

# ELEC 377

# Operating Systems

---

Tutorial / Lab

Week 2 / 3

# Pointers to functions

---

- in Java, you passed an object to have another function call you back

```
class foo implements ActionListener{  
    void action(){  
        ...  
    }  
    void init(){  
        ...  
        panel.addListener(this);  
    }  
}
```

# Pointers to functions

---

- in C, no object instances with methods to pass.
  - ◊ instead we pass a pointer to a function
  - ◊ a function name by itself is a constant pointer

```
int lt(int x, int y) { return x < y; }
```

```
int (*f)(int,int); // f is a function with 2 int parms  
                  // and returns int
```

```
int main(){  
    f = lt;  
    printf(“%d\n”,    (*f)(2,3)    ); // prints number 1  
}
```

# Pointers to functions

---

- brackets important!!!!
- \*f(2,3) -> call function named f with the parameters 2 and 3 and treat the result as a pointer and dereference the pointer
- (\*f)(2,3) -> use f as a pointer to a function and call with the parameters 2 and 3

# Lab 1 main program

---

```
int main(){
    struct dirent **namelist;
    int n;

    n = scandir("/proc", &namelist,
                isProcessDir, NULL);

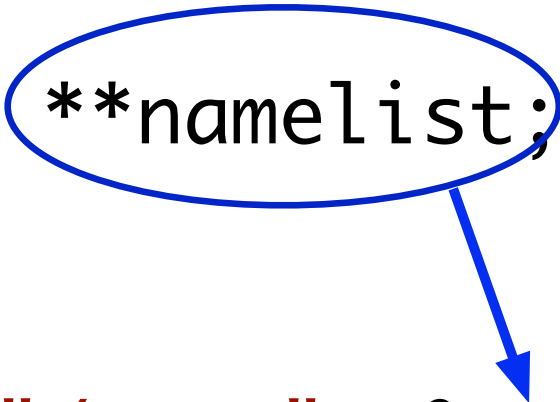
    . . .
}
```

# Lab 1 main program

---

```
int main(){  
    struct dirent **namelist;  
    int n;  
  
    n = scandir("/proc", &namelist,  
                isProcessDir, NULL);  
  
    . . .  
}
```

*array of pointers  
to struct dirent*



# Lab 1 main program

---

```
int main(){
    struct dirent **namelist;
    int n;

    n = scandir("/proc", &namelist,
                isProcessDir, NULL);

    . . .
}
```

# Lab 1

---

```
int isProcessDir(const struct dirent* d){  
    ... return 1 if a process, 0 if not a process...  
    ... i.e. all characters of the name  
        (d->d_name are digits...)  
}
```



# Lab 1 main program

---

```
int main(){
    struct dirent **namelist;
    int n;

    n = scandir("/proc", &namelist,
                isProcessDir, NULL);

    .... use namelist[i]->d_name ....
    ... string concatenation (status) ....
    .... open file and read info.
}
```

# Lab1

---

- Do the lab iteratively
  - start by writing the `isProcessdir` function and print `d_name` fields of the `namelist`.  
\*\*\*\*test \*\*\*\*
  - Then do the string concatenation and print the resulting file names
  - Then open the files to get and print the information.
- Links to tutorials for the following editors are on the website: `jed`, `joe`, `vi`

# Stack Overflow

---

- The point of this lab is to familiarize yourself with C, function pointers and the link between C and the low level memory representation.
  - Stack overflow will tell you about `strspan`, etc.
  - Not the point of the lab.
  - Also, do not use `fscanf`. Use `fgets` and work with the bytes of the line directly.

# Testing

---

- What command provides the same data as your program
  - What options
  - use the 'man' command to find out about options
- What are the differences, why are they there?