

الـ Program file المعرض يكون جواه ايـ ؟  
hexa و Elf binary Format  
ـ واد انا الحال initial (init) instructions  
ـ الـ entry Point initial data الـ data  
ـ الـ values الـ string او الـ variables مثلـ printf("Hello");  
ـ الـ symbol & relocation tables المـ information عن اماكن و اـ ايـ الـ variables  
ـ الـ code عنـ اـ ايـ الـ variables  
ـ الـ debugging المـ information المـ known عنـ اـ ايـ الـ variables  
ـ الـ shared libraries المـ information عنـ اـ ايـ الـ shared libraries  
ـ الـ dynamic linker (Pathname) المـ information عنـ اـ ايـ الـ dynamic linker  
ـ الـ symbol and relocation tables المـ information عنـ اـ ايـ الـ symbol and relocation tables  
ـ الـ shared-library and dynamic-linking information المـ information عنـ اـ ايـ الـ shared-library and dynamic-linking information

Ch6: Func. Processes

## Processes and Programs

A process is an instance of an executing program.

A program is a file containing a range of information that describes how to construct a process at run time.

ـ حقوق

Information exists in a program file:

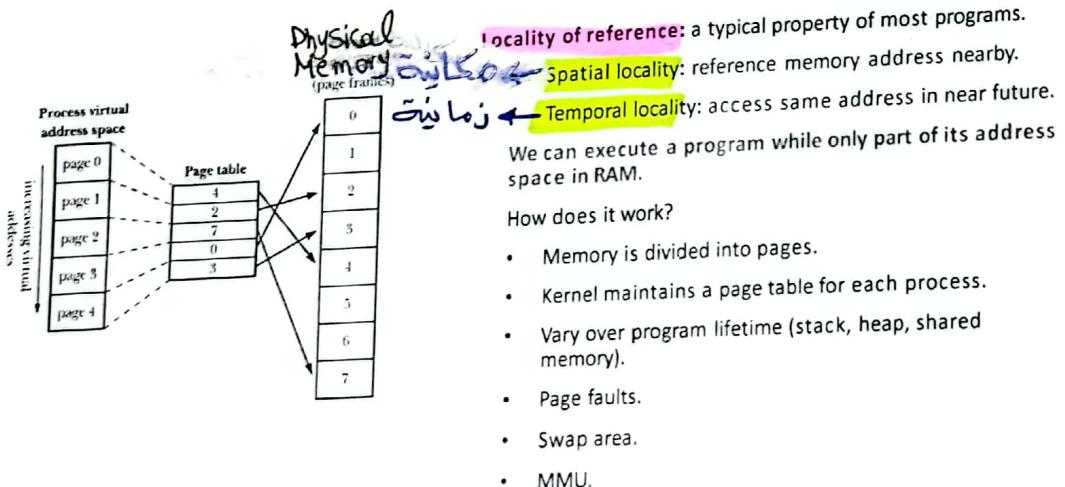
1. Binary format identification (a.out, COFF, ELF).
2. Machine-language instructions.
3. Program entry-point address.
4. Data (initial values and literal strings).
5. Symbol and relocation tables.
6. Shared-library and dynamic-linking information.

## Memory

الـ Process يعني نفسها مكونة من أربعة من جواه الدول في يتكون لـ Process components و user-space، Kernel-space، Variables، user-space يكون فيها الـ code والـ Variables أما الـ Kernel ففيها الحالات التي الـ Kernel تحتاجها أي Kernel Space

- A process consists of:
    - User-space memory containing program code and variables.
    - kernel data structures that maintain information about the state of the process:
      - 1). Process IDs (/proc/sys/kernel/pid\_max, getpid(), getppid()). → group ID, Pro
      - 2). Virtual memory tables.
      - 3). table of open file descriptors.
      - 4). Resource usage and limits.
      - 5). Others.

## Virtual Memory Management



3 فسال کہہ سؤال حلو، انت تکریف  
نہ مارے physical address بالا virtual address

الدجابت اه ، الحوار ممكن يتکمل کاری ، بس چیز  
ایشت کل address هست و تزود یا یعنی base به کانه  
فی ال Memory ، اما ال MMU بترکیح توندیل الحوار  
ده بال HW خلی ایش

طلب لو انت جاي تجيء address مُكتب و لقيته مش موجود؟ يعنى ~~Page Fault exception~~ والـ ~~Kernel~~ يتروح تجيء ~~Page~~ رى، ولو ~~RAM~~ عنده ملائمة، في ~~hard disk~~ عنده في ~~swap area~~ رى ~~hard disk~~ بترجع ترمي فيها ~~Page~~ عندها تحت مكانها جديرة، ووادب في اكتر ~~Page~~ غير مُستخدم او اقدم واحدة او اجدد واحدة على حسب ~~Algorithm~~ عنده

## Virtual Memory Advantages &

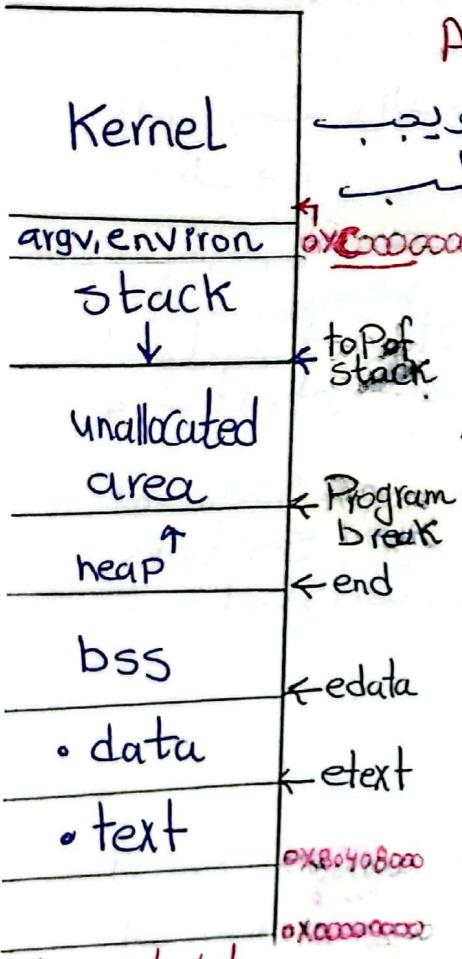
١- يُتَكَبِّلُ الـ Process بِتَأْسِيَتِكَعْ عَنْ بَاقِي الـ Processes  
عَنْهَا نَلْعَجُهُمْ لِعَوْنَاطِشِهِمْ اَعْمَلْتُمُهُمْ كَعَنْ تَأْسِيَتِكَعْ  
فِي Kernel هُنْ الـ isolated

بـ Share نفس الـ Memory بين processes كاري، يعني لو عندك متلا 4 برامج يستخدموا Vim متلا، هل لازم تتحط Vim عندك في الـ Memory بـ 4 مرات؟ لا حلها في Page 5 متلا، وروح اعمل (map) 4 Processes executes متلا ان الـ Vim في Page 5، هلفارقها معاهم؟ لا وهو بيـ و خلاصه مش خارق دعاه، كمان انت بـ كل fork عمظـم الوقت بتتحمل exec، خلينه تتحمل duplication فالـ Kernel بيـ تحمل ان Pages بين child والـ parent، لحد ما الـ child يـدأ يكتب لـ اـ يـحصل duplication، عـيل كده لا

كل الـ Program ينتهي بـ End ، لأنك منك تحتاج End - 5  
كثبات الـ Program ينتهي بـ End منه إلى الـ End لازم يكون موجود

## \* Memory Layout of a Process \*

\***بهم فنكك من الـ segments**  **انت کارفها من قبل کرد**  **و ذاکرتهما کتیر فاکت کید خاکرها**



A- Kernel :-

## B- argv, environ-

دعا الى انت Command line argument \*  
environment ، main ، Cmd بتبينها بالـ \$PATH Variables

## C-stack & heap -

end, etext, edata:-

طبا ناکشن اشوف الکلام رو یعنی الـelf

اميل -s readelf لاعر segments هتلائق ال ELF والـ etext edata وغيرها

او ممکن تطلع ال file او map file مختص من جو ا Sections و بیغولک کل Segment باری مین و عنیه ایه و خلماں فین و بیجت حاجات حلوا کردا

```
gcc -Wl,-Map=file.map file.c
```

او هنّاك Command size بيقولك حجم الملف file ، بس الاختيار اسْتَخِذ sections

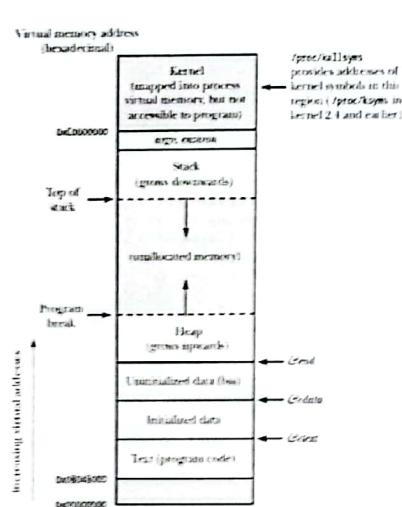
## \* Stack & Stack frame \*

- \* الـ Kernel بيكون ليها stack بعدين عن user كمان لوالـ user بوظ حاجت ميلكيتش في kernel والا بتعتبر مهمش
- \* كل Function عندك بنتكل بيكون ليها **Stack frame** خاص بيها ي يكون فيه **Function arguments** والـ **local vr.** وحيث الـ **Regs** اللي ليهم علاقة بيها والمكان اللي في وقفت فيه، كمان اما تكمل **call** لـ **Function** تاينته من جواها تعرف ترجع واما ارجع ~~stack~~ للـ **Function** اكمله او اخرك الـ **SP** فوقته والـ **Stack frame** بيكون موجود في **userstack**

# Virtual Memory Advantages

- 1 Process isolation (from other process and from the kernel).
- 2 Memory sharing.
  - ↳ Executing the same program. → Vim
  - ↳ IPC. → Fork "IPC → inter Process Communication"
- 3 Memory Protection (read-only, execute-only, RW).
- 4 Compiler and linker don't need to be concerned with the physical layout.
- 5 Loading programs faster.
- 6 Better CPU utilization.

# Memory Layout of a Process



## Process memory layout consists of:

- Text (read-only, sharable).
- Data.
- BSS (Block Started by Symbol).
- Stack.
- Heap (Program Break)

## Notes:

- Size command.
- etext, edata, end.
- Reentrant functions.

Formats چیزی کے منو اتنی options ہے ۔

## long-term Format:-

\* دع ي يكون قبلاً --، وانت بتكتب الـ option اكتر من حرف، كمثال:

→ ls --all

\* دعای بیک فن قبلها - و مکتب ال وحدت مثال Short-term formats

→ ls -a

\* مع العلم ان سواد-a او all- هي ولي نفس الحاجة عادي

note that :-

$$\Rightarrow Ls_{-\alpha-f} = Ls_{-\alpha f} = Ls_{-\alpha f} \Leftarrow$$

\* طب لو کندی Command بیاخد arguments و option و عملیات  
تلزق انتین options ی بکھن؟ کاری، بس المهم خلی  
آخر واحد هو الی بیاخد argument، یعنی ان option argument  
بیاخد argument و نہ لا، اکتیها کرہ

# Cmd -BA arg ✓

\* Cmd -AB arg ~~X~~

getopt

\* زعـ ما قـولـتـكـ في library routine بـسـخـ مـهـاـكـ شـاتـ  
الـ Parsing بـتـاعـ الـ Command ، لو فـتحـتـ الـ معـنـيـاتـ  
دـيـاعـتـها فـتـلـقـ بـيـقـولـكـ أـنـكـ بـتـبـكـلـهـاـ الـ argcـ وـ الـ argvـ  
الـ أـنـكـ بـيـتـبـكـلـهـاـ لـ الـ mainـ بـتـاعـتـهـاـ مـنـ الـ terminalـ ، وـ بـيـكـرـ غـلـكـ  
الـ elementـ بـتـاعـ الـ argvـ أـنـهـ أـبـ حـاجـتـ بـتـبـ أـبـ - او--  
لـ كـنـ مـشـ - او-- بـسـ لـدـنـ رـولـ لـ وـحـدـهـمـ فـيـ Commandsـ  
بـتـكـبـرـهـمـ optionـ اـلـاـ وـاـنـ getoptـ رـىـ كـلـ مـاـ تـعـلـمـهـاـ  
بـتـجـيلـكـ الـ optionـ الـ أـلـيـ عـلـىـ الـ دـورـ بـعـنـ اـولـ مـرـةـ تـجـيلـكـ  
اـولـ optionـ ، تـأـيـ مـرـةـ تـجـيلـهـاـ callـ فـنـهـاـ تـجـيلـكـ تـأـيـ optionـ  
وـ هـكـاـ ، وـ بـتـحـرـفـ الـ indexـ بـتـاعـ الـ optionـ  
اـ هـهـ optindـ دـهـ بـيـشـاـورـ عـالـ optionـ الـ جـائـ وـ تـقـرـ تـعـلـمـهـاـ  
variablesـ optionـ resetـ

Formats خیالاتیں کے ساتھ options ॥ \*

## long-term Format :-

دیجیکالا فرم فرمات <sup>option</sup> دیجیکالا فرم فرمات <sup>\*</sup>  
اکثر من حرف کھالے: وانت بتکتب ال

→ ls --all

\* دعى بيكيفن قبلها - وبكتبه الـ Short-term formats

→ ls -a

\* مع العلم أن سواداً - او الـ - all - في ولد نفس الحاجة عادى

note that :-

$$\Rightarrow Ls_{-\alpha-f} = Ls_{-\alpha f} = Ls_{-\alpha f \alpha} \Leftarrow$$

\* طب لو کندی Command بیاخد arguments و option و علیز  
تلزق اتنین options ی بکھن؟ کاری، بس المهم خلی  
آخر واحد هو الی بیاخد argument، بمهنی ان option argument  
بیاخد argument و لا، اکتیها کرہ

# Cmd -BA arg ✓

\* Cmd -AB arg ~~X~~

getopt

\* زعَ ما قَوْلِتَكَ فِي library routine بِسْتَخْدِمَهَا كَثِيرًا

الـ getopt بِتَابَعَ الـ Command، لَوْفَتَحَتَ الـ Parsing  
بِتَابَعَهَا فَتَلَقَّى بِيَقْوِلِكَ أَنَّكَ بِتَدْبِيَّلِهَا الـ argc وَالـ argv  
الَّتِي بِيَتَبَكِّرُوا لِلـ main بِتَابَعَهَا مِنَ الـ terminal وَبِيَكْرِيَّلِكَ  
الـ element بِتَابَعَ الـ argv أَنَّهُ أَدِيَ حَاجَتَهُ بِتَدْبِيَّلِهِ - او --  
لَكَنْ مَشْ - او -- بِسْ لَدَنْ - وَلَ لَوْدِقْمِ فِي Commands  
بِتَكْبِيرِهِمْ option اَلَا لَا وَانْ getopt رَى كُلَّ مَا تَعْمَلُهَا  
بِتَجْبِيلِكَ option الَّتِي عَلَيْهِ الـ loop، يَعْنِي أَوْلَ مَرَةٍ تَجْبِيلُكَ  
أَوْلَ option، تَأْتِي مَرَةٍ تَجْمَلُهَا call فِيهَا تَجْبِيلُكَ تَأْتِي option  
وَهُكَذا، وَبِتَحْرِفِ الـ index بِتَابَعَ الـ variables option  
اَمْهَدْ optind دَهْ بِيَشَارِعِ الـ option الَّتِي جَاءَ وَتَقْدِرُ تَعْمَلُهَا  
reset

10 \* وكل ما يكون سـ في كذلك argument او option getopt بترجمـه وـكـ اما تـمـلـها call وـتـنـيـطـ optind انه يـقـيـسـ index الى option الى جـائـيـ وـلـوـ مـفـيـشـ بـتـرـجـعـلـكـ

\* بـرـدـو بـتـاـخـدـ منـكـ الـ optstring وـدـهـ الـ string الـى  
بـتـتـسـلـيـ اـلـاـمـهـ بـيـكـوـنـ فـيـ الـ options الـاـخـرـ  
اـنـتـ بـتـسـتـاـخـدـهاـ،ـ يـكـيـفـ مـتـلـاـ اـنـتـ عـاـمـلـ الـ p~rogramـ  
بـتـاـخـدـ الـ options الـ aـ وـ الـ cـ بـسـ،ـ لـوـ يـكـيـتـ دـ المـعـرـوـضـ  
تـنـظـلـلـ الـ errorـ،ـ فـانـتـ بـتـرـجـعـ تـبـتـتـ فـيـ الـ aـ وـ الـ cـ

طلب بـيـكـتـ الـ optionsـ لـيـ اـلـاـسـ اـيـهـ؟

\* بـيـكـتـ الـ aـلـوـلـ بـتـرـجـعـ تـعـطـهـ :ـ،ـ وـاـيـeـ optionـ  
بـتـاـخـدـ argumentـ حـلـ بـعـدـهـ :ـ بـالـسـكـلـدـ "ـP~Xـ"ـ  
كـدـهـ Pـ بـيـ expectـ انهـ يـجـبـلـ argumentـ وـالـ Xـ لـدـهـ وـ  
الـ Pـ وـالـ Xـ فـيـ الـ optionsـ الـوـحـيدـيـنـ الـىـ مـوـجـوـدـيـنـ  
كـذـكـ وـلـوـ مـطـيـتـ اـتـيـنـ (ـ)ـ مـكـنـاـهـاـ انـ الـ argـ دـ optionalـ

طلب الـ :ـ الـ aـلـوـلـ زـيـ لـيـ؟

\* عـيـنـاتـ لـوـفـيـ كـذـكـ مـسـكـلـاتــ،ـ انـ مـتـلـدـ فـيـ expectـ optionـ  
عـنـدـكـ argumentـ وـاـنـتـ مـيـسـتـوـشـ بـرـجـعـلـكـ (ـ)ـ بـلـ ماـ  
يـكـلـ errorـ وـيـخـرـجـ

طلب الـ argumentـ فـيـرـجـعـ خـيـنـ؟

\* فـيـعـنـدـكـ optargـ اـلـ extendـ globalـ varـ مـعـمـولـهـ  
يـرـتـخـزـنـ فـيـ الـ argumentsـ

Example :-

```
int main (int argc, char* argv[])
{
    int int ch;
    while ((ch = getopt (argc, argv, "p:x")) != -1)
    {
        printf ("option is %c", ch);
        if (ch == 'p')
            printf ("Argument is %s", optarg);
        else if (ch == 'x')
            printf ("No arguments");
        else
            printf ("Error !!");
    }
}
```

3ay

341

## \*Environment List \*

○ ————— // ————— // ————— ○

\* الـ Environment var انتـ حارفـهم ، بـ الـ Command line Arguments  
يـكونـ عـلـيـهـا الـ env. var كلـها تـنـتـقـيـ بـ Null

وكان child لل Parent هو one way fork أو exec one-time ينفي وقت ال

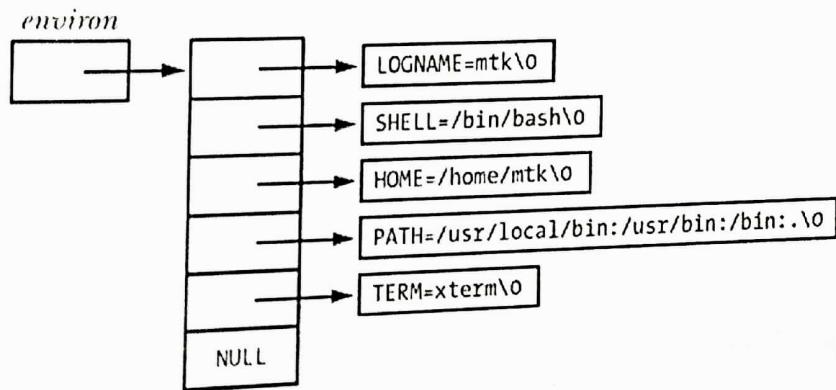
\*  
فـ `var` يـ `environ` تـ `getenv` تـ `getenv` يـ `al`  
ـ `/proc/pid/environ` او `environ` `env. var.` بـ `setenv`

کہ env. var. کی ~~کالبی~~ Printenv اور Command ~~کے~~ ~~کی~~ کال سیستم کے

unsetenv env var  $\rightarrow$  the env var is unset  $\rightarrow$  1 command is executed

فی عینک عدالتی کی مسحہ  
Env. Var || Clear, clearEnv | Command  
و نکلنا اول ما تکمل لرنامہ جسے

# Environment List



- Place of argv and environ arrays.
- One-way, one-time transfer from parent to child.
- environ variable.
- It can be passed to main() but with limited scope.
- /proc/PID/environ.
- export, printenv, unset commands.
- Modifying env in C: setenv, unsetenv.
- Clear env (possible leaks).