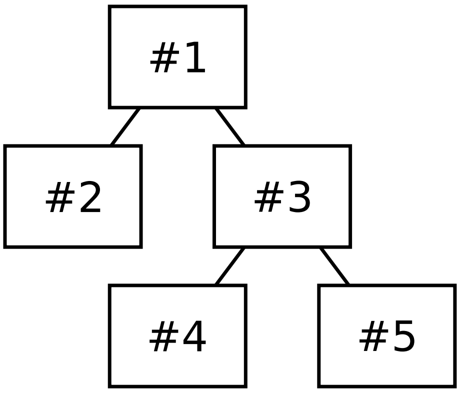
1 OPERATING SYSTEMS LABORATORY XIII - Laboratory Review

1.1 Processes

1. **Exercise:** A sample program which creates 4 child processes:



1.2 Memory

- **Exercise:** Study the following code:

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

void foo();
int *var;

int main()
{
  // Memory allocation for var
  var=(int*)malloc(sizeof(int)*4);

  // Set and display data
  var[0] = 4;
  printf("%d ", var[0]);

  // Free memory space of the pointers
  free(var);
  printf("%d ", var[0]);

  // Call a function for alloca()
  foo();
  //free(var);

  return 0;
}

void foo()
{
  var=(int*)alloca(sizeof(int)*4);
  var[0] = 4;
}
```

- Uncomment the free line after calling the foo() function and execute the code again. What happens?

1.3 File Operations

1. **Exercise:** Changing file access attributes:

   - Create a file:

     ```
     ceng328@cankaya$ touch foo
     ```

   - Compile and execute the following code:

     ```
     #include <sys/stat.h>
     int main()
     {
       // Create a stat structure for file information
       struct stat buf;

       // Take away all available rights from file
       chmod("foo", 0);

       // Give read rights for everyone and and execute right
       // for the owner
       chmod("foo", S_IRUSR | S_IRGRP | S_IROTH | S_IXUSR);

       // Take away read right from other users
       stat("foo", &buf);
       chmod("foo", buf.st_mode ^ S_IROTH);

       // Print current file access rights
       stat("foo", &buf);
       printf("foo - %4.4o\n", buf.st_mode & 07777);

       return 0;
     }
     ```

   - Give this file the following access rights:
     - rwxr-xr-x
     - r--------
     - r-xr-x---