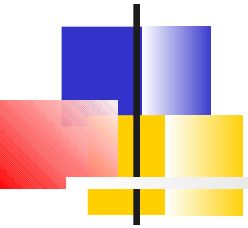


The Switch Statement

The `switch` Statement

- An optimized form when you need to check for one of several values.

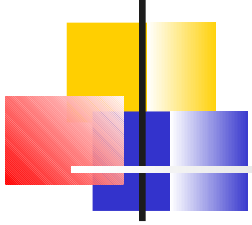
Gauge	Rho
12	5.211
14	8.285
16	13.170
18	20.950

- Merge multiple if ... else ... statements into a single compound statement.

The **switch** Statement

- A switch statment can replace a series of “if” ... “else”... statements

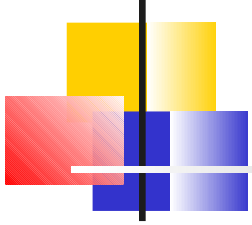
```
switch (gauge)
{
   case 12:
       rho = 5.211; break;
   case 14:
       rho = 8.285; break;
   case 16:
       rho = 13.17; break;
   case 18:
       rho = 20.95; break;
   default:
       printf (“Invalid gauge\n”);
}
```



The `switch` Statement

- The compiled code compares gauge to each “case” in turn.
- When an equal value is found, execute the corresponding code.

- If no equal value if found, execute the default code.
  - If no default, just continue below the block.



The `switch` Statement

Example: Suppose gauge is 16.

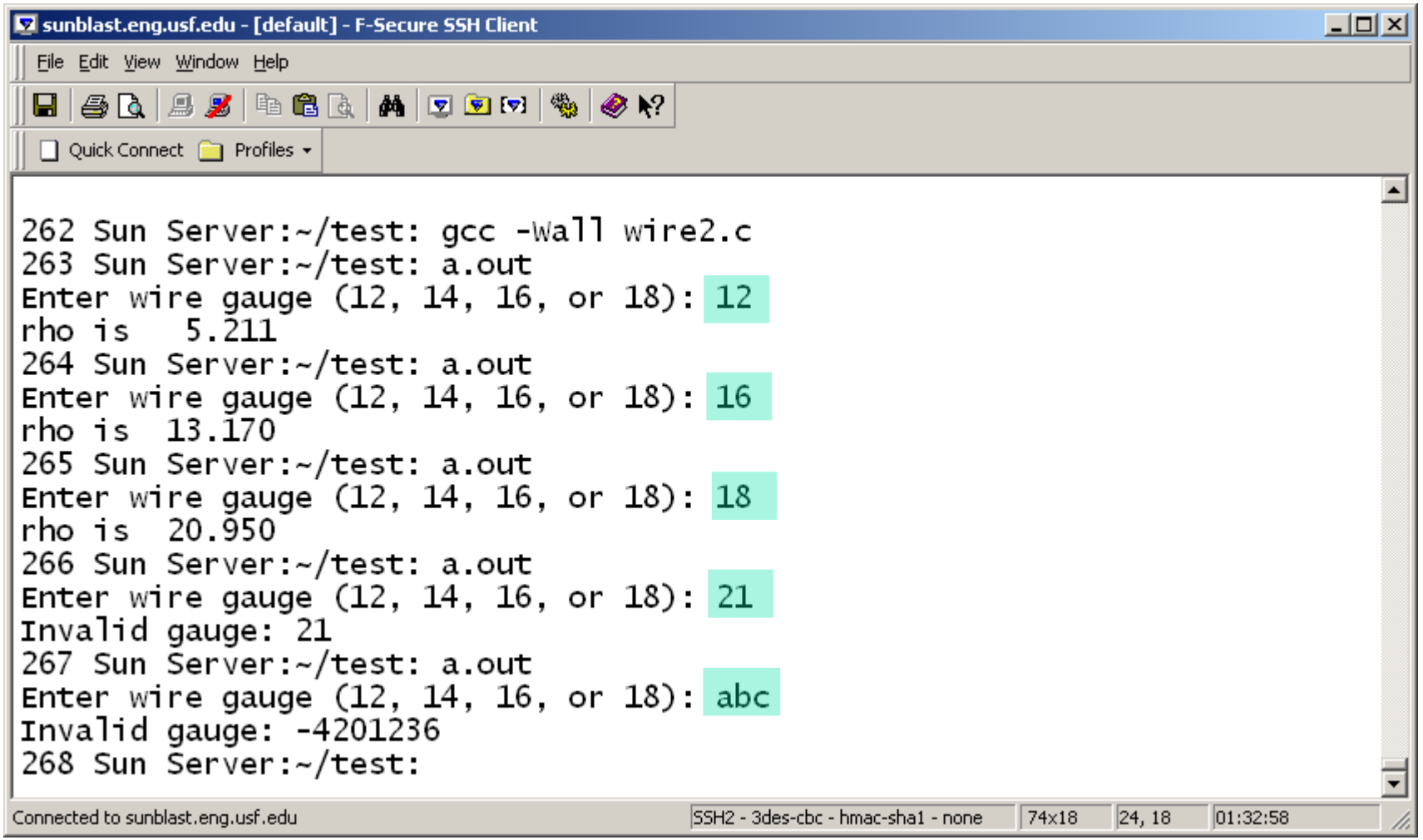
```
switch (gauge)
{
   case 12:
        rho = 5.211; break;
   case 14:
        rho = 8.285; break;
   case 16:
        rho = 13.17; break;
   case 18:
        rho = 20.95; break;
   default:
        printf (“Invalid gauge\n”);
}
/* Continue here. */
```

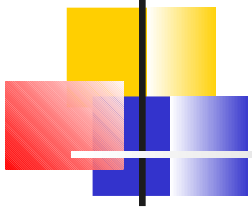
```
#include <stdio.h>
int main()
{
    int gauge;
    double rho;
    printf ("Enter wire gauge (12, 14, 16, or 18): ");
    scanf("%d", &gauge);

    switch (gauge)
    {
        case 12:
            rho = 5.211;   break;
        case 14:
            rho = 8.285;   break;
        case 16:
            rho = 13.17;   break;
        case 18:
            rho = 20.95;   break;
        default:
            printf ("Invalid gauge %d\n", gauge);
            return 1;
    }

    printf ("rho is %7.3f\n", rho);
    return 0;
}
```

Program wire2.c Running



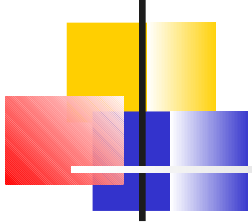


A common mistake:

No break

```
switch (gauge)
{
  case 12:
     rho = 5.211;
  case 14:
     rho = 8.285;
  case 16:
     rho = 13.17;
  case 18:
     rho = 20.95;
  default:
     printf (“Invalid gauge\n”); return 1;

}
```



Fall Through

What happens if there is no break at the end of the code for a case?

Answer: the code in the following case statement will be executed

This will continue until we reach a break statement or until we reach the end of the switch statement.

The **switch** Statement

Fall through:

```
switch (gauge)
{
   case 12:   rho = 5.211;
   case 14: rho = 8.285;
   case 16: rho = 13.17;
   case 18: rho = 20.95; break;
   default:   printf (“Invalid gauge\n”);
              return 1;
}
```

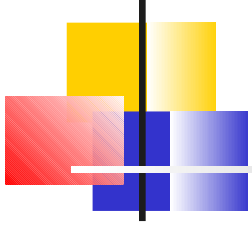
No break

If gauge has value 12, 14, 16 or 18, rho will end up with value 20.95

The **switch** Statement

If the switch statement is inside a function, you can use “return” instead of “break”.

```
switch (gauge)
{
  case 12: rho = 5.211;    return rho;
  case 14: rho = 8.285;    return rho;
  case 16: rho = 13.17;    return rho;
  case 18: rho = 20.95;    return rho;
  default: printf (“Invalid gauge\n”);
           abort();
}
/* Never reach this point. */
```



Programming Style Issues

A switch statement should look like a table.

The entire block should should be visible on your screen at the same time.

Keep it simple.

You can include multiple statements for a case.

- But only a few.
- Use a function if you need to do more.

The `switch` Statement

```
switch (gauge)
{
    case 12:
        rho = 5.211;
        phi = 3.457;
        break;
    case 14:
        rho = 8.285;
        phi = 4.657;
        break;
    case 16:
        rho = 13.17;
        phi = 8.381;
        break;
    case 18:
        rho = 20.95;
        phi = 12.45;
        break;

    default:
        printf (“Invalid gauge\n”);  abort();
}
```