. Vediamo come farlo: in UNIX si usa il flag **-nostartfiles** e si deve avere una funzione **\_start**. Altrimenti se non vi è presente questa funzione, la prima funzione invocata sarà quella presente nella sezione .text. Riassumendo: l'ambiente ci fa da raccordo tra le attività che il SO esegue per andare in startup e ciò che il programmatore vuole realizzare. Inoltre quando si esegue un programma, viene comunque lanciata una system call, la quale contiene una serie di informazioni che verranno mandate in startup nella zona ENVIRONMENT DATA. Vediamo alcuni esempi:

```
//please compile with -nostartfiles
#include <stdio.h>
#include <stdlib.h>

void _start(void){
    printf("hello world!\n");
    exit(0); //beware removing this!!
}
```

Il quale stampa "Hello word", **ma deve compilato con il flag -nostartfile**. Ma cosa succede quando commentiamo la syscall exit(0)? **Il sistema va in core dump!!** In quanto il ritorno da subroutine ha un valore errato di ritorno nella stack area per tornare al chiamante. Esempio 2 ( busy loop):

```
//please compile with -nostartfiles

int f(){
    while(1){
    }
}
```

```
/usr/bin/ld: attenzione: impossibile trovare il simbolo d'ingresso _start; ripristinato il predefinito 0000000000001000
```

```
//please compile with -nostartfiles

#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

//presente all'interno di qualche libreria esterna
//anche questo è array di puntatori
//inizializzato dal kernel del sistema operativo
extern char** environ;

void _start(void){
    //array di puntatori-> strings
    char ** addr = environ;

    printf("environ head pointer is at address: %u\n",(unsigned long)environ);

    while(*addr){
        //prendo il valore (stringa) contenuta in addr
        printf("%s\n",*(addr));
        //vado alla cella successiva del mio array di puntatori
        addr++;
    }
    exit(0);
}
```

```
SHELL=/bin/bash
SESSION_MANAGER=local/daniele-Vivobook-ASUSLaptop-X1605VA-X1605VA:/tmp/.ICE-unix/2544,unix/daniele-Vivobook-ASUSLaptop-X1605VA-X1605VA:/tmp/.ICE-unix/2544
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GNOME_KEYRING_CONTROL=/run/user/1000/keyring
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
MEMORY_PRESSURE_WARNIT=c29tZSAYMDAwMDAgIjAwMDAwMAA=
XMODIFIERS=@xkb:ibus
DESKTOP_SESSION=ubuntu
GTK_MODULES=gail:atk-bridge
PWD=/home/daniele/Scaricati/c-BASICS
LOGNAME=daniele
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=wayland
SYSTEMD_EXEC_PID=2005
XAUTHORITY=/run/user/1000/.nutter-Xwaylandauth.R1303
HOME=/home/daniele
USERNAME=daniele
IM_CONFIG_PHASE=1
```

addr

Char \*\* addr

S1  
S2  
S3  
S4  
S5  
i