Sistemas Operativos.
Práctica 02

Alumno:
Abrego Álavrez Jonathan

Laboratorista:
Cinthia Rodríguez Maya

25 de febrero de 2014

Encuenta el error si existe, en cada una de las siguientes porciones de código, explica como podrías corregir cada uno:

•Código 1

```
int *numero;
printf("%d", *numero);
```

El Código 1 esta correcto lo único que se podria decir que como es un apuntador en el identificador seria agregar "ptr" quedando algo del estilo:

```
int ptrNumero;
printf("%d", *ptrNumero);
```

•Código 2

```
float *ptrReal;
long *ptrEntero;
ptrEntero=ptrReal;
```

El Código 2 esta erroneo, ya que el tipo de cada uno es diferente a mi parecer una solución seria hacer un cambio de tipo (Casteo) de la siguiente manera:

```
 float *prtReal;
 long *prtEntero;
 prtEntero=(long*)prtReal;
```

•Código 3

```
int *x,yL;
x=y;
```

El código es erroneo, y una solución seria agregar un andpersan en la asignación antes de "y", quedando de la siguiente forma:

```
int *x,y;
x=&y;
```

•Código 4

```
char s[]="esto es un arreglo de caracteres";
while(*s!='\0 '){
  printf("%c",*s);
  s++;
}
```

El Código 4 es erroeno puesto que falta una forma de hacer que se vaya recorriendo el arreglo. Y la única solución que encontre para que se imprimiera el arreglo en terminal de forma adecuada es la siguiente:

```
char s[]="este es un ejemplo de arreglo de caracteres";
int i=0;
while(*s != '\0'){
  *s=(char*)s[i++];
   printf("%c",*s);
}
print("\n");
```

•Código 5

```
short *ptrNum, resultado;
void *ptrGenerico = ptrNum;
resultado = *ptrGenerico + 7;
```

El Código 5 es erroneo, según lo poco que tengo entendido no se puede asignar de un tipo definido (int, float, etc) hacia un apuntador y por eso creo que queda algo de la forma:

```
short *ptrNum, *resultado;
void *ptrGenerico = ptrNum;
*resultado = (short *)ptrGenerico + 7;
```

•Código 6

```
float x= 19.34;
float ptrX=&x;
printf("%f \n",ptrX);
```

El código 6 es erroneo puesto que el la parte del printf falta el asterisco antes de ptrX, entonces el código quedaria de la siguiente manera

```
float x=19.34;
float *ptrX=&x;
printf("%f \n",*ptrX);
```

Implementa el algoritmo de ordenamiento *Insertion Sort* utilizando apuntadores.

```
void insertionSort(int *a[], int tam) {
    int i;
    for (i=0; i < tam; i++) {
      int j;
      int *val = a[i];
      for (j = i - 1; j >= 0; j--) {
         if (a[j] <= val) break;
         a[j + 1] = a[j];
      }
      a[j + 1] = val;
     }
}
```

MINIX
Una vez que hayas instalado MINIX realiza las siguientes actividades.

- Explica lo que muestra la ejección del comando *sysenv*.
  Cuando se ejecuta el comando *sysenv* en la terminal se muestra lo siguiente "rootdevname=c0d0p0s0".Sysenv solicita el valor de una o más variables de arranque.
- Mediante *pwd* muesta cual es el directorio actual al ejecutar:

```
#cd /usr/src
#cd
#cd /usr/src
#cd boot
#cd
```

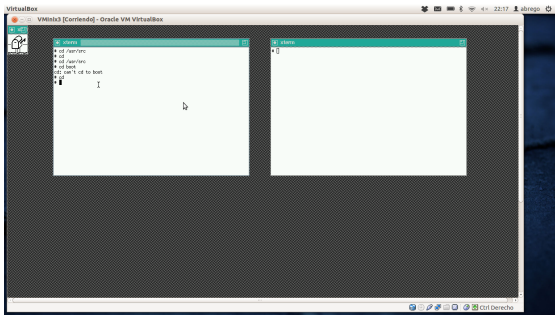

Figura 1: Screenshot pwd

Ejecutando los comandoas tal cual como vemos hay un punto donde dicho directorio ("boot") no esta ubicado sobre la ruta en la que estamos situados. Haciendo un poco de busqueda podemos ver que se encuentra ubicado en otra ruta diferente. Ahora haciendo un poco de uso de MINIX y viendo la sintaxis del comando podemos ver que suponiendo que boot se encontrar en donde se menciona que esta (en la práctica) al entrar el último "cd" este nos regresaría al directorio raiz, puesto que si no tiene argumentos es lo que hace dicho comando(vease man cd).Por lo tanto al hacer *pwd* nos mostraria que nuestra ubicación es */root*

- Mediante *ps* determina cuantos procesos estan ejecutandose en el sistema y cuáles son:
  El número de procesos activos y cuales son se muestran en la siguiente imagen:

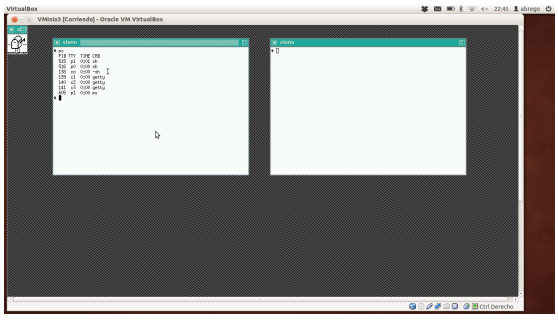

Figura 2: Screenshot ps

- Mediante *find* busca el archivo proc.c y establece su ruta.
  El archivo proc.c se encuentra ubicado en la ruta: "/usr/src/kernel"

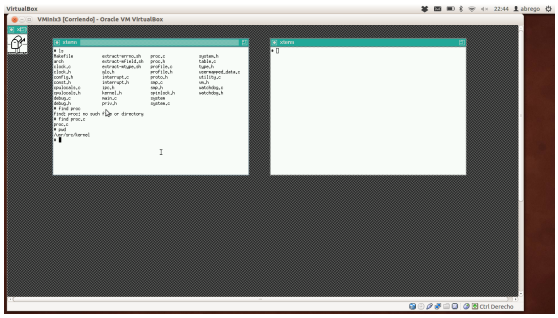

Figura 3: Screenshot proc.c

- Explica lo que hacen los siguientes comandos

  •`cp: Copia un archivo a otro, o copia uno o mas arcvhivos a un directorio.`

  •`chgpr: Permite cambiar el grupo y, opcionalmente, el usuario de los archivos a grupo y usuario. Solo el super usuario puede cambiar el grupo indicado grupos arbitrarios.`

  •`su: Permite loguearse temporalmente como el super usuario u otro usuario. Sin argumentos, se asume root. A los usuarios normales se le solicitará la passsword del usuario con cuyo nombre estan tratando de loguearse, pero cualquier usuario con un gid=0 esta no le será solicitada.`

- ¿Cuál es el contenido del archivo */usr /src/.profile* y para que sirve?
  El archivo */usr/src/.profile* permite al usuario configurar el entorno de su cuenta autómaticamente cuando entra al sistema.