

# 网安实验报告 1

57119118 尤何毅

## Task 1

### 实验内容：

使用 `printenv` 或者 `env` 之类打印环境变量。

使用 `export` 和 `unset` 指令设置或者删除环境变量。

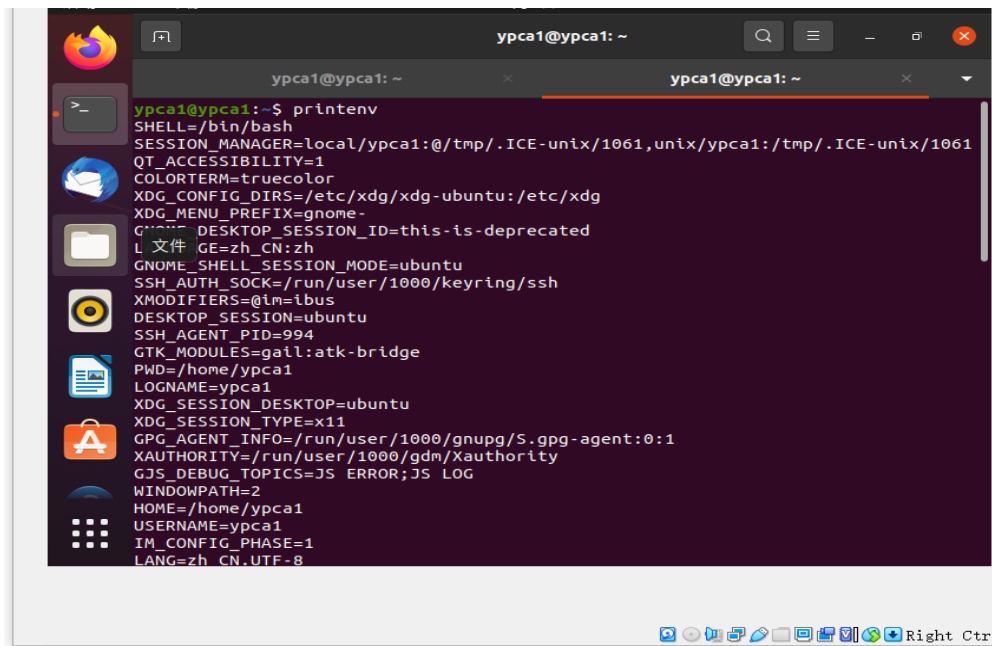
### 实验目的：

学习有关环境变量设置和删除的指令。

### 运行结果：

```
ypca1@ypca1:~$ /etc/passwd
bash: /etc/passwd: 权限不够
ypca1@ypca1:~$ env passwd
更改 ypca1 的密码。
Current password:
passwd: 认证令牌操作错误
passwd: 密码未更改
```

`printenv` 指令成功打印环境变量。



```
ypca1@ypca1: ~
ypca1@ypca1: ~
ypca1@ypca1:~$ printenv
SHELL=/bin/bash
SESSION_MANAGER=local/ypca1:~/tmp/.ICE-unix/1061,unix/ypca1:/tmp/.ICE-unix/1061
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
LANG=zh_CN:zh
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=994
GTK_MODULES=gail:atk-bridge
PWD=/home/ypca1
LOGNAME=ypca1
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/run/user/1000/gdm/Xauthority
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
WINDOWPATH=2
HOME=/home/ypca1
USERNAME=ypca1
IM_CONFIG_PHASE=1
LANG=zh_CN.UTF-8
```

```
ypca1@ypca1:~$ TESTVAR="this is a test variable"
ypca1@ypca1:~$ echo $TESTVAR
this is a test variable
ypca1@ypca1:~$ printenv TESTVAR
this is a test variable
ypca1@ypca1:~$ export TESTVAR
ypca1@ypca1:~$ printenv TESTVAR
this is a test variable
ypca1@ypca1:~$ unset TESTVAR
ypca1@ypca1:~$ printenv TESTVAR
ypca1@ypca1:~$
```

使用 `export` 指令成功设置了一个环境变量 `TESTVAR`，并用 `printenv` 访问；之后用 `unset` 将其删除，再访问则打印不出值了。

实验体会：

使用 `export` 和 `unset` 可以直接对环境变量进行操作，十分方便。

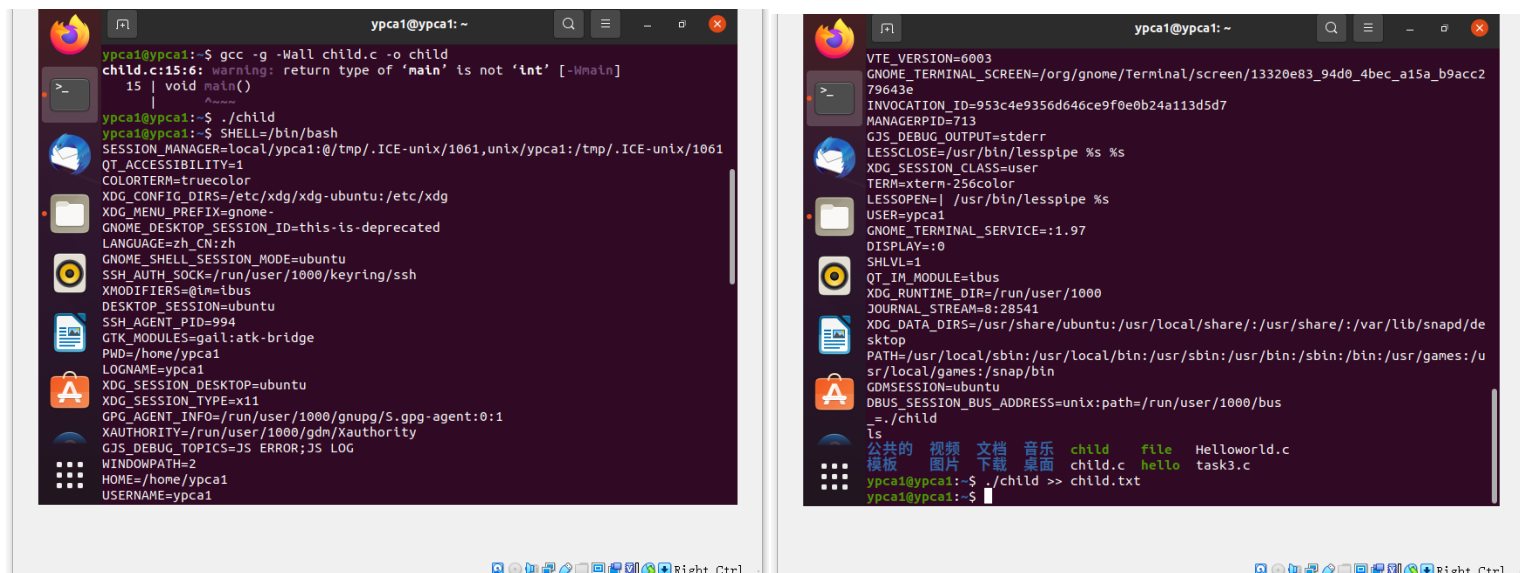
## Task 2

实验内容：编译已有程序并运行，再把输出结果存在一个文件中

实验目的：研究子进程如何从其父进程获取其环境变量

源程序：

步骤 1

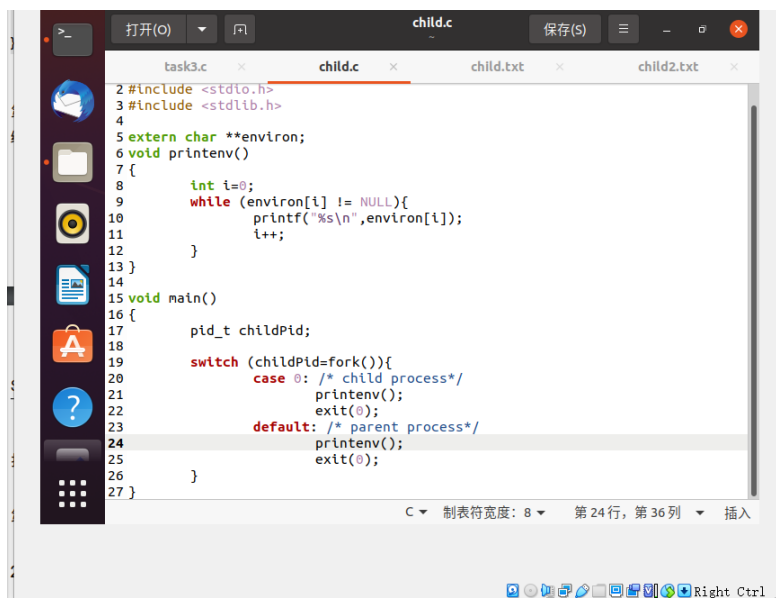


```
task3.c  ×  child.c  ×  child.txt  ×
1 #include <unistd.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 extern char **environ;
6 void printenv()
7 {
8     int i=0;
9     while (environ[i] != NULL){
10         printf("%s\n",environ[i]);
11         i++;
12     }
13 }
14
15 void main()
16 {
17     pid_t childPid;
18
19     switch (childPid=fork()){
20         case 0: /* child process*/
21             printenv();
22             exit(0);
23         default: /* parent process*/
24             //printenv();
25             exit(0);
26     }
```

步骤 2

```
ypca1@ypca1:~$ ./child2 >> child2.txt
ypca1@ypca1:~$ gcc -g -Wall child.c -o child2
child.c:15:6: warning: return type of 'main' is not 'int' [-Wmain]
 15 | void main()
    |
ypca1@ypca1:~$ ./child2 >> child2.txt
ypca1@ypca1:~$
```

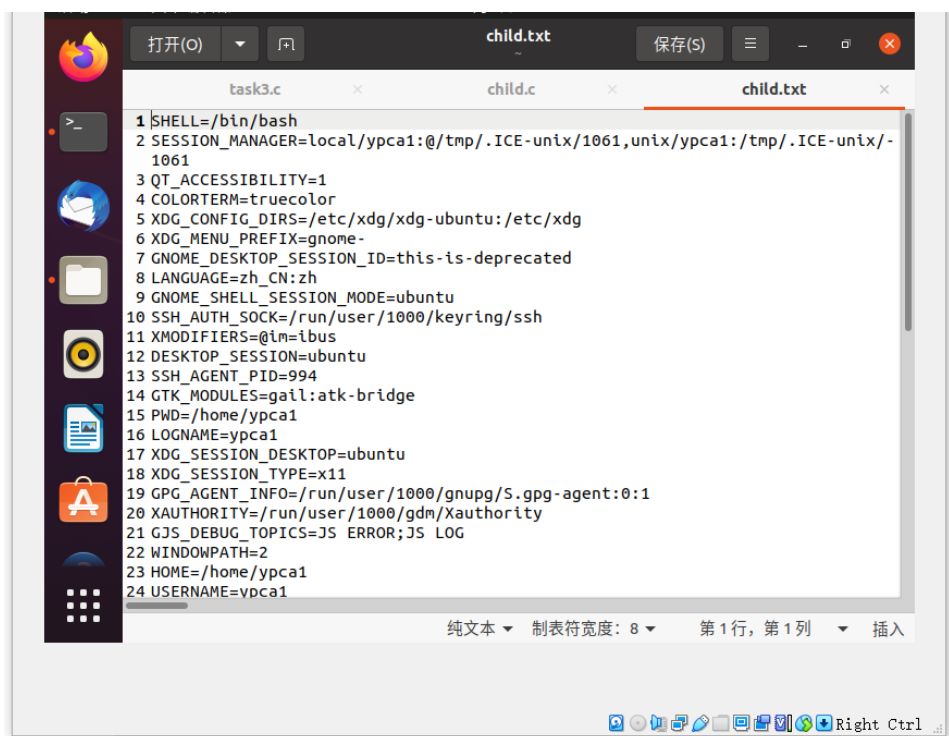
修改后的 child.c



```
task3.c  ×  child.c  ×  child.txt  ×  child2.txt  ×
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 extern char **environ;
6 void printenv()
7 {
8     int i=0;
9     while (environ[i] != NULL){
10         printf("%s\n",environ[i]);
11         i++;
12     }
13 }
14
15 int main()
16 {
17     pid_t childPid;
18
19     switch (childPid=fork()){
20         case 0: /* child process*/
21             printenv();
22             exit(0);
23         default: /* parent process*/
24             printenv();
25             exit(0);
26     }
27 }
```

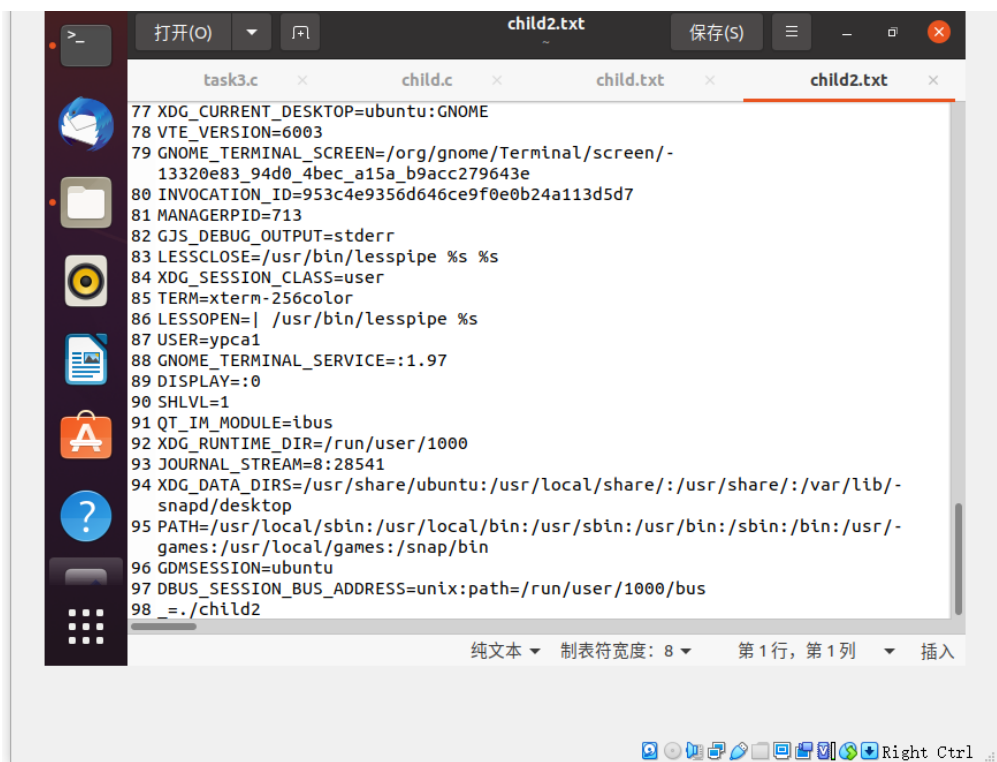
运行结果：如图，成功存进了 child.txt 文件和 child2.txt 文件中

结果显示，新进程成功读取了环境变量，并且存储在一个 txt 文件中可以查看。最终结果显示，区别在于取消注释后输出了 parent process 中环境变量的值，而第一次输出到 child.txt 只有 child process 的环境变量。



The screenshot shows a terminal window with three tabs: task3.c, child.c, and child.txt. The child.txt tab is active and displays a list of environment variables, each preceded by a line number from 1 to 24. The variables include SHELL, SESSION\_MANAGER, QT\_ACCESSIBILITY, COLORTERM, XDG\_CONFIG\_DIRS, XDG\_MENU\_PREFIX, GNOME\_DESKTOP\_SESSION\_ID, LANGUAGE, GNOME\_SHELL\_SESSION\_MODE, SSH\_AUTH\_SOCK, XMODIFIERS, DESKTOP\_SESSION, SSH\_AGENT\_PID, GTK\_MODULES, PWD, LOGNAME, XDG\_SESSION\_DESKTOP, XDG\_SESSION\_TYPE, GPG\_AGENT\_INFO, XAUTHORITY, GJS\_DEBUG\_TOPICS, WINDOWPATH, HOME, and USERNAME.

```
1 SHELL=/bin/bash
2 SESSION_MANAGER=local/ypca1:@/tmp/.ICE-unix/1061,unix/ypca1:/tmp/.ICE-unix/-
  1061
3 QT_ACCESSIBILITY=1
4 COLORTERM=truecolor
5 XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
6 XDG_MENU_PREFIX=gnome-
7 GNOME_DESKTOP_SESSION_ID=this-is-deprecated
8 LANGUAGE=zh_CN:zh
9 GNOME_SHELL_SESSION_MODE=ubuntu
10 SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
11 XMODIFIERS=@im=ibus
12 DESKTOP_SESSION=ubuntu
13 SSH_AGENT_PID=994
14 GTK_MODULES=gail:atk-bridge
15 PWD=/home/ypca1
16 LOGNAME=ypca1
17 XDG_SESSION_DESKTOP=ubuntu
18 XDG_SESSION_TYPE=x11
19 GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
20 XAUTHORITY=/run/user/1000/gdm/Xauthority
21 GJS_DEBUG_TOPICS=JS ERROR;JS LOG
22 WINDOWPATH=2
23 HOME=/home/ypca1
24 USERNAME=ypca1
```



The screenshot shows a terminal window with four tabs: task3.c, child.c, child.txt, and child2.txt. The child2.txt tab is active and displays a list of environment variables, each preceded by a line number from 77 to 98. The variables include XDG\_CURRENT\_DESKTOP, VTE\_VERSION, GNOME\_TERMINAL\_SCREEN, INVOCATION\_ID, MANAGERPID, GJS\_DEBUG\_OUTPUT, LESSCLOSE, XDG\_SESSION\_CLASS, TERM, LESSOPEN, USER, GNOME\_TERMINAL\_SERVICE, DISPLAY, SHLVL, QT\_IM\_MODULE, XDG\_RUNTIME\_DIR, JOURNAL\_STREAM, XDG\_DATA\_DIRS, PATH, GAMES\_PATH, GDMSESSION, DBUS\_SESSION\_BUS\_ADDRESS, and the command being executed.

```
77 XDG_CURRENT_DESKTOP=ubuntu:GNOME
78 VTE_VERSION=6003
79 GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/-
  13320e83_94d0_4bec_a15a_b9acc279643e
80 INVOCATION_ID=953c4e9356d646ce9f0e0b24a113d5d7
81 MANAGERPID=713
82 GJS_DEBUG_OUTPUT=stderr
83 LESSCLOSE=/usr/bin/lesspipe %s %s
84 XDG_SESSION_CLASS=user
85 TERM=xterm-256color
86 LESSOPEN=| /usr/bin/lesspipe %s
87 USER=ypca1
88 GNOME_TERMINAL_SERVICE=:1.97
89 DISPLAY=:0
90 SHLVL=1
91 QT_IM_MODULE=ibus
92 XDG_RUNTIME_DIR=/run/user/1000
93 JOURNAL_STREAM=8:28541
94 XDG_DATA_DIRS=/usr/share/ubuntu:/usr/local/share:/usr/share:/var/lib/-
  snapd/desktop
95 PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/-
  games:/usr/local/games:/snap/bin
96 GDMSESSION=ubuntu
97 DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
98 _=./child2
```

## 实验体会：

编译初因为不清楚如何将编译的结果直接显示出来使用了一些时间进行网上搜索，最终得到解决。

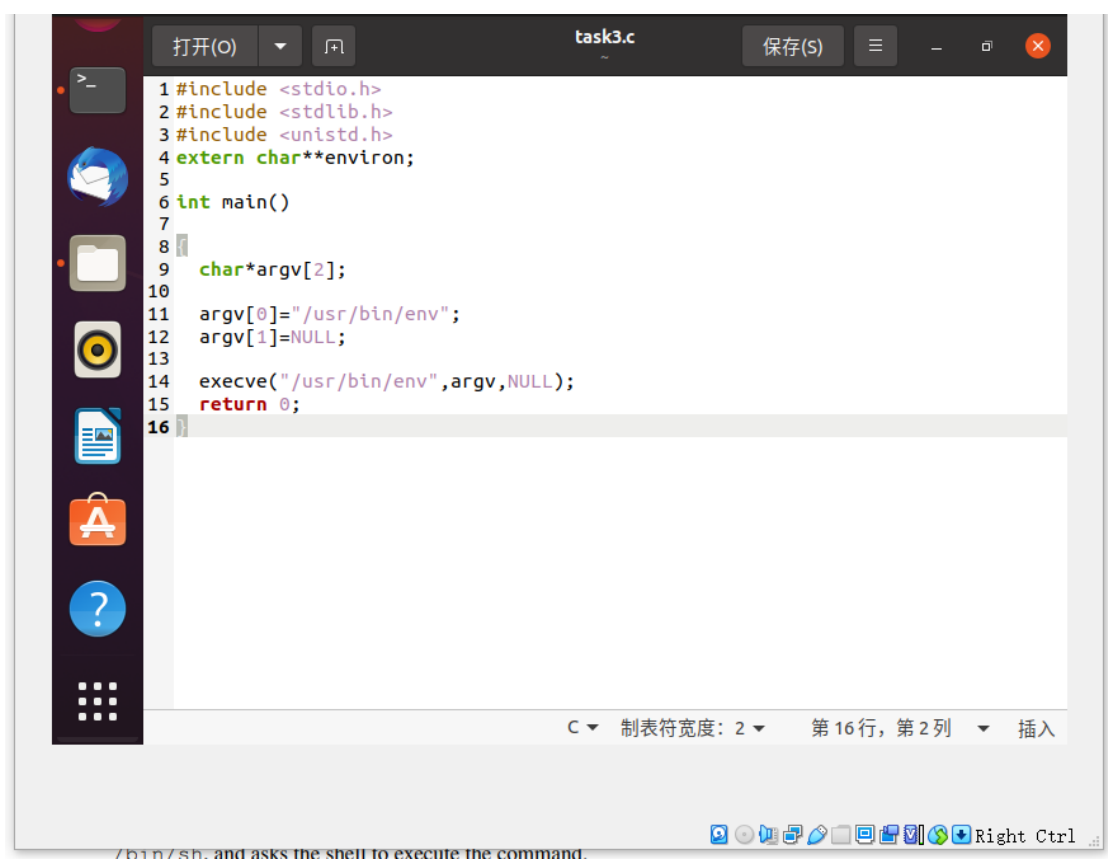
## Task 3

实验内容：编译并运行程序，描述观察到的结果。

实验目的：了解环境变量和 `execve()` 函数

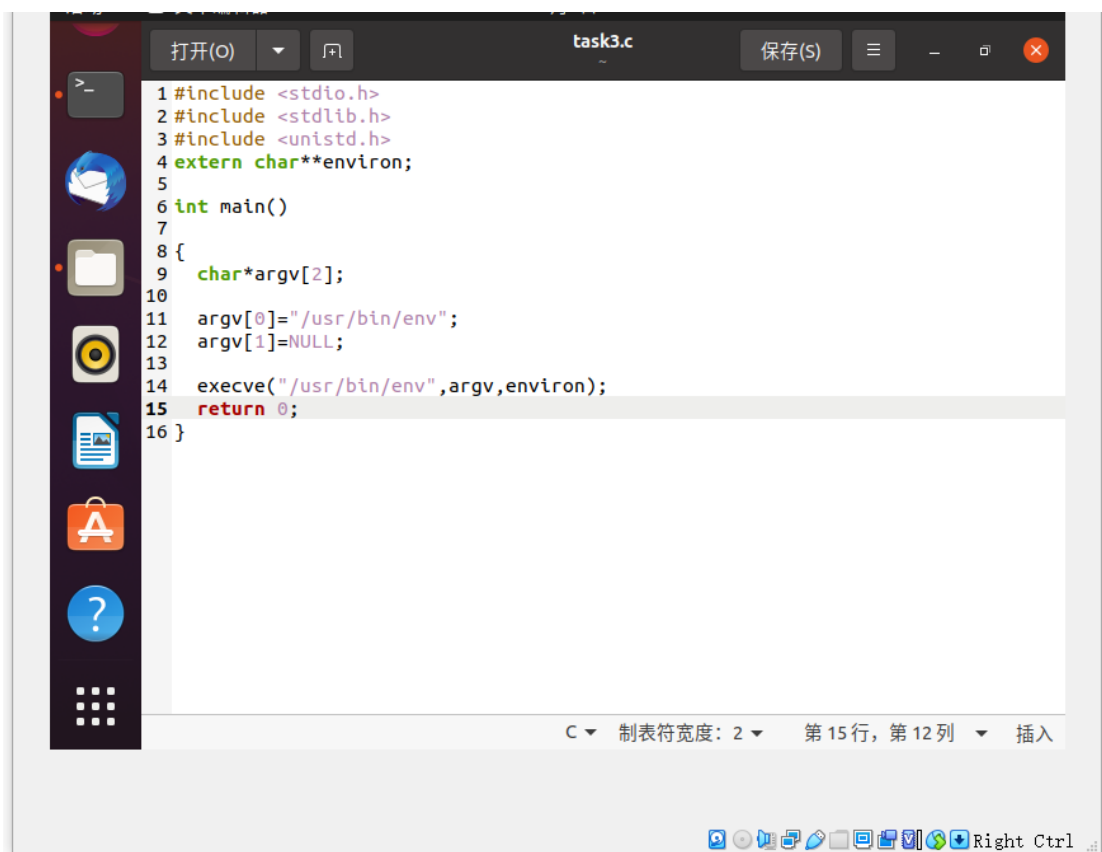
源程序：

更改前：



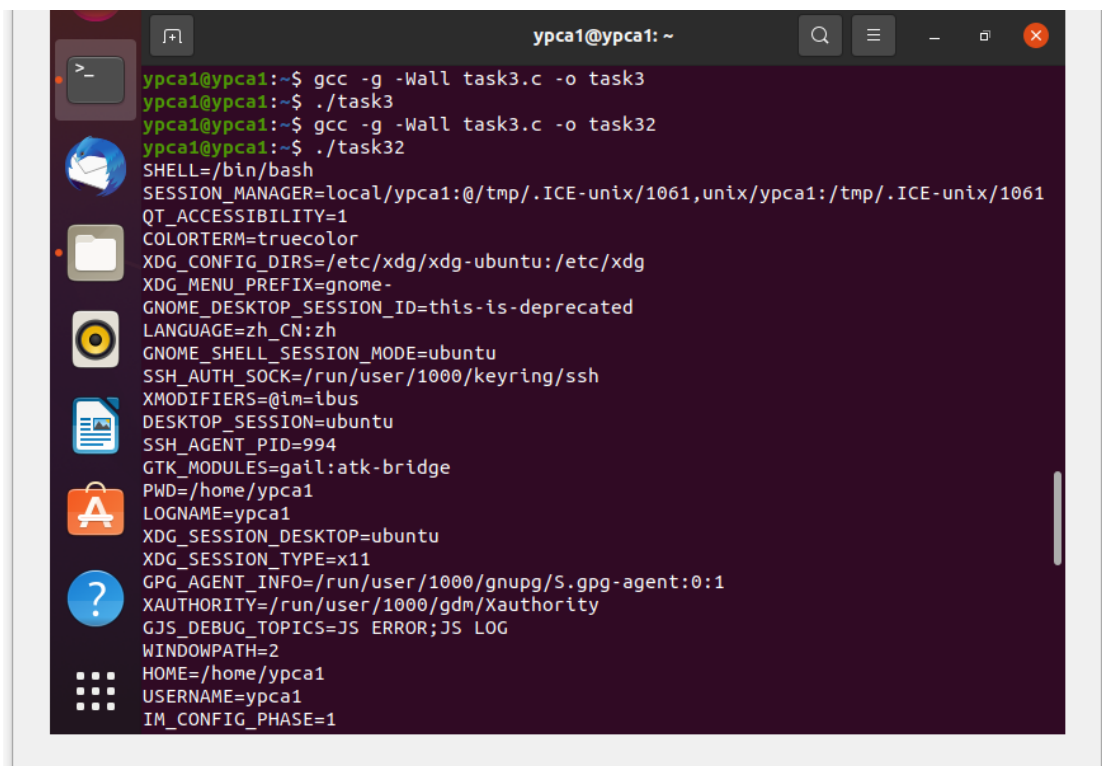
```
1#include <stdio.h>
2#include <stdlib.h>
3#include <unistd.h>
4extern char**environ;
5
6int main()
7
8{
9    char*argv[2];
10
11    argv[0]="/usr/bin/env";
12    argv[1]=NULL;
13
14    execve("/usr/bin/env",argv,NULL);
15    return 0;
16}
```

更改后：



```
1#include <stdio.h>
2#include <stdlib.h>
3#include <unistd.h>
4extern char**environ;
5
6int main()
7{
8{
9    char*argv[2];
10
11    argv[0]="/usr/bin/env";
12    argv[1]=NULL;
13
14    execve("/usr/bin/env",argv,environ);
15    return 0;
16 }
```

运行结果:



```
ypca1@ypca1: ~
ypca1@ypca1:~$ gcc -g -Wall task3.c -o task3
ypca1@ypca1:~$ ./task3
ypca1@ypca1:~$ gcc -g -Wall task3.c -o task32
ypca1@ypca1:~$ ./task32
SHELL=/bin/bash
SESSION_MANAGER=local/ypca1:@/tmp/.ICE-unix/1061,unix/ypca1:/tmp/.ICE-unix/1061
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
LANGUAGE=zh_CN:zh
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=994
GTK_MODULES=gail:atk-bridge
PWD=/home/ypca1
LOGNAME=ypca1
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/run/user/1000/gdm/Xauthority
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
WINDOWPATH=2
HOME=/home/ypca1
USERNAME=ypca1
IM_CONFIG_PHASE=1
```

结果表明, 前一次未输出环境变量而后一次可以成功输出环境变量。

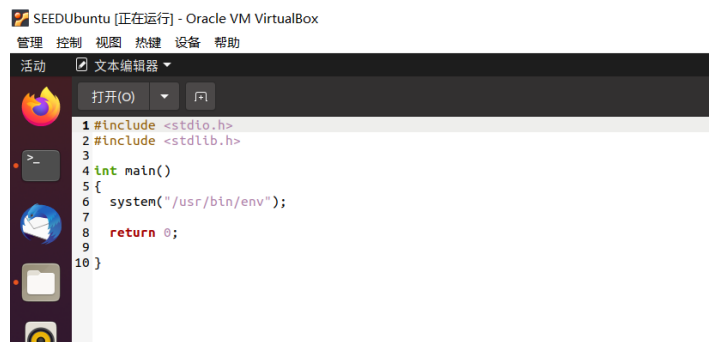
实验体会: 刚开始没发现少了一个头文件, 运行不成功, 试了好几次才发现。

## Task 4

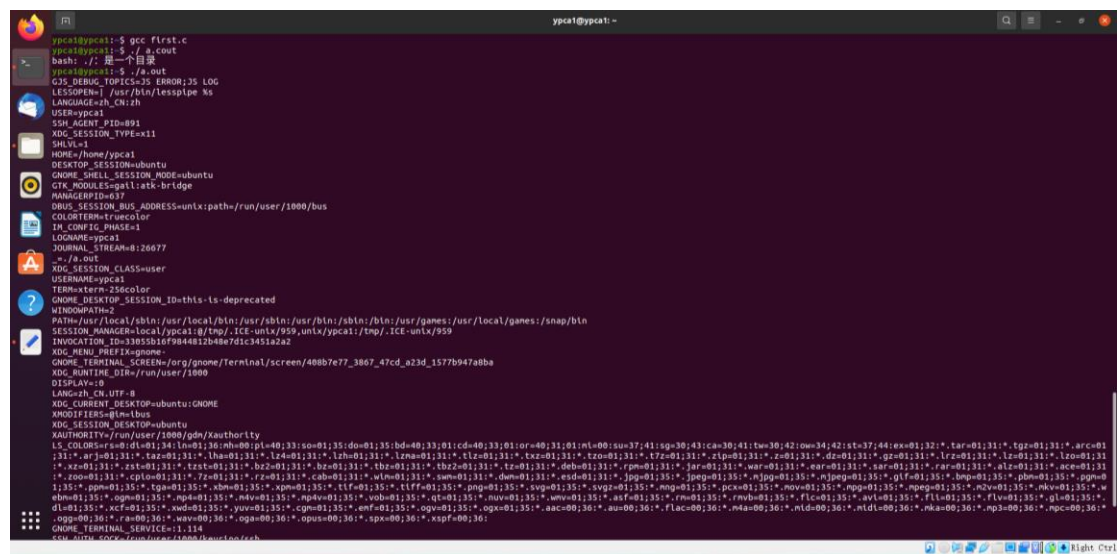
**实验内容：**编译并运行程序，描述观察到的结果。

**实验目的：**了解环境变量和 `execve()` 函数

源程序：



运行结果：



## Task 5

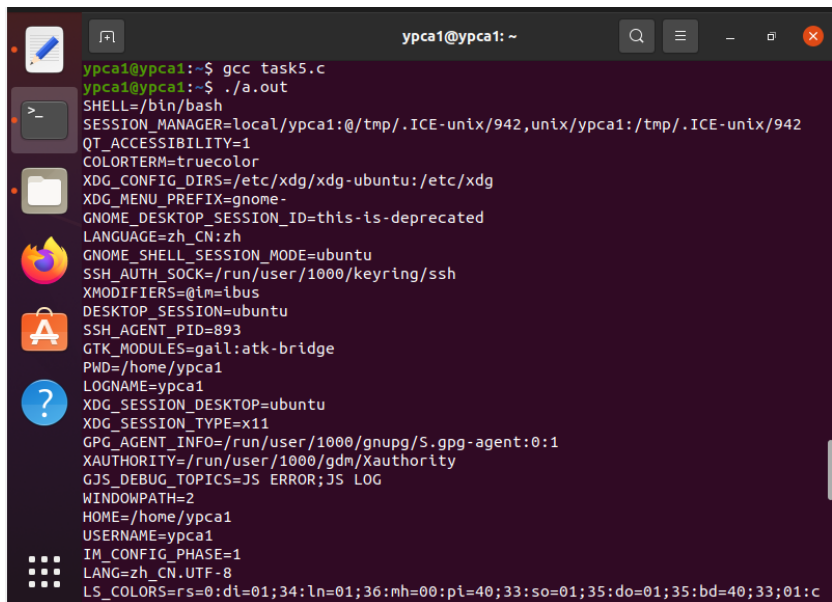
实验内容： 了解 set-uid 程序

源程序：



```
1#include <stdio.h>
2#include <stdlib.h>
3
4extern char** environ;
5
6void main()
7{
8    int i=0;
9    while (environ[i] !=NULL){
10        printf("%s\n",environ[i]);
11        i++;
12    }
13}
```

运行结果：



```
ypca1@ypca1: ~$ gcc task5.c
ypca1@ypca1: ~$ ./a.out
SHELL=/bin/bash
SESSION_MANAGER=local/ypca1:/tmp/.ICE-unix/942,unix/ypca1:/tmp/.ICE-unix/942
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
LANGUAGE=zh_CN:zh
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=893
GTK_MODULES=gail:atk-bridge
PWD=/home/ypca1
LOGNAME=ypca1
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTORITY=/run/user/1000/gdm/Xauthority
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
WINDOWPATH=2
HOME=/home/ypca1
USERNAME=ypca1
IM_CONFIG_PHASE=1
LANG=zh_CN.UTF-8
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:c
```



```
ypca1@ypca1: ~  
ypca1@ypca1:~$ sudo chown root task5.c  
ypca1@ypca1:~$ sudo chmod 4755 task5.c  
ypca1@ypca1:~$ gcc task5.c  
ypca1@ypca1:~$ ./a.out  
SHELL=/bin/bash  
SESSION_MANAGER=local/ypca1:0/tmp/.ICE-unix/942,unix/ypca1:0/tmp/.ICE-unix/942  
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  
LANGUAGE=zh_CN:zh  
GNOME_SHELL_SESSION_MODE=ubuntu  
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh  
XMODIFIERS=@im=ibus  
DESKTOP_SESSION=ubuntu  
SSH_AGENT_PID=893  
GTK_MODULES=gail:atk-bridge  
PWD=/home/ypca1  
LOGNAME=ypca1  
XDG_SESSION_DESKTOP=ubuntu  
XDG_SESSION_TYPE=x11  
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1  
XAUTHORITY=/run/user/1000/gdm/Xauthority  
GJS_DEBUG_TOPICS=JS ERROR;JS LOG  
WINDOWPATH=2  
HOME=/home/ypca1  
USERNAME=ypca1  
IM_CONFIG_PHASE=1
```

## Task 6

源程序：

```
task6.c  
1 int main()  
2 {  
3     system("ls");  
4     return 0;  
5 }
```

运行结果：

```
ypca1@ypca1: ~  
ypca1@ypca1:~$ export PATH=/home/seed:$PATH  
ypca1@ypca1:~$ gcc task6.c  
task6.c: In function 'main':  
task6.c:3:2: warning: implicit declaration of function 'system' [-Wimplicit-fun  
ction-declaration]  
3 | system("ls");  
ypca1@ypca1:~$ sudo chown root task6.c  
[sudo] ypcal 的密码:  
ypca1@ypca1:~$ sudo chmod 5744 task6.c  
ypca1@ypca1:~$ gcc task6.c  
task6.c: In function 'main':  
task6.c:3:2: warning: implicit declaration of function 'system' [-Wimplicit-fun  
ction-declaration]  
3 | system("ls");  
ypca1@ypca1:~$
```

Task 8

源程序：

更改前

更改后

```
1 #include <string.h>  
2 #include <stdio.h>  
3 #include <stdlib.h>  
4  
5 int main(int argc, char *argv[])  
6 {  
7     char* v[3];  
8     char* command;  
9  
10    if(argc < 2){  
11        printf("Please type a file name.\n");  
12        return 1;  
13    }  
14    v[0]="/bin/cat"; v[1]=argv[1]; v[2] =NULL;  
15    command =malloc(strlen(v[0]) +strlen(v[1])+2);  
16    sprintf(command,"%s %s",v[0],v[1]);  
17  
18    system(command);  
19    //execve(v[0],v,NULL);  
20    return 0;  
21 }
```

```
1 #include <string.h>  
2 #include <stdio.h>  
3 #include <stdlib.h>  
4  
5 int main(int argc, char *argv[])  
6 {  
7     char* v[3];  
8     char* command;  
9  
10    if(argc < 2){  
11        printf("Please type a file name.\n");  
12        return 1;  
13    }  
14    v[0]="/bin/cat"; v[1]=argv[1]; v[2] =NULL;  
15    command =malloc(strlen(v[0]) +strlen(v[1])+2);  
16    sprintf(command,"%s %s",v[0],v[1]);  
17  
18    //system(command);  
19    execve(v[0],v,NULL);  
20    return 0;  
21 }
```

运行结果：

```
ypca1@ypca1:~$ gcc task8.c
task8.c: In function 'main':
task8.c:19:2: warning: implicit declaration of function 'execve' [-Wimplicit-fu
nction-declaration]
   19 |     execve(v[0],v,NULL);
       |     ~~~~~
ypca1@ypca1:~$ ./a.out
Please type a file name.
ypca1@ypca1:~$ sudo chown root task8.c
[sudo] ypca1 的密码:
ypca1@ypca1:~$ sudo chmod 4755 task8.c
ypca1@ypca1:~$ gcc task8.c
task8.c: In function 'main':
task8.c:19:2: warning: implicit declaration of function 'execve' [-Wimplicit-fu
nction-declaration]
   19 |     execve(v[0],v,NULL);
       |     ~~~~~
ypca1@ypca1:~$ ./a.out
Please type a file name.
ypca1@ypca1:~$
```

